

# **Looking Into E-Learning in Bangladesh Implementing SUDOKU Puzzle and KUMON Learning System**

**A Thesis  
Submitted to the Department of Computer Science and Engineering  
Of  
BRAC University  
By**

**Sheikh Mohammad AshrafUllah Mishu**

**ID. # 02201122**

**Md. Rahat Khairullah**

**ID. # 02201154**

**Sumit Roy**

**ID. # 03101056**

**In Partial Fulfillment of the requirements for the Degree**

**Of**

**Bachelor of Science in Computer Science  
August 2007**

## Declaration

We, Sheikh Mohammad AshrafUllah Mishu university ID: 02201122, Sumit Roy University ID: 03101056 and Md. Rahat Khairullah University ID: 02201154, have completed the proposed Thesis, “**Looking Into E-Learning in Bangladesh: Implementing SUDOKU puzzle and KUMON Learning System**” Under **CSE 400** course based on the result found by us.

We therefore declare that this project has not been published previously neither in whole nor in part of any degree accepts this publication. We also mentioned work done by other researchers by reference.

Signature of Supervisor

Signature of

-----  
Bushra Tawfiq Chowdhury  
Lecturer  
CSE department  
BRAC University

-----  
Students  
CSE department  
BRAC University

## **Acknowledgement**

We would like to express our gratitude to our respected thesis supervisor Bushra Tawfiq Chowdhury, for her earnest help and all sort of supportive suggestions and cooperation. We would like to express our gratefulness to our co advisor Dr. Yousuf Mahbubul Islam for his contribution on our thesis. Our special thanks also go to Risat Pathan for his important suggestions and useful resources. Without their co-operation it would not be possible for us to organize such an enormous project and a thesis report on a short span of time.

We would like to give special thanks to Monawar Hosain, lecturer (DU) for his dynamic guidance in the coding part.

We also convey our gratefulness to all the faculty members and staffs of the computer Science and engineering Department of BRAC University.

## **Abstract**

To build a knowledge based web portal from the 'Distant Learning' point of view where, a well-known puzzle game "SUDOKU", a very new math learning method "KUMON" and image based "Logical Learning" will be implemented.

SUDOKU is a number oriented puzzle game with different levels for any one, where one has to put numbers in certain rules. There are some ways to generate the puzzle, we will generate the puzzle in a different way which is more easier and more efficient. There will be an analysis on different puzzle generating algorithms and on the new one.

KUMON is a different kind of learning program designed to help children master the basic of math and reading. The method uses abacus as a tool. Our portal will provide this learning system where any one can access and improve their selves.

Image based "Logical Learning" will be introduced in our portal. This is a method where students will force to think logically with the help of images relevant to the topics.

## Index

Topic Name	Page No
<b>1. Introduction</b>	<b>1</b>
<b>1.1 E learning</b>	<b>1</b>
<b>1.1 Distance learning:</b>	<b>2</b>
<b>1.2 E learning world wide:</b>	<b>2</b>
<b>1.3 E-learning in Bangladesh</b>	<b>3</b>
<b>1.4 Our contribution in E learning in Bangladesh</b>	<b>4</b>
<b>2. Our Target</b>	<b>4</b>
<b>3. Web portal</b>	<b>4</b>
<b>4. Why Sudoku?</b>	<b>5</b>
<b>4.1 Cognitive Assessment</b>	<b>6</b>
<b>4.2 Elimination</b>	<b>6</b>
<b>4.3 Logical Thinking</b>	<b>6</b>
<b>4.4 Analysis</b>	<b>6</b>
<b>5. Why Kumon?</b>	<b>6</b>
<b>6. Sudoku</b>	<b>7</b>
<b>6.1 History</b>	<b>8</b>
<b>6.2 How to Play</b>	<b>9</b>
<b>6.3 Sudoku Techniques:</b>	<b>12</b>
<b>6.3.1 Elimination Technique</b>	<b>13</b>
<b>6.3.2 Column, Row, and Mini-grid Elimination</b>	<b>13</b>
<b>6.3.3 Intermediate Techniques</b>	<b>18</b>
<b>6.3.3.1 Lone Rangers</b>	<b>18</b>
<b>6.3.3.2 Lone Rangers in a Mini-grid</b>	<b>19</b>
<b>6.3.3.3 Lone Rangers in a Row</b>	<b>21</b>

<b>6.3.3.4 Lone Rangers in a Column</b>	<b>22</b>
<b>6.3.4 Advanced Techniques</b>	<b>24</b>
<b>6.3.4.1 Looking for Twins</b>	<b>24</b>
<b>6.3.4.2 Looking for Triplets</b>	<b>28</b>
<b>6.4.2.1 Variants of Triplets</b>	<b>29</b>
<b>6.4.2.2 Brute-Force Elimination</b>	<b>32</b>
<b>6.4 Available algorithms</b>	<b>36</b>
<b>6.4.1 Sudoku Generation Algorithm</b>	<b>38</b>
<b>6.4.2 Solving Sudoku by backtracking</b>	<b>39</b>
<b>6.4.3 Solving Sudoku by a brute-force algorithm</b>	<b>41</b>
<b>6.4.4 Solving Sudoku via Stochastic Search / Optimization methods.</b>	<b>42</b>
<b>6.5 Complexity analysis</b>	<b>42</b>
<b>6.5.1 Hints and Heuristics</b>	<b>44</b>
<b>6.5.2 Counting Solutions</b>	<b>44</b>
<b>6.5.3 NP or Not NP That Is the Question</b>	<b>46</b>
<b>6.5.4 Logic Rules</b>	<b>47</b>
<b>6.5.5 A Satisfied Mind</b>	<b>48</b>
<b>6.6 Proposed algorithm:</b>	<b>49</b>
<b>6.7 Flow chart for our new system</b>	<b>51</b>
<b>6.8 Our first try:</b>	<b>52</b>
<b>6.8.1 What is Local Search?</b>	<b>52</b>
<b>6.8.2 Problem faced with our first try:</b>	<b>53</b>
<b>6.8.3 New solution developed using AI approach.</b>	<b>54</b>
<b>6.8.4 Algorithm used in our system (Hill climbing)</b>	<b>55</b>
<b>7. Kumon</b>	<b>57</b>
<b>7.1 To implement Kumon we performed</b>	<b>57</b>
<b>7.2 The Promises of Distance Learning</b>	<b>57</b>
<b>7.3 Concerns</b>	<b>58</b>
<b>7.3.1 Instructor Concerns</b>	<b>58</b>

<b>7.3.2 Student Concerns</b>	<b>58</b>
<b>7.4 Design considerations</b>	<b>59</b>
<b>7.4.1 Systematic design and development</b>	<b>59</b>
<b>7.4.2 Interactivity</b>	<b>59</b>
<b>7.4.3 Active learning</b>	<b>59</b>
<b>7.5 Internet conducted Web-based Kumon method</b>	<b>60</b>
<b>7.6 How do we relate Kumon with distance education?</b>	<b>60</b>
<b>7.7 Starting time</b>	<b>62</b>
<b>7.8 Step of success with Kumon:</b>	<b>62</b>
<b>7.9 Parents question and our feedback</b>	<b>64</b>
<b>7.10 Parental involvement</b>	<b>65</b>
<b>7.11 Advantages</b>	<b>65</b>
<b>7.12 Evaluation</b>	<b>65</b>
<b>7.13 Traditional Kumon Script</b>	<b>66</b>
<b>7.14 Problem with traditional Kumon script</b>	<b>67</b>
<b>7.15 Our proposed web based Kumon exam Script</b>	<b>68</b>
<b>7.16 Database Design</b>	<b>69</b>
<b>7.17 ER-diagram</b>	<b>70</b>
<b>7.18 System Architecture</b>	<b>71</b>
<b>7.19 Implementation: Web site</b>	<b>72</b>
<b>8. Conclusion</b>	<b>80</b>
<b>9. Bibliography</b>	<b>81</b>

## Figure List

Lists Of Figure	Page No
<b>Figure 1 The word Sudoku in Japanese</b>	<b>7</b>
<b>Figure 1-1 A Sudoku grid</b>	<b>7</b>
<b>Figure 1-2 deriving the number for a cell based on elimination</b>	<b>13</b>
<b>Figure 1-3 Partially filled Sudoku puzzle</b>	<b>13</b>
<b>Figure 1-4 Scanning by column</b>	<b>14</b>
<b>Figure 1-5 Scanning by row</b>	<b>14</b>
<b>Figure 1-6 Filling in the value for (2,1)</b>	<b>15</b>
<b>Figure 1-7 Scanning by column and row</b>	<b>15</b>
<b>Figure 1-8 Scanning within the mini-grid</b>	<b>16</b>
<b>Figure 1-9 Examining cells (1,4)</b>	<b>16</b>
<b>Figure 1-10 Filling in (1, 4) with the value 3</b>	<b>16</b>
<b>Figure 1-11 Confirming the value for (2,6) by scanning column, row, and mini-grid</b>	<b>17</b>
<b>Figure 1-12 Filling in the value for (1,7)</b>	<b>17</b>
<b>Figure 1-13 Identifying a long ranger in a row</b>	<b>18</b>
<b>Figure 1-14 A partial Sudoku puzzle</b>	<b>19</b>
<b>Figure 1-15 Confirming the values of three cells</b>	<b>19</b>
<b>Figure 1-16 Possible values for the cells after applying the CRME technique</b>	<b>20</b>
<b>Figure 1-17 Examining the third mini-grid</b>	<b>20</b>
<b>Figure 1-18 Confirming the value for (7,2)</b>	<b>21</b>
<b>Figure 1-19 Searching a Sudoku puzzle for lone rangers in rows</b>	<b>21</b>
<b>Figure 1-20 Lone ranger detected in row 5</b>	<b>22</b>
<b>Figure 1-21 Searching a Sudoku puzzle for lone rangers in columns</b>	<b>22</b>
<b>Figure 1-22 Finding a lone ranger in a columns</b>	<b>23</b>
<b>Figure 1-23 Confirming yet another cells</b>	<b>24</b>
<b>Figure 1-24 A partially solved Sudoku puzzle with the possible values for empty cells</b>	<b>25</b>
<b>Figure 1-25 identifying the twins</b>	<b>25</b>
<b>Figure 1-26 Eliminating 2 and 3 as possible values for other cells in the same row as the twins</b>	<b>26</b>



<b>Figure 1-28 New pairs of twins emerging after the first scanning</b>	<b>27</b>
<b>Figure 1-29 Confirming cell (2,1) and subsequently (2,4)</b>	<b>27</b>
<b>Figure 1-30 The grid after scanning for twins in the rows, columns, and mini-grid</b>	<b>28</b>
<b>Figure 1-31 A partial Sudoku puzzle</b>	<b>29</b>
<b>Figure 1-32. Modifying the mini-grid where the triplets</b>	<b>29</b>
<b>Figure 1-34 A set of triplets</b>	<b>30</b>
<b>Figure 1-34 Examples of triplets—two cells with three possible values and one cell with two possible values</b>	<b>31</b>
<b>Figure 1-36. Verifying an example of triplets</b>	<b>31</b>
<b>Figure 1-37 Two more instances of triplets—one cell with three possible values and two cells with two possible values</b>	<b>32</b>
<b>Figure 1-38 A partially solved Sudoku puzzle</b>	<b>33</b>
<b>Figure 1-39 The solved puzzle</b>	<b>33</b>
<b>Figure 1-40 The state of the grid after assigning 1 to cell (5,4)</b>	<b>34</b>
<b>Figure 1-41 A deadlock situations for cell (5,5)</b>	<b>35</b>
<b>Figure 1-42</b>	<b>51</b>
<b>Figure 1-43</b>	<b>53</b>
<b>Figure 1-44</b>	<b>55</b>
<b>Figure 1-45 Traditional exam script</b>	<b>66</b>
<b>Figure 1-46 Sample exam script.</b>	<b>68</b>
<b>Figure 1-47 ER-diagram of the system</b>	<b>70</b>
<b>Figure 1-48 System architecture</b>	<b>71</b>
<b>Figure 1-49 Home page</b>	<b>72</b>
<b>Figure 1-50 student registration form</b>	<b>73</b>
<b>Figure 1-51 Student login page</b>	<b>74</b>
<b>Figure 1-52 Exam schedule created for the student.</b>	<b>75</b>
<b>Figure 1-53 Auto checked answer script</b>	<b>76</b>
<b>Figure 1-54 Result record from the database</b>	<b>77</b>
<b>Figure 1-55 All students' records</b>	<b>78</b>

<b>Figure 1-56 Searching record with name</b>	<b>79</b>
---	-----------

### Table Index

<b>List of Tables</b>	<b>Page No</b>
<b>Table 1 Student</b>	<b>69</b>
<b>Table 2 Question</b>	<b>69</b>
<b>Table 3 Marks</b>	<b>69</b>
<b>Table 4 Cumulative Marks</b>	<b>70</b>

# 1. Introduction

## 1.1 E learning

E Learning is an umbrella term that describes learning done at a computer, usually connected to a network, giving us the opportunity to learn almost anytime, anywhere.

E-Learning is not unlike any other form of education - and it is widely accepted that e-Learning can be as rich and as valuable as the classroom experience or even more so. With its unique features e Learning is an experience that leads to comprehension and mastery of new skills and knowledge, just like its traditional counterpart.

Instructional Design for e Learning has been perfected and refined over many years using established teaching principles, with many benefits to students. As a result colleges, universities, businesses, and organizations worldwide now offer their students fully accredited online degree, vocational, and continuing education programs in abundance.

Some other terms frequently interchanged with e Learning include:

- Online learning
- Online education
- Distance education
- Distance learning
- Technology-based training
- Web-based training
- Computer-based training (generally thought of as learning from a CD-ROM)

E Learning is a broad term used to describe learning done at a computer. Most of the time E-learning term frequently interchanged with Distance learning.

### **1.1 Distance learning:**

A method of teaching where students receive training from one or more physically distant (and often distributed) location(s). Students typically use various materials (books, references, CD-ROMs) and media (TV, Internet, and postal mail) to replace direct face-to-face learning.

Distance learning allows you to take a course anytime and anywhere. It gives you the flexibility of studying when you want and where you want (from your home, from your office, while traveling). Distance learning courses teach the very same competencies as courses offered at a campus; the difference is in the delivery. Instead of having an instructor provide a lecture and lead a classroom discussion, distance courses are delivered over the Internet, by using combinations of print, audio and/or videocassette, and correspondence. Most courses require a textbook and a specially developed course packet. Some courses require CD-ROMs, audio and/or videocassettes - what you will need is specified with each course. [22]

### **1.2 E learning worldwide**

The e learning is widely used in most of the developed countries to promote distance education (DE) and life long learning. It can be defined as an innovative approach for delivering electronically mediated, well-designed, learner-centered, and interactive learning environments to anyone, anyplace, anytime by utilizing the internet and digital technologies in concern with instructional design principles (Anonymous 2003, Hedge and Hayward, 2004). Applications and processes of e-learning include web-based learning, computer-based learning, virtual classrooms, and digital collaboration, where contents is delivered *via* the internet, intranet/extranet, audio and/or video tape, satellite TV and CD-ROM (Islam 1997). E learning is now a multi-billion dollar activity worldwide. The rapid and intensive use of ICTs in education in the developed countries facilitated to the establishment of 100% ICT-based universities called

'virtual universities'. In addition, many world-leading conventional universities are now also offering some of their academic courses through various ICTs for their distant learners and established themselves as the 'dual mode universities'. The historic launching of 700 courses from 33 academic disciplines as 'Open courseware's' by <http://ocw.mit.edu/OcwWeb/Global/all-courses.htm> Massachusetts Institute of Technology (MIT) offers a tremendous resource for faculties, students and self-learners around the world. In contrast, the infrastructure of ICTs in the developing or the least developed countries is very weak and thus, intensive use of e-learning in DE is still a dream for their universities and institutes. Recently, ICTs are rapidly expanding in some of the developing countries, and hence, it offers an opportunity to consider the use of ICTs in the promotion of DE. It offers students considerable benefits including increase access to learning, life-long learning opportunities, and convenience of time and place (Pierre 1998).

### **1.3 E-learning in Bangladesh**

Bangladesh is one of the most densely populated countries in the world with nearly 140 million people within an area of 147,570 square kilometers. Its vast population would be the major resources of the country. However, in transforming the potential people into a productive force and ensuring a dynamic environment for social, economic and political development is still a big challenge for its government. Though the literacy rate is officially said to be 66%, but according to private survey the rate is only 42%. Education, therefore, has been recognized as a priority sector by all governments since her independence in 1971. Distance education is an important alternative for educating mass people in Bangladesh for many socio-economic reasons. More importantly, the opportunity for higher education is extremely limited in Bangladesh, and therefore, even students, who can afford to finance their studies, it is very difficult to get admission into the universities due to limited capacity. Dropouts in education from primary to the university level are very high in Bangladesh mainly due to economic and other social reasons. This scenario is well reflecting in very

high enrollment (approx. 400 thousands) of students at BOU, the first and only national distance learning university. Recently, ICTs are rapidly expanding in Bangladesh due to the deregulation of laws and policies by the Government. Therefore, introduction of more e learning or ICTs may boost DE in Bangladesh. [22] [23]

The e learning was first introduced in Bangladesh in 1956 by a radiobroadcasting program, and later expanded much by the establishment of BOU in 1992. However, BOU is still using mostly traditional one-way media and far behind to use modern interactive ICTs in delivering its courses. [22] [23].

#### **1.4 Our contribution in E learning in Bangladesh**

While working with e learning in Bangladesh we came to know that nothing much has done in this sector yet. We can even count the number of web sites regarding e learning in Bangladesh. So we came forward and wanted to contribute in this sector in Bangladesh.

#### **2. Our Target**

Our target is to build a rich web portal where people of all age can visit. They can take advantages of e learning by participating to various exams, quiz, puzzles or interesting things. And all of them will be free.

#### **3. Web portal**

A web portal commonly referred to as simply a portal, a web site or service that offers a broad array of resources and services, such as e-mail, forums and search engines.

As our web portal is an e learning web portal we will offer some learning systems which means in our portal we will include

- Logical section
- Mathematical section
- English section
- Psychology section

But as we know we have very short and limited time to actually implement such a rich web portal, which will run through Internet, for simplicity we have worked with two of the parts of our proposed portal. We worked on Logical and mathematical sections.

And in these two sections we introduced Sudoku as an example of logical section and Kumon method as an example of mathematics section. Going deep into our paper we will have discussions on these two items.

While working on implementing the Sudoku puzzle we found that we may also contribute in generating the puzzle, especially on the algorithm part. So start working with it and we developed a new simple algorithm.

In our paper we talked a lot about Sudoku, its algorithms, our proposed algorithm and analysis on that.

#### **4. Why Sudoku?**

Sudoku is a puzzle games consisting of numbers. Like any game that can take possession of our kids, playing Sudoku has all the elements of a good puzzle game: fun, excitement, and mental challenge. It is simply addictive – in a good and positive sense. However, more than these recreational values that Sudoku offers to its avid gamers, Sudoku also offers to develop more than your child's numerical skills.[15]

#### **4.1 Cognitive Assessment**

Cognitive assessment of the grids and especially the numbers given within the grids, develops in your child the ability and skill to identify numbers in specific columns and to retain these numbers as the need be to arrive at the correct numerical answer.[19]

#### **4.2 Elimination**

By assessing which numbers are already given in particular squares in Sudoku grids, your child can employ the process of eliminating these given numbers to arrive at possible numerical answers.

#### **4.3 Logical Thinking**

With the numbers already given and the grids that have to be filled with numbers that have not yet been eliminated as the possible answers, your child can develop his or her logical thinking skills like deduction and induction.

#### **4.4 Analysis**

To arrive at the possible numerical answers, a child needs to analyze using the process of elimination and on the evaluation of trying out a number if it fits into the grid or not.

### **5. Why Kumon?**

The objective of this thesis is to analysis distance education and to explore a new type of distance education .we presented a new net based learning system for children. In this thesis we tried to explore a new teaching method for Bangladeshi younger generation. Though distance learning is not widely used in Bangladesh but it created a greater interest on the education system those who know about e learning. Kumon is a very popular teaching method in Canada, Australia even in India. But in Bangladesh it is not revealed



yet. We tried to create interest among people about Kumon. So we make it web based to add the Kumon to e learning.

## 6. Sudoku

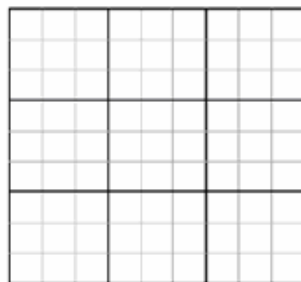
Sudoku is a puzzle game that is taking the world by storm. The name Sudoku comes from the Japanese word (**shown in Figure 1-1**) that means, "Number place." The first Sudoku puzzle was published in the United States, but Sudoku initially became popular in Japan, in 1986, and did not attain international popularity until 2005.[15]

数独

**Figure 1 the word Sudoku in Japanese**

We also can say that, Sudoku is a logic-based number placement puzzle. The objective is to fill a 9x9 grid so that each column, each row, and each of the nine 3x3 boxes (also called blocks or regions) contain the digits from 1 to 9. The puzzle setter provides a partially completed grid.

The rules of Sudoku are extremely simple; yet solving a Sudoku puzzle is an intellectual challenge. A Sudoku puzzle contains a 9x9 grid, which is divided into nine smaller 3x3 grids (known as mini-grids). Figure 1-2 shows a Sudoku grid with its nine mini-grids.



**Figure 1-1 A Sudoku grid**

## 6.1 History

The puzzle was designed by Howard Garns, a retired architect and freelance puzzle constructor, and first published in 1979. Although likely inspired by the Latin square invention of Leonhard Euler, Garns added a third dimension (the regional restriction) to the mathematical construct and (unlike Euler) presented the creation as a puzzle, providing a partially-completed grid and requiring the solver to fill in the rest. The puzzle was first published in New York by the specialist puzzle publisher Dell Magazines in its magazine Dell Pencil Puzzles and Word Games, under the title Number Place (which we can only assume Garns named it).[15]

Nikoli introduced the puzzle in Japan in the paper Monthly Nikolist in April 1984 as *Suuji WA dokushin Ni kagiru*, which can be translated, as "the numbers must be single" or "the numbers must occur only once" (literally means "single; celibate; unmarried").

The image shows a page from a French magazine with a Sudoku puzzle. The puzzle is a 9x9 grid with some numbers filled in. The text on the page is in French and includes a section titled 'DIVERTISSEMENTS QUOTIDIENS' and another titled 'PROGRAMME'. The Sudoku grid is as follows:

8	3	4	1	5	6	9	7	2
6	7	2	9	8	3	5	4	1
9	5	1	3	7	2	8	6	4
2	8	6	4	9	5	3	1	7
7	4	9	6	2	8	1	3	5
5	2	8	7	3	1	4	9	6
3	1	7	5	4	6	2	8	9
4	9	5	2	6	7	3	8	1
1	6	3	8	1	9	7	5	4

In 1989, Loadstar/Softdisk Publishing published DigitHunt on the Commodore 64, which was apparently the first home computer version of Sudoku. At least one publisher still uses that title. [8]

Yoshimitsu Kanai published his computerized puzzle generator under the name Single Number for the Apple Macintosh [11] in 1995 in Japanese and English, and in 1996 for the Palm (PDA). [8]

Bringing the process full-circle, Dell Magazines, which publishes the original Number Place puzzle, now also publishes two Sudoku magazines: Original Sudoku and Extreme Sudoku. Additionally, Kappa reprints Nikoli Sudoku in GAMES Magazine under the name Squared Away; the New York Post, USA Today, The Boston Globe, Washington Post, and San Francisco Chronicle now also publish the puzzle. It is also often included in puzzle anthologies, such as The Giant 1001 Puzzle Book (under the title Nine Numbers).

## 6.2 How to Play

Sudoku requires no calculation or arithmetic skills. It is essentially a game of placing numbers in squares, using very simple rules of logic and deduction. Children and adults can play it and the rules are simple to learn. But this game can help children to build their counting and numbering skills.[10]

The objective of the game is to fill all the blank squares in a game with the correct numbers. There are three very simple constraints to follow. In a 9 by 9 square Sudoku game:

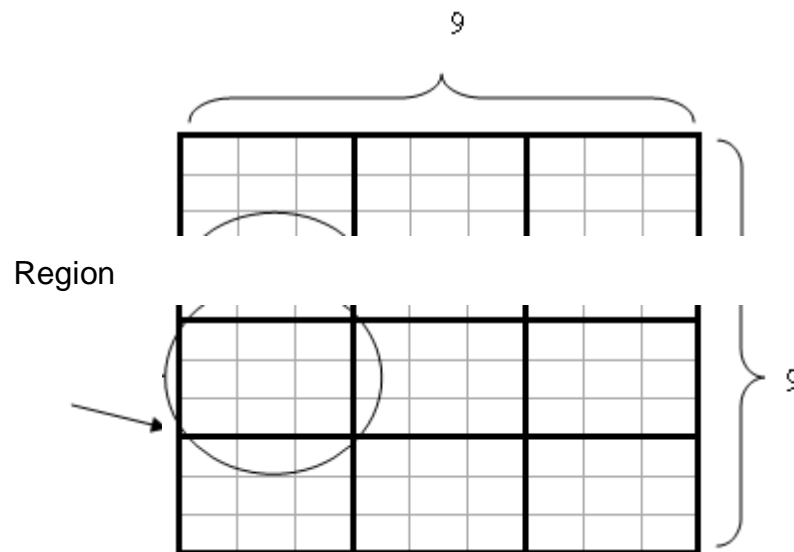
- Every row of 9 numbers must include all digits 1 through 9 in any order
- Every column of 9 numbers must include all digits 1 through 9 in any order
- Every 3 by 3 region of the 9 by 9 square must include all digits 1 through 9

Similarly, smaller Sudoku puzzles, such as the 4x4 puzzle, must have the numerals 1 through 4 in each row, column and subsection. Larger Sudoku

games (16 by 16) must have numerals 1 through 16 in each row, column and region. The principles are the same whatever the size of the game.

Every Sudoku games begin with a number of squares already filled in, and the difficulty of each game is largely a function of how many squares are filled in. The more squares that are known, the easier it is to figure out which numbers go in the open squares. As you fill in squares correctly, options for the remaining squares are narrowed and it becomes easier to fill them in. There are some graphical representations of it, [10] [6]

- Sudoku is played over a 9x9 grid, divided to 3x3 sub grids called "regions":



- Sudoku begins with some of the grid cells already filled with numbers [6] [10]

7			2	5			9	8
		6					1	
			6	1		3		
9					1			
				8		4		9
		7	5		2	8		1
	9	4			3			
				4	9	2	3	
6	1						4	

- The object of Sudoku is to fill the other empty cells with numbers between 1 and 9 (1 number only in each cell) according the following guidelines:[10] [6]

- Number can appear only once on each row:

Allowed 

7			2	5		1	9	8
---	--	--	---	---	--	---	---	---

Not allowed 

7			2	5		8	9	8
---	--	--	---	---	--	---	---	---

- Number can appear only once on each column:

Allowed

9
1
<b>2</b>
3
4

Not allowed

9
1
<b>3</b>
3
4

- Number can appear only once on each region:

Allowed

7		
	<b>3</b>	6

Not allowed

7		
	<b>6</b>	6

### 6.3 Sudoku Techniques:

There are various techniques that we can use to solve a Sudoku puzzle. For a start, we describe the fundamental technique you can use to solve most of the easy Sudoku puzzles. We first walk you through the technique so that you understand how it works, and then we show you the implementation details. [13] [24]

While the technique covered can be used to solve most of the easy Sudoku puzzles, it is not sufficient to solve other, more complex Sudoku puzzles. To help you to accomplish that, we will discuss more advanced techniques also.

### 6.3.1 Elimination Technique

A process of elimination can solve most Sudoku puzzles. For example, if eight out of nine cells in a row are filled, then the remaining cell must be the number that has not been used in the row. In the case of **Figure 1-2**, the value of the remaining cell, (5,1), must be 5, since 1 through 4 and 6 through 9 have already been used in that row. [13] [24]

1	2	3	4		6	7	8	9
1	2	3	4	5	6	7	8	9

**Figure 1-2** *Deriving the number for a cell based on elimination*

When you try to place a number in a cell, examining just its row usually is insufficient, because typically, unlike Figure 1-2, not all the other cells in the row are filled. You have to also scan its column and, if that is not enough, scan within its mini-grid. We call this technique Column, Row, and Mini-grid Elimination (CRME). [13]

### 6.3.2 Column, Row, and Mini-grid Elimination

To see how CRME works, let's start off with the simplest scenario. **Figure 1-3** shows a partially filled Sudoku puzzle. [13] [24]

4		3	7	8	5	1	2	6
2								
1								
	5	8						
6	1	4						
7		9						
9								
5								

**Figure 1-3** *Partially filled Sudoku puzzle[13]*

Scanning each cell from left to right, top to bottom, the first empty cell you encounter is (2,1). Examining the column that it is in tells you that the possible values left for this cell are 2, 3, 4, 6, 7, 8, and 9 (see **Figure 1-4**). [13] [24]

4		3	7	8	5	1	2	6
2								
1								
	5	8						
6	1	4						
7		9						
9								
5	↓							

**Figure 1-4 Scanning by column[13]**

However, scanning horizontally across the row reduces the number of possible values to one, which is 9, because all the other values have already been used, as shown in **Figure 1-5**.

4		3	7	8	5	1	2	6
2								
1								
	5	8						
6	1	4						
7		9						
9								
5	↓							

**Figure 1-5 Scanning by row [24]**

Since the only possible value is 9, you can now fill in (2,1) with 9 (**Figure 1-6**).



4	9	3	7	8	5	1	2	6
2								
1								
	5	8						
6	1	4						
7		9						
9								
5								

**Figure 1-6 Filling in the value for (2,1) [24]**

Continuing with the scanning, the next empty cell is (2,2). Scanning by its column and row, as shown in **Figure 1-7**, yields the possible values of 3, 4, 6, 7, and 8. [13] [24]

4	9	3	7	8	5	1	2	6
2								
1								
	5	8						
6	1	4						
7		9						
9								
5								

**Figure 1-7 Scanning by column and row [13]**

By scanning the mini-grid next (see **Figure 1-8**), you see that it already has the values of 1, 2, 3, 4, and 9, so the possible values are now reduced to 6, 7, and 8. Because (2, 2) has more than one possible value remaining after we have searched the column, row, and mini-grid to eliminate values, the answer is not conclusive. But at least we now know that only the values 6, 7, and 8 are possibilities for (2, 2). [13] [24]

4	9	3	7	8	5	1	2	6
2								
1								
	5	8						
6	1	4						
7		9						
9								
5								

**Figure 1-8 Scanning within the mini-grid**

Continuing with the scan, the next interesting cell is (1,4), as shown in **Figure 1-9**. [24]

4	9	3	7	8	5	1	2	6
2								
1								
	5	8						
6	1	4						
7		9						
9								
5								

**Figure 1-9 Examining cells (1,4)**

Scanning its column yields 3 and 8 as possible values, but scanning its row confirms that the number is 3, and so you can now fill in (1, 4) with 3, as shown in **Figure 1-10**. [13] [24]

4	9	3	7	8	5	1	2	6
2								
1								
3	5	8						
6	1	4						
7		9						
9								
5								

**Figure 1-10 Filling in (1, 4) with the value 3**

Continuing with the scan, the next conclusive cell is (2,6). Scanning by column and row does not yield a specific number, but scanning its mini-grid confirms that the missing number is 2 (see **Figure 1-11**).[13] [24]

4	9	3	7	8	5	1	2	6
2								
1								
3	5	8						
6	1	4						
7	2	9						
9								
5	↓							

**Figure 1-11** *Confirming the value for (2,6) by scanning column, row, and mini-grid*

This particular cell is worth noting because it illustrates that numbers confirmed in earlier scans (cell (1,4) in this example) can often help in confirming other cells. If you had not previously filled in the number for (1,4), then it would not be possible to confirm the number for (2,6). Finally, the last cell that you can confirm is (1, 7), which you can confirm by simply performing a column scan (see **Figure 1-12**).[13] [24]

4	9	3	7	8	5	1	2	6
2								
1								
3	5	8						
6	1	4						
7	2	9						
8								
9								
5	↓							

**Figure 1-12** *Filling in the value for (1,7)*

### 6.3.3 Intermediate Techniques

We already talk about how to use the CRME technique to solve some simple Sudoku puzzles. However, there are more-challenging Sudoku puzzles that require much more analysis and new techniques to solve, and that's where the limitations of the CRME technique become apparent. We try to continue our discovery of new techniques so that we can tackle the more-challenging Sudoku puzzles. Now we will be looking for some not-so-obvious patterns in Sudoku puzzles and analyzing how we can exploit those patterns to take us one step closer to solving tough Sudoku puzzles. [13] [24]

#### 6.3.3.1 Lone Rangers

Lone ranger is a term that we use to refer to a number that is one of multiple possible values for a cell but appears only once in a row, column, or mini-grid. To see what this means in practice, consider the row shown in **Figure 1-13**. In this row, six cells have already been filled in, leaving three unsolved cells (shown as shaded cells) with their possible values written in them (derived after applying the CRME technique). Notice that the second cell is the only cell that contains the possible value 8. Since no other cells in this row can possibly contain the value 8, this cell can now be confirmed with the value 8. In this case, the 8 is known as a lone ranger. [13] [24]

1	348	5	9	6	7	2	34	34
---	-----	---	---	---	---	---	----	----

**Figure 1-13 Identifying a lone ranger in a row** [13] [24]

Lone rangers are extremely useful in helping to confirm the number for a cell and are often useful in more complex Sudoku puzzles. Lone rangers can appear in a row, column, or mini-grid. Let's see how you can use lone rangers to solve your Sudoku puzzles.

### 6.3.3.2 Lone Rangers in a Mini-grid

Consider the grid shown in **Figure 1-14**.

6		7			1			
	8		7				6	9
	3	1		4	5	2		7

**Figure 1-14** A partial Sudoku puzzle[13] [24]

Using the CRME technique, we can confirm the values of only three cells, as shown in **Figure 1-15**.

6		7			1			
	8		7				6	9
9	3	1	6	4	5	2	8	7

**Figure 1-15** Confirming the values of three cells

We definitely can do better. As a start, let's examine the possible values for all other cells. **Figure 1-16** shows the possible values after partially applying the CRME technique. [13] [24]

6	245	7	2389	2389	1	345	345	345
245	8	245	7	23	23	1345	6	9
9	3	1	6	4	5	2	8	7

**Figure 1-16 Possible values for the cells after applying the CRME technique**

One interesting observation is found by looking at the third mini-grid, shown in **Figure 1-17**. [13] [24]

	7	8	9
1	345	345	345
2	1345	6	9
3	2	8	7

**Figure 1-17 Examining the third mini-grid**[13] [24]

If you observe cell (7, 2), one of the possible values is 1, along with the other numbers like 3, 4, and 5. However, the number 1 appears as a possible value only for (7,2) and not for the other cells within the mini-grid. Logically, we can now conclude that as long as a number appears only once (as a possible value) within the mini-grid, that number can be confirmed as the number for the cell. This is logical, because cells (7, 1), (8,1), and (9,1) cannot contain the value 1, and hence only (7,2) can contain 1. Following this argument, we can now put a 1 in (7, 2), as shown in **Figure 1-18**. [13] [24]

6		7			1			
	8		7			1	6	9
9	3	1	6	4	5	2	8	7

**Figure 1-18** Confirming the value for (7,2) [13] [24]

### 6.3.3.3 Lone Rangers in a Row

Lone rangers do not just occur within mini-grids; sometimes they occur within rows.

Consider the puzzle shown in **Figure 1-19**.

5	8	3		2			9
			5	9	3	2	
4	9	2					
		5	7			1	
	6	7				5	3
		8		5	9		
					5	1	2
		9	2		6		
2	5			8		6	7

**Figure 1-19** Searching a Sudoku puzzle for lone rangers in rows

Scanning for lone rangers in the mini-grids does not get you anywhere, but if you look at row 5 (see **Figure 1-20**), you will see that there is a lone ranger in cell (6, 5). [13] [24]

And that effectively confirms (6, 5) with the value 2.

39	234	5	7	346	248	1	248	4689
19	6	7	148	14	1218	5	3	489
13	1234	8	1346	5	9	467	247	46
3678	347	46	349	347	5	3489	1	2

**Figure 1-20 Lone ranger detected in row 5**[13] [24]

#### 6.3.3.4 Lone Rangers in a Column

Similar to lone rangers in rows, lone rangers also exist in columns. Consider the example in **Figure 1-21**.

1	5	8	6				2
	7	2		1		8	6
6			8	2			
	8						
		1	9	2			
	2					3	
8			3	1		2	4
4	1	3		5			7
2				4		1	

**Figure 1-21 Searching a Sudoku puzzle for lone rangers in columns**

If you look at column 8, you will notice that cell (8, 5) contains a lone ranger, 8 (see **Figure 1-22**). And that confirms (8, 5) to be 8. [13] [24]



79	3479	49	2
59	8	459	6
79	134579	459	13579
567	145679	4569	1579
578	2	4568	578
78	145679	3	15789
1	569	2	4
89	69	7	89
1	3569	1	3589

**Figure 1-22 Finding a lone ranger in a columns**[13] [24]

Once (8, 5) is confirmed, you can now apply the CRME technique again. This removes the number 8 from the list of possible values at cells (6, 5), (9, 5), and (9, 6). [13] [24]

And now you will discover yet another lone ranger in the same column, as pointed out in **Figure 1-23**. And that effectively confirms (8, 4) as a 6. [13] [24]

19	3479	49	2
29	8	459	6
79	134579	459	13579
67	145679	4569	1579
67	2	8	57
78	145679	3	1579
1	569	2	4
89	69	7	89
1	3569	1	3589

**Figure 1-23 Confirming yet another cells**

### 6.3.4 Advanced Techniques

So far, we have talked about a couple of techniques that have proven invaluable in solving quite a few Sudoku puzzles. However, some difficult puzzles exist that just refuse to surrender to CRME or lone rangers. Now we try to describe three additional techniques that we can use to solve some difficult puzzles: looking for twins, looking for triplets, and, lastly, brute-force elimination, a technique of last resort if everything else fails. [13] [24]

#### 6.3.4.1 Looking for Twins

To understand the usefulness of looking for twins, consider the partial Sudoku puzzle shown in **Figure 1-24**, which includes lists of possible values for

unresolved cells.

6	245	7	2389	2389	1	345	345	345
245	8	245	7	23	23	1	6	9
9	3	1	6	4	5	2	8	7
1	24	6	5	23	7	8	9	234

**Figure 1-24** A partially solved Sudoku puzzle with the possible values for empty cells [13] [24]

Observe the two cells (5,2) and (6,2) in **Figure 1-25**. They both contain possible values of 2 and 3. In this scenario, if (5,2) takes the value 2, then (6,2) must take the value 3. Conversely, if (6,2) takes the value 2, then (5,2) must take 3. All other cells in row 2 besides these two cells cannot contain either 2 or 3. Because the two cells (5,2) and (6,2) have identical lists of possible values and are in the same row, they are known as twins. [13] [24]

6	245	7	2389	2389	1	345	345	345
245	8	245	7	23	23	1	6	9
9	3	1	6	4	5	2	8	7
1	24	6	5	23	7	8	9	234

**Figure 1-25** identifying the twins [13] [24]

Scanning across the row, you now can eliminate 2 and 3 as possible values for any of the other cells, as shown in **Figure 1-26**, in which 2 has been

deleted as a possible value for cells (1,2) and (3,2). [13] [24]

6	245	7	2389	2389	1	345	345	345
<del>2</del> 45	8	<del>3</del> 45	7	23	23	1	6	9
9	3	1	6	4	5	2	8	7
1	24	6	5	23	7	8	9	234

**Figure 1-26 Eliminating 2 and 3 as possible values for other cells in the same row as the twins**

Similarly, because the twins appear in the same mini-grid, all the other cells in this. [13] [24]

Mini-grid cannot possibly take the values 2 and 3. Thus, you can eliminate 2 and 3 as possible values for cells (4,1) and (5,1), as shown in **Figure 1-28**.

6	245	7	<del>2</del> 89	<del>3</del> 89	1	345	345	345
<del>2</del> 45	8	<del>3</del> 45	7	23	23	1	6	9
9	3	1	6	4	5	2	8	7
1	24	6	5	23	7	8	9	234

**Figure 1-27 Eliminating 2 and 3 as possible values for other cells in the same mini-grid as the twins** [13] [24]

If you noticed, there are now three pairs of twins in the grid, as identified in **Figure 1-28**. Two pairs are in the first two rows and the third pair is in column 5.

6	245	7	89	89	1	345	345	345
45	8	45	7	23	23	1	6	9
9	3	1	6	4	5	2	8	7
1	24	6	5	23	7	8	9	234

**Figure 1-28 New pairs of twins emerging after the first scanning** [13]  
[24]

For the twins “23,” you can scan the column they are in and remove all occurrences of 2 and 3. For example, if we examine the possible values for column 5, we can see that the values 2 and 3 can be eliminated from many of the cells in this column. [13] [24]

For the twins “45,” scanning by their mini-grid allows you to remove the 4 and 5 in cell (2, 1), as shown in **Figure 1-29**. Doing so causes cell (2,1) to be confirmed with 2, which in turn causes (2,4) to be confirmed with 4. [13] [24]

6	2	7	89	89	1	345	345	345
45	8	45	7	23	23	1	6	9
9	3	1	6	4	5	2	8	7
1	4	6	5	23	7	8	9	23
				1689				

**Figure 1-29 Confirming cell (2,1) and subsequently (2,4)**

For the twins “89,” there isn’t much you can do to the row and mini-grid that they are in. This leaves the grid as shown in **Figure 1-30** as you can see, scanning for twins in rows, columns, and mini-grids leaves us better off than before the scanning. In our example, two cells get confirmed in the process. [13] [24]

6	2	7	89	89	1	345	345	345
45	8	45	7	23	23	1	6	9
9	3	1	6	4	5	2	8	7
1	4	6	5	23	7	8	9	23
				1689				

**Figure 1-30** the grid after scanning for twins in the rows, columns, and mini-grid

#### 6.3.4.2 Looking for Triplets

While twins can move you closer to solving a Sudoku puzzle, occasionally you will also come across triplets. Consider the partial Sudoku puzzle shown in **Figure 1-31**. [13] [24]

<b>1</b>	<b>2467 89</b>	<b>2467 89</b>	
<b>246</b>	<b>3</b>	<b>2467 89</b>	
<b>246</b>	<b>246</b>	<b>5</b>	

**Figure 1-31 A partial Sudoku puzzle**

There are no twins in the mini-grid, but you can spot three cells with the same possible values. I call these triplets. Like twins, triplets are useful in further eliminating possible values for other cells. In the puzzle shown, the numbers 2, 4, and 6 must definitely be placed in one of the three cells containing 2, 4, and 6 as the possible values. Once this reasoning is established, you can cross out the 2, 4, and 6 from the other cells in the mini-grid. The grid now looks like **Figure 1-32**. [13] [24]

<b>1</b>	<b>246 789</b>	<b>246 789</b>	
<b>246</b>	<b>3</b>	<b>246 789</b>	
<b>246</b>	<b>246</b>	<b>5</b>	

**Figure 1-32. Modifying the mini-grid where the triplets are found**

Although in this example, identifying the triplets did not result in confirming any of the cells, this technique is useful for further reducing the possibilities for other cells, which in turn may be confirmed by some other techniques such as CRME and lone rangers. [13] [24]

### 6.4.2.1 Variants of Triplets

In the last section, you saw that triplets are made up of three cells, with each containing the same three possible values. However, this definition is not always strictly adhered to. There are three different scenarios that can be

#### Classified as triplet

- **Scenario 1:** Three cells with the same three possible values.
- **Scenario 2:** Two cells with three possible values and one cell containing two possible values that are a subset of the three possible values.
- **Scenario 3:** One cell with three possible values and two cells containing two possible values that are a subset of the three possible values.

It is important to identify the three scenarios, because you have to write code to look out for the various variants of triplets. To understand the three different scenarios, let's consider the following examples.

- **Scenario 1**

The first scenario is the most obvious. As long as three cells have an identical set of three possible values, they are deemed to be triplets, as shown in **Figure 1-33**. You can then eliminate the triplets as possible values from the rest of the cells.



<b>123</b>	<b>123</b>	<b>123</b>
<b>X</b>	<b>X</b>	<b>X</b>
<b>X</b>	<b>X</b>	<b>X</b>

**Figure 1-34 A set of triplets**

- **Scenario 2**

The second scenario is less obvious: two cells with three possible values and one cell containing two possible values that are a subset of the three possible values. Three examples are shown in **Figure 1-34**. [13] [24]

<b>123</b>	<b>123</b>	<b>23</b>
<b>X</b>	<b>X</b>	<b>X</b>
<b>X</b>	<b>X</b>	<b>X</b>

<b>123</b>	<b>123</b>	<b>13</b>
<b>X</b>	<b>X</b>	<b>X</b>
<b>X</b>	<b>X</b>	<b>X</b>

<b>123</b>	<b>123</b>	<b>12</b>
<b>X</b>	<b>X</b>	<b>X</b>
<b>X</b>	<b>X</b>	<b>X</b>

**Figure 1-34 Examples of triplets—two cells with three possible values and one cell with two possible values**

To clarify why the examples in **Figure 1-35** can be classified as triplets, let's walk through another example. Starting with the first (leftmost) example in **Figure 1-35**, assume that the third cell now takes the value 2. That causes the first two cells to become twins, with possible values 1 and 3, as shown in **Figure 1-36**. Based on the earlier discussion on twins, the first two cells must assume one of the other possible values. [13] [24]

<b>123</b>	<b>123</b>	<b>23</b>	<b>13</b>	<b>13</b>	<b>2</b>
<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>

**Figure 1-36. Verifying an example of triplets**

Effectively, this ensures that the three cells must contain one of the three possible values: 1, 2, or 3. All other cells that contain 1, 2, or 3 as possible values must now be eliminated from their list of possible values.

- **Scenario 3**

The third scenario is even less obvious than the second scenario. **Figure 1-37** shows two such examples.

<b>123</b>	<b>13</b>	<b>12</b>	<b>123</b>	<b>13</b>	<b>23</b>
<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>

**Figure 1-37 two more instances of triplets—one cell with three possible values and two cells with two possible values**

### 6.3.4.3 Brute-Force Elimination

Up to this point, you should be able to solve most Sudoku puzzles using the techniques described. However, sometimes (especially for difficult puzzles) all the techniques seem to be useless. In such cases, you really need to make an

educated guess and put a value in a cell and then apply all the techniques covered so far. We call this technique **brute-force elimination**. [13] [24]

Consider the partially solved Sudoku puzzle shown in **Figure 1-38**. Applying all the other techniques could not solve the puzzle. [13] [24]

		6		7	4	1	2	
		2		8	1	4	6	
1	9	4	2	6	3	8	7	5
6	2	9	3		5	7		
	5	7			6	3	9	
3		1	7	9		6	5	
9	1	3	6	5	7	2		
		5				9	3	6
2	6	8	4	3	9	5	1	7

**Figure 1-38** A partially solved Sudoku puzzle

The most natural way to solve the puzzle would be to do some guesswork. You can select a value for an unsolved cell and apply the earlier techniques to see if that will solve the puzzle. If that doesn't solve the puzzle, select a value for the next unsolved cell and repeat the same process until the puzzle is solved. [13] [24]

Now, the question is: which cell do we start with? Well, quite obviously you should start with the cell with the least number of possible values. Scanning from left to right, top to bottom, you can see that cell (1,1) is the first choice. In (1,1) there are two possible values, 5 and 8. You can choose either 5 or 8, but for simplicity you can always start with the first number. So let's choose 5 for (1,1) and then proceed to solve the puzzle using the techniques we have discussed. Voila! The puzzle is solved, as shown in **Figure 1-39**. [13] [24]

5	8	6	9	7	4	1	2	3
7	3	2	5	8	1	4	6	9
1	9	4	2	6	3	8	7	5
6	2	9	3	4	5	7	8	1
8	5	7	1	2	6	3	9	4
3	4	1	7	9	8	6	5	2
9	1	3	6	5	7	2	4	8
4	7	5	8	1	2	9	3	6
2	6	8	4	3	9	5	1	7

**Figure 1-39 The solved puzzle**

In this example, we are pretty lucky. Simply selecting a cell and assigning a value to it solved the entire puzzle. However, sometimes you will need to select a number, run through the techniques, and then select another cell and run through the same process again. For difficult puzzles, you may need to repeat this several times. [13] [24]

Sometimes, you may just make the wrong decisions and cause the puzzle to be unsolvable. Using the same example, suppose that the first cell we select is not (1, 1) but (5, 4), with possible values of 1 and 4. Assuming that we selected 1 for (5, 4), run the grid through all the various techniques. **Figure 1-40** shows the result of the grid after assigning 1 to cell (5, 4) and then using the other techniques to derive other cells. [13] [24]

		6		7	4	1	2	
		2		8	1	4	6	
1	9	4	2	6	3	8	7	5
6	2	9	3	1	5	7		
4	5	7	8		6	3	9	
3		1	7	9	2	6	5	
9	1	3	6	5	7	2		
		5	1	2	8	9	3	6
2	6	8	4	3	9	5	1	7

**Figure 1-40 the state of the grid after assigning 1 to cell (5,4)**

At this stage, an error will occur. If you apply the CRME technique to cell (5, 5), you will realize that cell (5, 5) have no possible values. Scanning its row and column, all the numbers from 1 to 9 have already been used, as shown in **Figure 1-41**. [13] [24]

		6		7	4	1	2		
		2		8	1	4	6		
1	9	4	2	6	3	8	7	5	
6	2	9	3	1	5	7			
4	5	7	8		6	3	9		
3		1	7	9	2	6	5		
9	1	3	6	5	7	2			
		5	1	2	8	9	3	6	
2	6	8	4	3	9	5	1	7	

**Figure 1-41 A deadlock situations for cell (5,5)**

In this case, you need to backtrack. You need to backtrack to where you have previously guessed a value. In this case, you need to backtrack to cell (5,4) and, in the process, erase all the cells you have confirmed based on assigning 1 to (5,4). Now, instead of assigning 1 to (5,4), try the next number, 4. This time around, if you apply all the techniques you have learned, you will solve the puzzle. [13] [24]

Based on the description of the brute-force elimination technique, observe the following:

- The brute-force elimination technique can be implemented programmatically using recursion. To allow for moves to backtrack, you need to “remember” the state of the grid before assigning a value to a cell, which enables you to restore the grid to its previous state. [13] [24]
- Selecting the right number to assign to a cell is important. In the example, selecting a 1 for (5, 4) causes the puzzle to have no solution, but selecting 4 for (5, 4) solves the puzzle. To solve a puzzle, you can always start from the first possible number and work your way toward the solution. To minimize the

possibility of backtracks; always select the cell with the least number of possible values when applying the brute-force technique. [13] [24]

- Even though we call this technique brute-force elimination, solving the puzzle using this technique does not imply that we are solving the entire puzzle by guesswork. In fact, for most puzzles, you need to apply the brute-force elimination technique only a few times, and then you can solve the rest of the grid by other logical techniques. [13] [24]

#### 6.4 Available algorithms

Sudoku puzzles are often ranked by difficulty. Perhaps surprisingly, the number of givens has little or no bearing on a puzzle's difficulty. A puzzle with a minimum number of givens may be very easy to solve, and a Sudoku with more than the average number of givens can still be extremely difficult to solve.

Computer solvers can estimate the difficulty for a human to find the solution, based on the complexity of the solving techniques required.

Some initial layouts of Sudoku may have more than one solution and even no solution, but such are not considered proper Sudoku puzzles; like in most other pure-logic puzzles, a unique solution is expected.

***The algorithm of solving any Sudoku puzzle consists of 3 processes [34]:***

- **Scanning of rows/columns and regions** looking for missed numbers to fill in by a process of elimination. Scanning comes to a halt when no further numbers can be discovered. At this point you should turn on your logical thinking.
- **Marking up** suspected candidate numbers in the blank cells. There are two popular methods: subscripts and dots. In the subscript notation the candidate numbers are written in subscript in the cells. The second notation is a pattern of dots with a dot in the top left hand corner representing a 1 and a dot in the bottom right hand corner representing a 9.

An alternative technique that some find easier is to mark up those numbers that a cell cannot have.

- **Analyzing**

Here you eliminate candidates and make predictions "what-if". In elimination, progress is made by successively eliminating candidate numbers from one or more cells to leave just one choice. After each answer has been achieved, another scan may be performed

Starting with a matrix of 81 squares each with 9 'possible' values

As you fill in a new number, remove all the possible in (a) the same row, (b) the same column and (c) the same box. [40]

- **First Stage:** Work through each square in turn and see if it only has one possibility. If so, we've solved that square, so fill it in and remove all associated possibilities. Repeat this until you get no more changes.

This first stage solves most easy and medium problems outright.

- **2nd Stage:** Now work through each row, column and box in turn and check all remaining values 1..9 in turn in each unsolved square of the set. See if we can find a square that can only have one possible value. This lets us fill in some more and then we repeat with stage 1 again.

This second stage solves most hard ones.

- **3rd Stage:** If we've not solved it by now, we've finished the simple deterministic process and we now have to guess one of the possible. Find the square with the least number of possible and pick one of them. This gets tricky because we now have to remember all we fill in from now on so we can delete them all if we're wrong. Having guessed one square, repeat stage 1 and 2 again. [40]

If our guess solves it, we're done. If not, delete all the guesses and pick the next guess. Repeat until we solve it.

**There are three general approaches taken in the creation of serious Sudoku-solving programs[35]:**

Human solving methods, rapid-style methods, and pure brute-force algorithms.[35]

**Human-style** [35] solvers will typically operate by maintaining a mark-up matrix, and search for contingencies, matched cells, and other elements that a human solver can utilize in order to determine and exclude cell values.

Many rapid-style solvers employ backtracking searches, with various pruning techniques also being used in order to help reduce the size of the search tree. The term rapid-style may be misleading: Most human-style solvers run considerably faster than a rapid-style solver, although the latter takes less time to write and is more easily adapted to larger grids. A purely brute-force algorithm is very simple and finds a solution to a puzzle essentially by "counting" upward until a string of 81 digits is constructed which satisfies the row, column, and box constraints of the puzzle.

**Rapid solvers** [35] are preferred for trial-and-error puzzle-creation algorithms, which allow for testing large numbers of partial problems for validity in a short time; human-style solvers can be employed by hand-crafting puzzle smiths for their ability to rate the challenge of a created puzzle and show the actual solving process their target audience can be expected to follow.

Although typical Sudoku puzzles (with 9×9 grid and 3×3 regions) can be solved quickly by computer, the generalization to larger grids is known to be NP-complete. Various optimization methods have been proposed for large grids.

#### ***6.4.1 Sudoku Generation Algorithm [36]***

How can you generate Sudoku puzzles? Here is what has done in simple little python program.

First, we need a Sudoku solver. My solver uses three simple strategies to solve a board; they are simple to implement and seem fast enough.

If only one number fits in a square without row, column, and box conflicts, we fill it in.



If a number needed by a row, column, or box can only go in one square in that row, column, or box, we fill it in.

If we can't fill in anything using rules 1 or 2, then we find a most-constrained place or number where we can guess (for example, two choices is better than three), and we try all the guesses.

The last bit about trying all the guesses will require some backtracking. When we are selecting a place to guess, we choose one randomly among the most-constrained places, and we shuffle the choices to try them in a random order.

The result is if we tell the solver to solve an empty board, we get a nice random fully-solved Sudoku board.

To generate Sudoku puzzles, we start with a solved board, and we choose some minimal hints to reveal as follows.

Going through the squares in shuffled order, reveal each square only if it is not deduced by the other revealed squares (using the solver without guessing). This produces a list of about 30 to 40 hints which together determine the 81 squares of the board.

Going through the chosen hints in shuffled order, attempt dropping each one, and fully solve the board far enough to find two solutions if there are two. Replace the hint unless the solution is still unique with the hint removed.

Notice that generating a minimal puzzle this way requires us to do the work of solving a Sudoku board about twenty times, so it takes a lot more work to generate a minimal Sudoku puzzle than it does to solve one. If there are ways of generating Sudoku puzzles more quickly.

#### **6.4.2 Solving Sudoku by backtracking [37]**

The basic backtracking algorithm can be adapted to solve Sudoku. This is straightforward. Say a zone is a subset of  $N$  boxes of an  $N \times N$  grid, which must contain the numbers from 1 to  $N$ . A standard Sudoku contains 27 zones, namely 9 rows, 9 columns and 9 squares that are  $3 \times 3$ . In a jigsaw Sudoku, zones having irregular boundaries, like a jigsaw piece, replace the square zones. One

possible algorithm that uses backtracking to solve such Sudoku constructs a graph on  $N^2$  vertices, one vertex for each box of the grid. Two vertices are connected by an edge if there exists a zone containing the two boxes. The problem is then equivalent to coloring this graph with  $N$  colors, where adjacent vertices may not have the same color. This is done by starting with an empty assignment of colors (for jigsaw Sudoku) and assigning colors to vertices one after another, using some fixed order of the vertices. Whenever a color is being assigned, we check whether it is compatible with the existing assignments, i.e. whether the new color occurs among the neighbors of that vertex. If it doesn't, then we may assign it to the vertex and try to process another vertex. We backtrack once all  $N$  colors have been tried for a given vertex. If all vertices have been assigned a color, then we have found a solution. There are of course much more sophisticated algorithms to solve graph coloring. If the Sudoku contains initial data, i.e. some boxes have already been filled, and then these go into the color assignment before backtracking begins and the vertex sequence includes only the empty boxes.

The above algorithm was used to solve a 10x10 jigsaw Sudoku that was proposed on [Les-Mathematiques.net](http://Les-Mathematiques.net) a link to the proposal may be found in the section for external links. The first section of the program defines the 10 jigsaw pieces (zones), the second the row and column zones. Thereafter the graph is constructed as an adjacency list. The search procedure prints completed solutions (when all 100 boxes have been assigned). Otherwise it computes the set of colors present among the neighbors of the next vertex to be processed, and recursively tries those assignments that do not conflict with this set. The search starts with an empty assignment.

The data structures used during the backtracking search are chosen to make this easy and fast, although further optimization is possible. The search state is stored in three data structures: a hash table whose keys are the vertices and whose values are the colors that have been assigned to them. There is an array that contains the vertices that have not yet been assigned a color. Finally, there is a hash table whose keys are again the vertices and whose values are

hash tables containing the colors present among the neighbors of the respective vertex, as well as a hint as to who assigned them.

### **6.4.3 Solving Sudoku by a brute-force algorithm [35]**

Some hobbyists have developed computer programs that will solve Sudoku puzzles using a brute force algorithm. Although it has been established that approximately  $6.67 \times 10^{21}$  final grids exist, using a brute force algorithm can be a practical method to solve puzzles using a computer program if the code is well designed.

An advantage of this method is that if the puzzle is valid, a solution is guaranteed. There is not a strong relation between the solving time and the degree of difficulty of the puzzle; generating a solution is just a matter of waiting until the algorithm advances to the set of numbers that satisfies the puzzle. The disadvantage of this method is that it may be comparatively slow when compared to computer solution methods modeled after human-style deductive methods.

Briefly, a brute force program would solve a puzzle by placing the digit "1" in the first cell and checking if it is allowed to be there. If there are no violations (checking row, column, and box constraints) then the algorithm advances to the next cell, and places a "1" in that cell. When checking for violations, it is discovered that the "1" is not allowed, so the value is advanced to a "2". If a cell is discovered where none of the 9 digits is allowed, then the algorithm leaves that cell blank and moves back to the previous cell. The value in that cell is then incremented by one. The algorithm is repeated until the allowed value in the 81st cell is discovered. The construction of 81 numbers is parsed to form the 9 x 9 solution matrix.

"Near worst case" Sudoku puzzle for brute force solver. Solving this puzzle by brute-force requires a large number of iterations because it has a low number of clues (17), the top row has no clues at all, and the solution has "987654321" as its first row. Thus a brute-force solver will spend an enormous amount of time "counting" upward before it arrives at the final grid, which satisfies the puzzle. If one iteration is defined as one attempt to place one value in one cell, then this puzzle requires 641,580,843 iterations to solve. These iterations do

not include the work involved at each step to learn if each digit entered is valid or not (required for every iteration).

#### **6.4.4 Solving Sudoku via Stochastic Search / Optimization methods.[37]**

Recently, some researchers have also shown how Sudoku can be solved using stochastic -- i.e. random-based -- search. Perhaps the first algorithm of this kind was produced by Rhyd Lewis in Lewis, R (2007) Metaheuristic Can Solve Sudoku Puzzles Journal of Heuristics, vol. 13 (4), pp 387-401.

This particular algorithm starts by randomly assigning numbers to the blank cells in the grid in a haphazard fashion. The number of errors in the grid is then calculated and the objective of the algorithm is to then ``shuffle`` the numbers around the grid until the number of mistakes has been reduced to zero. A solution to the puzzle will then have been found. In this algorithm, the optimization is achieved using simulated annealing, though other optimization methods such as tabu search may prove just as applicable.

The advantage of this type of method is that the puzzle does not have to be ``logic-solvable`` in order for the algorithm to be able solve it. In other words, unlike other methods, the puzzles that are given to this algorithm do not have to be specially constructed so that they provide sufficient clues for filling the grid using forward chaining logic only. In fact, the only pre-requisite for the stochastic search algorithm to work is that puzzle has at least one solution.

The simulated annealing method in Lewis's paper is also quite fast (though perhaps not as quick as some logic-based techniques with logic solvable puzzles): depending on the type of instance given to the algorithm, generally 9x9 puzzles will be solved in less than 1 second on a typical year-2000 lap top; 16x16 puzzles will take around 10-15 seconds.

#### **6.5 Complexity analysis**

The general problem of solving Sudoku puzzles on  $n^2 \times n^2$  boards of  $n \times n$  blocks is known to be NP-complete. This gives some vague indication of why Sudoku is hard to solve, but on boards of finite size the problem is finite and can be solved by a deterministic finite automaton that knows the entire game tree.

---

However, for a non-trivial starting board, the game tree is very large and so this method is not feasible. The problem of solving a puzzle that is known to have only one solution is in UP.

Solving Sudoku puzzles can be expressed as a graph-coloring problem. The aim of the puzzle in its standard form is to construct a proper 9-colouring of a particular graph, given a partial 9-colouring. The graph in question has 81 vertices, one vertex for each cell of the grid.

The vertices can be labeled with the ordered pairs  $(x, y)$ , where  $x$  and  $y$  are integers between 1 and 9. In this case, two distinct vertices labeled by [39]

---


$$(x, y) \text{ And } (x', y') \text{ are joined by an edge if and only if:}$$

$$x = x' \text{ Or,}$$

$$y = y' \text{ Or,}$$

$$\lceil x/3 \rceil = \lceil x'/3 \rceil \text{ And } \lceil y/3 \rceil = \lceil y'/3 \rceil$$

Assigning an integer between 1 and 9 to each vertex, in such a way that vertices that are joined by an edge do not have the same integer assigned to them then completes the puzzle.

A valid Sudoku solution grid is also a Latin square. There are significantly fewer valid Sudoku solution grids than Latin squares because Sudoku imposes the additional regional constraint. Nonetheless, Bertram Felgenhauer calculated the number of valid Sudoku solution grids for the standard 9x9 grid in 2005 to be 6,670,903,752,021,072,936,960, which is roughly the number of micrometers to the nearest star. This number is equal to  $9! \times 722 \times 27 \times 27,704,267,971$ , the last factor of which is a prime. The result was derived through logic and brute force computation. The derivation of this result was considerably simplified by analysis provided by Frazer Jarvis and Ed Russell has confirmed the figure independently. Russell and Jarvis also showed that when symmetries were taken into account, there were 5,472,730,538 solutions. The number of valid Sudoku solution grids for the 16x16 derivation is not known.

The maximum number of givens that can be provided while still not rendering the solution unique, regardless of variation, is four short of a full grid; if two instances of two numbers each are missing and the cells they are to occupy are the corners of an orthogonal rectangle, and exactly two of these cells are within one region, there are two ways the numbers can be added. The inverse of this — the fewest givens that render a solution unique — is an unsolved problem, although the lowest number yet found for the standard variation without a symmetry constraint is 17, a number of which have been found by Japanese puzzle enthusiasts, and 18 with the givens in rotationally symmetric cells.

### ***6.5.1 Hints and Heuristics [38]***

Some Sudoku can be solved by nothing more than repeated application of these two rules—but if all the puzzles were so straightforward, the fad would not have lasted long. Barry Cipra, a mathematician and writer in Northfield, Minnesota, describes a hierarchy of rules of increasing complexity. The rules mentioned above constitute level 1: They restrict a cell to a single value or restrict a value to a single cell. At level 2 are rules that apply to pairs of cells within a row, column or block; when two such cells have only two possible values, those values are excluded elsewhere in the neighborhood. Level-3 rules work with triples of cells and values in the same way. In principle, the tower of rules might rise all the way to level 9.

This sequence of rules suggests a simple scheme for rating the difficulty of puzzles. Unfortunately, however, not all Sudoku can be solved by these rules alone; some of the puzzles seem to demand analytic methods that don't have a clear place in the hierarchy. A few of these tactics have even acquired names, such as "swordfish" and "x-wing." The subtlest of them are no local rules that bring together information from across a wide

### ***6.5.2 Counting Solutions [38]***

In the search for general principles, a first step is to generalize the puzzle itself. The standard 81-cell Sudoku grid is not the only possibility. For any

positive integer  $n$ , we can draw an order- $n$  Sudoku grid with  $n^2$  rows,  $n^2$  columns and  $n^2$  blocks; the grid has a total of  $n^4$  cells, which are to be filled with numbers in the range from 1 to  $n^2$ . The standard grid with 81 cells is of order 3. Some publishers produce puzzles of order 4 (256 cells) and order 5 (625 cells). On the smaller side, there's not much to say about the order-1 puzzle. The order-2 Sudoku (with 4 rows, columns and blocks, and 16 cells in all) is no challenge as a puzzle, but it does serve as a useful test case for studying concepts and algorithms.

How many Sudoku solutions exist for each  $n$ ? To put the question another way: Starting from a blank grid—with no givens at all—how many ways can the pattern be completed while obeying the Sudoku constraints? As a first approximation, we can simplify the problem by ignoring the blocks in the Sudoku grid, allowing any solution in which each column and each row has exactly one instance of each number. A pattern of this kind is known as a Latin square, and it was already familiar to Leonhard Euler more than 200 years ago.

Consider the  $4 \times 4$  Latin square (which corresponds to the order-2 Sudoku). Euler counted them: There are exactly 576 ways of arranging the numbers 1, 2, 3 and 4 in a square array with no duplications in any row or column. It follows that 576 is an upper limit on the number of order-2 Sudoku. (Every Sudoku solution is necessarily a Latin square, but not every Latin square is a valid Sudoku.) In a series of postings on the Sudoku Programmers Forum, Frazer Jarvis of the University of Sheffield showed that exactly half the  $4 \times 4$  Latin squares are Sudoku solutions; that is, there are 288 valid arrangements. (The method of counting is summarized in the illustration on the next page.)

Moving to higher-order Sudoku and larger Latin squares, the counting gets harder in a hurry. Euler got only as far as the  $5 \times 5$  case, and the  $9 \times 9$  Latin squares were not enumerated until 1975; the tally is 5,524,751,496,156,892,842,531,225,600, or about  $6 \times 10^{27}$ . The order-3 Sudoku must be a subset of these squares. Bertram Felgenhauer of the Technical University of Dresden in collaboration with Jarvis counted them in June 2005. The total they computed is 6,670,903,752,021,072,936,960, or  $7 \times 10^{21}$ . Thus,

among all the 9 x 9 Latin squares, a little more than one in a million is also Sudoku grids.

When all these symmetries are taken into account, the number of essentially different Sudoku patterns is reduced substantially. In the case of the order-2 Sudoku, it turns out there are actually only two distinct grids! Applying various symmetry operations can generate all the rest of the 288 patterns. In the order-3 case, the reduction is also dramatic, although it still leaves an impressive number of genuinely different solutions: 3,546,146,300,288, or  $4 \times 10^{12}$ .

Does the large number of order-3 Sudoku grids tell us anything about the difficulty of solving the puzzle? Maybe. If we set out to solve it by some kind of search algorithm, then the number of patterns to be considered is a relevant factor. But any strategy that involves generating all 6,670,903,752,021,072,936,960 grids is probably not the best way to go about solving the puzzle.

### ***6.5.3 NP or Not NP That Is the Question [38]***

Computer science has an elaborate hierarchy for classifying problems according to difficulty, and the question of where Sudoku fits into this scheme has elicited some controversy and confusion. It is widely reported that Sudoku belongs in the class NP, a set of notoriously difficult problems; meanwhile, however, many computer programs effortlessly solve any order-3 Sudoku puzzle. There is actually no contradiction in these facts, but there is also not much help in dispelling the confusion.

Complexity classes such as NP do not measure the difficulty of any specific problem instance but rather describe the rate at which difficulty grows as a function of problem size. If we can solve an order- $n$  Sudoku, how much harder will we have to work to solve a puzzle of order  $n + 1$ ? For problems in NP, the effort needed grows exponentially.

Most discussions of the complexity of Sudoku refer to the work of Takayuki Yato and Takahiro Seta of the University of Tokyo, whose analysis relates the task of



solving Sudoku to the similar problem of completing a partially specified Latin square. The latter problem in turn has been connected with others that are already known to be in NP. This process of "reduction" from one problem to another is the standard way of establishing the complexity classes of computational problems. Yato and Seta employ an unusual form of reduction that addresses the difficulty of finding an additional solution after a first solution is already known. In Sudoku, of course, well-formed puzzles are expected to have only one solution. Yato and Seta say their result applies nonetheless. I don't quite follow their reasoning on this point, but the literature of complexity theory is vast and technical, and the fault is likely my own.

One easily measured factor that might be expected to influence difficulty is the number of givens. In general, having fewer cells specified at the outset ought to make for a harder puzzle. At the extremes of the range, it's clear that having all the cells filled in makes a puzzle very easy indeed, and having none filled in leaves the problem under-specified. What is the minimum number of givens that can ensure a unique solution? For an order- $n$  grid, there is a lower bound of  $n^2 - 1$

Can the  $n^2 - 1$  bound be achieved in practice? For  $n = 1$  the answer is yes. On the order-2 grid there are uniquely solvable puzzles with four givens but not, I think, with three. (Finding the arrangements with just four givens is itself a pleasant puzzle.) For order 3, the minimum number of givens is unknown. Gordon Royle of the University of Western Australia has collected more than 24,000 examples of uniquely solvable grids with 17 givens, and he has found none with fewer than 17, but a proof is lacking.

#### **6.5.4 Logic Rules [38]**

Many puzzle constructors distinguish between puzzles that can be solved "by logic alone" and those that require "trial and error." If you solve by logic, you never write a number into a cell until you can prove that only that number can appear in that position. Trial and error allows for guessing: You fill in a number tentatively, explore the consequences, and if necessary backtrack, removing

your choice and trying another. A logic solver can work with a pen; a backtracker needs a pencil and eraser.

The distinction between logic and backtracking seems like a promising criterion for rating the difficulty of puzzles, but on a closer look, it's not clear the distinction even exists. Is there a subset of Sudoku puzzles that can be solved by backtracking but not by "logic"? Here's another way of asking the question: Are there puzzles that have a unique solution, and yet at some intermediate stage reach an impasse, where no cell has a value that can be deduced unambiguously? Not, I think, unless we impose artificial restrictions on the rules allowed in making logical deductions.

Backtracking itself can be viewed as a logical operation; it supplies a proof by contradiction. If you make a speculative entry in one cell and, as a consequence, eventually find that some other cell has no legal entry, then you have discovered a logical relation between the cells. The chain of implication could be very intricate, but the logical relation is no different in kind from the simple rule that says two cells in the same row can't have the same value. (David Eppstein of the University of California at Irvine has formulated some extremely subtle Sudoku rules, which capture the kind of information gleaned from a backtracking analysis, yet work in a forward-looking, no speculative mode.)

### **6.5.5 A Satisfied Mind [38]**

From a computational point of view, Sudoku is a constraint-satisfaction problem.

Backtracking is also the simpler approach, in the sense that it relies on one big rule rather than many little ones. At each stage you choose a value for some cell and check to see if this new entry is consistent with the rest of the grid. If you detect a conflict, you have to undo the choice and try another. If you have exhausted all the candidates for a given cell, then you must have taken a wrong turn earlier, and you need to backtrack further. This is not a clever algorithm; it amounts to a depth-first search of the tree of all possible solutions—a tree that could have 981 leaves. There is no question that we are deep in the exponential

territory of NP problems here. And yet, in practice, solving Sudoku by backtracking is embarrassingly easy.

Brian Hayes

### **6.6 Proposed algorithm:**

After searching the net, newspapers and magazines we found some of the algorithms has three or four steps to define a solved Sudoku puzzle or to solve a given puzzle.

#### **Here are some steps like that**

- Scanning:
- Cross-hatching
- Marking up
- Analyzing

#### **And some other heavy weight algorithms like**

- Solving Sudoku by backtracking
- Solving Sudoku by a brute-force algorithm
- Solving Sudoku via Stochastic Search / Optimization methods.
- Rapid-style solvers employing backtracking searches
- Rapid-style solvers employing trial-and-error puzzle

Some algorithm uses Elimination Techniques with five/six steps, which include Column, Row, and Mini grid (3X3 box) Elimination. And most of those algorithms need heavy memory for recursion solutions

Here, first a player is given a puzzle where values in some or many cells are given already. One has to fill the empty cells according to the rules, then he/she will submit the solved puzzle and with various algorithms that we already talked about the puzzle will be checked. The output will be generated and shown.

Working with them we found some of them are fast but some takes too much time to even generate a solved puzzle. And the code that solves a puzzle can only solve easy levels of Sudoku and if we are lucky it gives result to a much hard puzzle with in 42 hours.

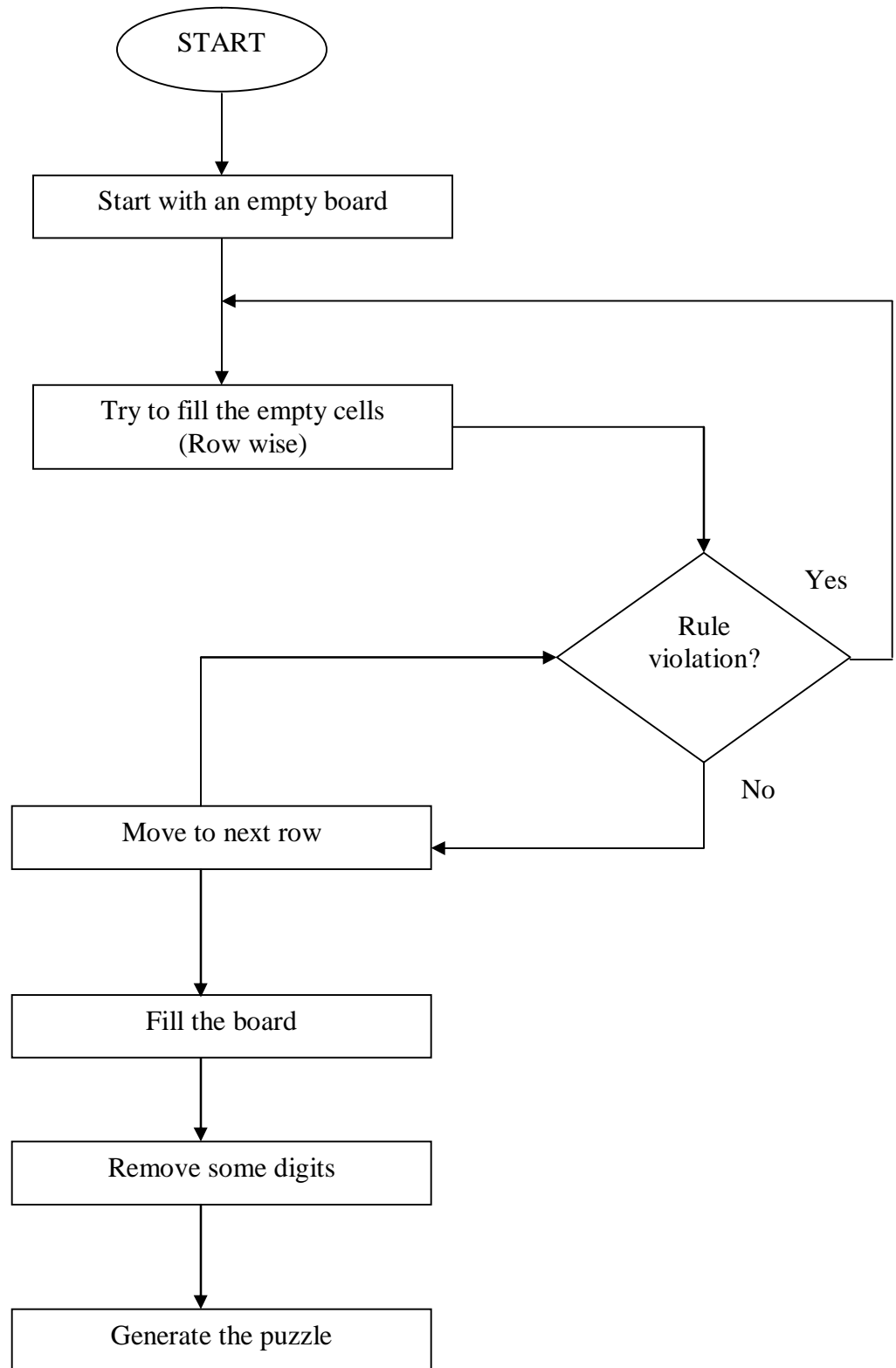
So we developed an idea to make every thing faster. We will go through some

**Simple steps, like**

- Generate unique solved Sudoku puzzle board each time.
- Selectively remove some elements from the board.
- Generate the puzzle
- Remove elements in a way that meets the difficulty level requirements.
- When our system is checking the puzzle we will match every cells with the solved board cells

No extra memory no heavy weight algorithm is there and no backtracking, no recursion is needed.

The flow chart of our new system is given below.

**6.7 Flow chart for our new system****Figure 1-42**

## 6.8 Our first try:

As we tried to implement a new system we had to try with different approaches. 9X9 cells board is a huge board, working with 81 cells is a difficult thing and as we said we will try to implement a simple and lightweight algorithm. According to our new algorithm first we have to generate a solved board with simple technique. So to generate such a solved board we tried Local search

### 6.8.1 what is Local Search? [41]

Local search is a Meta heuristic for solving computationally hard optimization problems. Local search can be used on problems that can be formulated as finding a solution maximizing a criterion among a number of candidate solutions. Local search algorithms move from solution to solution in the space of candidate solutions (the search space) until a solution deemed optimal is found or a time bound is elapsed. Some problems where local search has been applied are:

A local search algorithm starts from a candidate solution and then iteratively moves to a neighbor solution. This is only possible if a neighborhood relation is defined on the search space. The same problem may have multiple different neighborhoods defined on it; local optimization with neighborhoods that involve changing up to  $k$  components of the solution is often referred to as  $k$ -opt.

Typically, every candidate solution has more than one neighbor solution; the choice of which one to move to is taken using only information about the solutions in the neighborhood of the current one, hence the name local search. When taking the one locally maximizing the criterion does the choice of the neighbor solution, the Meta heuristic takes the name hill climbing.

Termination of local search can be based on when the best solution found by the algorithm has not been improved in a given number of steps. Local search algorithms are typically incomplete algorithms, as the search may stop even if the best solution found by the algorithm is not optimal

In our approach we tried to fill the empty cells in row major way. Like we first put a value between 1 and 9 in the first row of the first column, then we take

another random number between 1 and 9 but not used yet, we check all the rules the row rule, the column rule and the sub box rule for the next cell. If we find one then we put that digit. This way we tried to fill the cells.

Here we did not calculate all the candidate solutions for a particular cell. If we could fill the first row we move to the first cell of the next row and start putting unique digits. But filling a 9X9 board is not possible. If we are too much lucky we can fill one or two or may be three rows maximum.

### 6.8.2 Problem faced with our first try:

What we did: -

- Values generate and set.
- No storage of possible values for a cell.
- Placing wrong but feasible data halts the code
- Solutions are always not there (some times not at all).

7	4	6	2	9	1	5	8	3
8	5	2	4	6	7	9	1	

**Figure 1-43**

In our first try we just go on the rows and put sweet able values in the cells. We didn't check the other possible candidates for each cell. So if we put a value and move on, then for that particular cell we may not be able to find the solution. And as we could not move back and replace that particular cell the program halts and there is no solution.

To solve this problem we had to calculate all possible candidate values for each cell. But our goal was to implement a much lightweight algorithm. So we took help of an AI approach named **Hill climbing**.

### 6.8.3 New solution developed using AI approach.

- **Hill climbing**

#### **What is hill climbing?**

Hill climbing is an optimization technique, which belongs to the family of local search. It is a relatively simple technique to implement, making it a popular first choice. Although more advanced algorithms may give better results, there are situations where hill climbing works well.

Hill climbing can be used to solve problems that have many solutions but where some solutions are better than others. The algorithm is started with a (bad) solution to the problem, and sequentially makes small changes to the solution, each time improving it a little bit. At some point the algorithm arrives at a point where it cannot see any improvement anymore, at which point the algorithm terminates. Ideally, at that point a solution is found that is close to optimal, but it is not guaranteed that hill climbing will ever come close to the optimal solution.

An example of a problem that can be solved with hill climbing is the Traveling salesman problem. It is easy to find a solution that will visit all the cities, but this solution will probably be very bad compared to the optimal solution. The algorithm starts with such a solution and makes small improvements to it, such as switching the order in which two cities are visited. Eventually, a much better route is obtained.

Hill climbing is used widely in artificial intelligence fields, for reaching a goal state from a starting node. Choice of next node and starting node can be varied to give a list of related algorithms.



#### 6.8.4 Algorithm used in our system (Hill climbing)

Facing problems with our first approach we tried the hill climbing approach an AI approach. Here, like first try we start putting digits from the first cell then move to next cell of that row and so on. When a row is filled we move to next one. In this new approach we still don't calculate all the possible candidates for every cell like other available algorithms.

Now, if we get stuck or if we cannot put the most appropriate digit or there is a violation of rules we move back to the first cell of that row. As we know in the hill climbing approach if there is no good result then there is no solution. We back track and start with another starting position.

If we stuck again or the violation remains and can't put the desired digit we keep on trying for few times. If no luck then we move back to the first cell of the previous row. And start putting values again.

We got the solution.

This way we generate a solved Sudoku puzzle board, which meets all the requirements. Now to generate an unsolved puzzle we will selectively remove some of the elements from the solved board.

Once a complete grid is generated, it is now time to determine how many cells must be

Taken out (left empty). Based on my experience solving Sudoku puzzles, I have designated

The number of empty cells for each level, as shown in Table

#### **Number of Empty Cells for Each Difficulty Level**

	<b>Level</b>	<b>Empty Cells</b>
1	Easy	40 to 45
2	Medium	46 to 49
3	Difficult	50 to 53
4	Extremely Difficult	54 to 58

**Figure 1-44**

The difference between Easy, and Medium puzzles is that more logical decisions are required and less immediate solvable clues are available as we progress. The Easy puzzles can be solved in less than 5 minutes while the Medium can take about twice the time.

Difficult and Extremely Difficult Plus puzzles require the use of more advanced techniques. Here we need to use pencil marks as well as count rows and columns, checking the numbers to see which ones are missing and where they can be placed. Again, the difference between Medium and Medium Plus is the frequency of logic decisions and the number of bottlenecks encountered during the solution process.

Now the puzzle is ready to be played. Well as we generated this puzzle from a solved board we know there is definitely a solution exists. There may be more solutions as well as, but we wont consider now. Playing the game, when the player submits the puzzle we will match that with the solved board that we already have. So there is no calculation no extra checking no other complex matching.

## **7. Kumon**

### **7.1 To implement Kumon we performed the following Procedures:**

- Gathering comprehensive knowledge about the regular teaching method, which is widely used in our country in primary level of education. We asked students about their present teaching system. We asked whether they all are satisfied with their present teaching method of math and English.
- Collected original Kumon script from abroad.
- Finding the problem of the traditional method of teaching and the solution of the problem.
- Design a process flow of the new web based Kumon teaching method.
- Developed a prototype for the system using PHP and Microsoft SQL server.
- Get parental feedback from survey

. Distance learning and its relationship to emerging computer technologies have together offered many promises to the field of education. In practice however, the combination often falls short of what it attempts to accomplish. Some of the shortcomings are due to problems with the technology; others have more to do with administration, instructional methods, or students. Despite the problems, many users like technologies such as compressed video and see continued growth in the area. This paper will examine some of the current research and thought on the promises, problems, and the future possibilities in modern distance learning, particularly types that are delivered via electronic means.

### **7.2 The Promises of Distance Learning**

Many of the promises of distance learning are financial in nature. Universities hope to save money by delivering education to students that are unable to attend classes because of time or distance. The theory is that class

size increases while the overhead remains the same. The convenience of time and space is a big promise made by distance learning. Students do not have to physically be with the instructor in space and, depending on the method used, they do not have to be together in time as well. This is a great advantage for non-traditional students who cannot attend at regular times.

## **7.3 Concerns**

### **7.3.1 Instructor Concerns**

Instructors have concerns [25] about distance learning, primarily how it will change their role in education. There was a moderately positive attitude about distance learning in general, but moderately negative attitudes about their own use of it. Instructors worry about putting their course materials online because once there, the knowledge and course design skill in that material is out of their possession. This puts the administration in a position to hire less skilled, and cheaper, workers to deliver the technologically prepackaged course. Instructors are not always convinced that administration is behind distance learning. The rewards are not always there for the good distance-learning instructor. The increased amount of time necessary to adequately prepare for distance learning takes away from the activities they will be evaluated on, such as grant writing and publishing. Many of the instructors concerns are valid and should be addressed by administration as distance learning becomes more common, as is predicted to happen.

### **7.3.2 Student Concerns**

There are the students and their concerns with distance learning classes. Not all students are suited to this type of learning and not all subjects are best taught via this medium. More mature students are the most likely to find success with distance learning. The successful student needs to have a number of characteristics such as tolerance for ambiguity, a need for autonomy, and an

ability to be flexible. Students need the attention of the instructors [25]. This may be truer in a distance situation than in a traditional classroom.

## **7.4 Design considerations**

### **7.4.1 Systematic design and development [26]**

The instructional development process for distance education, consisting of the customary stages of design, development, evaluation, and revision. In designing effective distance instruction, one must consider the goals, needs, and characteristics of teachers and students, but also content requirements and technical constraints. If unusual delivery systems are required, they must be made accessible to all participants. Revision based on feedback from instructors, content specialists, and learners is an ongoing process. Provisions must be made for continually updating courses, which depend on volatile information, to keep the subject matter current and relevant.

### **7.4.2 Interactivity**

Distance education systems involve interactivity [27] between teacher and students, between students and the learning environment, and among students themselves. Though students felt that the accessibility of distance learning courses far outweighs the lack of dialogue.

### **7.4.3 Active learning**

As active participants in the learning process, students affect the manner in which they deal with the material to be learned. Learners must have a sense of ownership of the learning goals. They must be both willing and able to receive instructional messages.

## 7.5 Internet conducted Web-based Kumon method

### **Kumon: Definition [28]**

We have chosen Internet conducted distance education .A new method of teaching mathematics is becoming increasingly popular in the UK — the Kumon Method. This Japanese system of learning is based on repetition and practice, with students spending a little time every day working on sums. *Practice makes perfect. Learn by your mistakes.*

### **7.6 How do we relate Kumon [28] with distance education?**

We were thinking of something new could be added in distance education we searched for some successful traditional method of teaching and we find a teaching method called “Kumon” which is widely being used in some developed countries. We found that Kumon is very much expensive there, so we choose this traditional teaching method and tried to add the method to distance education to minimize the cost. On most of the country those who are using Kumon, is not computerized.

Children are getting more interest on computer for gaming, drawing, But those are not used to develop their brain. We tried to attach an education system to computer so children could find more interest on that education too. We have chosen the method of teaching and made it web-based. In Kumon what we think of as a very Japanese emphasis on the physical process of drawing the numbers and on physically handling the world generally. (Think of the Japanese fascination with hand-done graphics.) One of the ancillary games for the children to play is simply placing numbers on a number board in the web page. This doesn't just help them to understand numbers. It also helps them to get better at simply handling things, while thinking at the same time. As with so much of Kumon, doing the number board so that every number is where it should be is in principle very easy, so no child is humiliated by not being able to do it. But doing it fast isn't so easy, so the cleverer ones are kept interested. .

In our website each child doing an individually selected clutch of repetitive problems. Even more tedious analysis to tell you what each child should be doing

next. This is surely the sort of stuff that computers and their recent combined offspring, the Internet — were invented to supervise. But I sense that the Kumon people resist such notions. *There's so far been no mention of computers in any of the Kumon back-up or sales literature that I've seen.*

Kumon would still be a incurably obscure cottage industry in Japan run by Toru Kumon's friends and relatives. So no "computers for kids". In this respect there is something very traditional about Kumon. We choose Kumon to implement on website so this could appeal to any parents who have been proudly shown the new "computer centre" in their child's regular school, bursting with under-used, over-priced and easy-to-steal jumped-up pocket calculators and/or dictionaries, but who can't understand why that same child can't add up properly the way most children could in the past.

For good reasons, The Kumon people have to maintain the chain of command, to keep the parents involved, to keep you coming to the Kumon website and to keep track of how your child is doing. Twenty minutes a day is okay for committed parents who see the point of it all, we only get seriously stuck into something if we have entirely re-arranged it, into a *game*. So computers *will* eventually address what seems to be the long-term limitation of Kumon, which is that children can now do Kumon mathematics entirely on sitting on his/her own computer, that is, with just his own computer, his own Internet connection, and a bill that the average kid could pay out of pocket money.

Web based Kumon is fine as a helpful mechanism for all those *families* who want their children's to come out from "traditional" education, so to speak, for the ills of the relentlessly deteriorating regular system. The massive *improvement* in the educational level of the entire population that the modern economy and modern society now seem to be demanding won't happen until computers are fed into the system, and prices slashed from £38 per month to nearer £0 per month.

## **7.7 Starting time [29]**

### **What age do children start with Kumon?**

While parents are enrolling their children to school, the most common time is now during the early primary years. Parents are typically motivated to see their children develop into confident learners through strong study habits so that school is not a struggle and that the benefits of advanced reading and calculation abilities flow through to all their other subjects/ activities.

### **Is Kumon a "remedial" or" enrichment" programmed?**

With materials and instruction tailored to the individual child, all students make successful learning and progress. It is common knowledge that starting as young as possible and developing skills early will lead to the greatest cumulative benefits. Kumon recommends preschool enrolment or during the early primary years for enrichment purposes.

### **Is it one-on-one tuition?**

Kumon study offers more than one-on-one tuition. The Kumon Method facilitates[29] self-acquisition of the skills and study habits needed to improve academic performance. This empowers children to succeed on their own, giving them a sense of accomplishment that fosters further achievement. A child who develops learning tools in this way may never have to rely on costly tutoring.

## **7.8 Step of success with Kumon [26]:**

- **Success from the first day**

Each Kumon student begins at a comfortable starting point - determined by our initial 'Diagnostic Test' [30] - with work that he or she can complete easily. Confidence is vital in all learning so this immediate experience ensures that he/she can focus easily, quickly develop a daily routine and master the basics all at the same time. In this way, stronger concentration builds gradually and learning becomes more enjoyable and less pressured.



- **Advanced step by step**

Instant feedback [31]/result maintains motivation because each child knows how each piece relates to the last. New concepts are illustrated with clear examples and students solve exercises based on these examples to gain a thorough understanding. Progression is so gradual it enables students to acquire the skills to advance immediately.

- **Speed & Accuracy [32]**

In order to demonstrate appropriate command of the material, students are guided to complete materials with perfect scores and within set time periods. Speed and accuracy are measurable indicators of mastery and show a readiness to move on to the next skill level. Strengthening speed and accuracy builds a child's capacity to tackle more difficult learning later.

- **Practice until perfect [33]**

The Kumon Method allows child to advance steadily at a comfortable pace equal to his or her ability and initiative. Your child's individualized programmed is never compromised by the needs of a group or a prescribed teaching agenda. The first priority of the Kumon Supervisor is to enable each and every child to perform and progress to his or her full potential - including advanced study whenever possible.

- **Realizing each child's potential**

The Kumon Method allows child to advance steadily at a comfortable pace equal to his or her ability and initiative. Your child's individualized programmed is never compromised by the needs of a group or a prescribed teaching agenda. The first priority of the Kumon Supervisor is to enable each and every child to perform and progress to his or her full potential - including advanced study whenever possible.

## **7.9 Parents question and our feedback**

We proposed the method to some parents and faced some common question from them.

- **How much time commitment does Kumon require?**

Most students attend Kumon twice a week year-round and do brief daily assignments the other five days. Kumon sessions and assignments can usually be completed in less than twenty minutes, for mathematics or English. Parental involvement is strongly encouraged and recommended for optimal results.

- **How long do students stay enrolled?**

Our goal is to develop children's potential to comfortably learn materials and through experience we now aim for all students to move well ahead of their school curriculum. This takes some students longer than others but by this stage, most students have become self-motivated and show a new curiosity and enthusiasm to challenge them to tackle new materials.

- **How long will it take for my child's results to improve at school?**

Confidence will lift almost immediately. As you can imagine, school classes are required to progress through the curriculum as a group whereas Kumon allows the child's ability and initiative to govern how far they go. The moment students feel more confident and personally involved in their learning is the starting point for better school grades.

- **Why haven't I heard more about Kumon?**

Kumon has chosen to devote most of its resources to developing rather than promoting its programs. While Kumon has grown steadily in Australia and New Zealand -largely by word-of-mouth-it has been a household name primarily in the Asian-Australian community. In Bangladesh it is almost unknown to people because till now no one introduced this technique yet.

- **How much does it cost?**

In Australia and New Zealand, the monthly fee is \$100 per month per subject. There is an initial enrolment fee of \$60 in Australia and New Zealand. Correspondence study is also available under the same enrolment fees but with monthly fees of \$107 in Australia and New Zealand. Kumon will conduct your child's Diagnostic Test free of charge. But in our web-based system we minimized the cost almost \$10.

### **7.10 Parental involvement**

Parents today are becoming more involved in their children's education and are looking beyond school for supplementary assistance. As this is a web-based educational system, a child sometime could not catch the system. Initially, instructor or parent should guide the child. Our system will give parents the additional support they require to help provide their children with the best possible opportunity of achieving in life.

### **7.11 Advantages [29]**

Kumon study can give a child

- A solid foundation for academic success
- Confidence to take on new challenges
- A positive attitude to study
- Strength of character to overcome obstacles
- Options in school and life that they would not otherwise have had
- The opportunity to realize their full potential

In short, together Kumon can give your child the best chance of leading a happy and successful life.

### **7.12 Evaluation**

At first we provided tradition exam script (paper based) to some of the students and then we asked same group of students to participate on the web-

based exam. They felt lot more interest on the web based Kumon script. But initially we found that they could not adopt with the web-based script.

From evaluation we found that without any parental or instructor involvement it is impossible for a student to take part. So in our site further we will provide instruction, which will guide student thoroughly how to take part.

### 7.13 Traditional Kumon Script

The image shows two pages of a traditional Kumon exam script. The left page is labeled C46a and the right page is labeled C50a. Both pages are for the topic 'Multiplication up to 9'.

**Page C46a:** Student Name: Sahan, Date: 16/7/11, Time: 1:32 to 1:43. The page includes a 'Multiplication Table' with instructions to read it aloud and tick the box. The table lists multiplication facts from  $1 \times 9 = 9$  to  $9 \times 9 = 81$  with their corresponding words. A 'Multiply' section contains 9 problems, with the first five solved: (1)  $1 \times 9 = 9$ , (2)  $2 \times 9 = 18$ , (3)  $3 \times 9 = 27$ , (4)  $4 \times 9 = 36$ , (5)  $5 \times 9 = 45$ .

**Page C50a:** Student Name: Sahan, Date: 16/7/11, Time: 1:32 to 1:43. The page includes instructions to fill in blanks with appropriate numbers. Three rows of boxes contain numbers, with some handwritten numbers in the boxes. A 'Multiply' section contains 16 problems, with the first eight solved: (1)  $6 \times 9 = 54$ , (2)  $9 \times 9 = 81$ , (3)  $5 \times 9 = 45$ , (4)  $3 \times 9 = 27$ , (5)  $4 \times 9 = 36$ , (6)  $8 \times 9 = 72$ , (7)  $7 \times 9 = 63$ , (8)  $2 \times 9 = 18$ .

**Figure 1-45 Traditional exam script**

The traditional Kumon script has no relation with computer. Student get question and solve the question repeatedly on paper. From practice they become ready for the tougher question.

#### **7.14 Problem with traditional Kumon script**

In traditional Kumon method a student need to take admission in Kumon center.

He has to attend on exam on the Kumon center 2 times a week. It is bit difficult for children to present in Kumon center beside their school education. The whole teaching method is very costly. Our goal is to minimize the cost. As in Bangladesh there were no Kumon center yet it will be difficult to set up Kumon center quickly in this country. So we make it web based to send Kumon center at every home.

## 7.15 Our proposed web based Kumon exam Script

# Kumon, Bangladesh

Time left (minutes): **20**

**login**

>instructor  
>student

**Sign up**

>instructor  
>student

Exam Script-1	Day-1	January	Week-1
---------------	-------	---------	--------

01. There are 2 boxes containing 43 pencils each. How many pencils are there altogether?

<p>(Solve by Adding)</p> <table style="margin-left: 20px;"> <tr><td>43</td></tr> <tr><td>43</td></tr> <tr><td><input style="width: 40px; height: 20px;" type="text"/></td></tr> </table>	43	43	<input style="width: 40px; height: 20px;" type="text"/>	<p>(Solve by Multiplying)</p> <table style="margin-left: 20px;"> <tr><td>43</td></tr> <tr><td>2</td></tr> <tr><td><input style="width: 40px; height: 20px;" type="text"/></td></tr> </table>	43	2	<input style="width: 40px; height: 20px;" type="text"/>
43							
43							
<input style="width: 40px; height: 20px;" type="text"/>							
43							
2							
<input style="width: 40px; height: 20px;" type="text"/>							

---

02. There are 2 boxes containing 24 erasers each. How many eraser are there altogether?

<p>(Solve by Adding)</p> <table style="margin-left: 20px;"> <tr><td>24</td></tr> <tr><td>24</td></tr> <tr><td><input style="width: 40px; height: 20px;" type="text"/></td></tr> </table>	24	24	<input style="width: 40px; height: 20px;" type="text"/>	<p>(Solve by Multiplying)</p> <table style="margin-left: 20px;"> <tr><td>24</td></tr> <tr><td>2</td></tr> <tr><td><input style="width: 40px; height: 20px;" type="text"/></td></tr> </table>	24	2	<input style="width: 40px; height: 20px;" type="text"/>
24							
24							
<input style="width: 40px; height: 20px;" type="text"/>							
24							
2							
<input style="width: 40px; height: 20px;" type="text"/>							

---

03. There are 2 boxes containing 52 nootbooks each. How many nootbooks are there altogether?

<p>(Solve by Adding)</p> <table style="margin-left: 20px;"> <tr><td>52</td></tr> <tr><td>52</td></tr> <tr><td><input style="width: 40px; height: 20px;" type="text"/></td></tr> </table>	52	52	<input style="width: 40px; height: 20px;" type="text"/>	<p>(Solve by Multiplying)</p> <table style="margin-left: 20px;"> <tr><td>52</td></tr> <tr><td>2</td></tr> <tr><td><input style="width: 40px; height: 20px;" type="text"/></td></tr> </table>	52	2	<input style="width: 40px; height: 20px;" type="text"/>
52							
52							
<input style="width: 40px; height: 20px;" type="text"/>							
52							
2							
<input style="width: 40px; height: 20px;" type="text"/>							

**Figure 1-46 Sample exam script.**

## 7.16 Database Design

Table 2

Student

Student:

<u>Student-ID</u>	NAME	password

Table 3

Question

Question:

<u>Question-Id</u>	Question-Level	Question-Set

Table 4

Marks

Marks

<u>Student-Id</u>	Question-Id	Date	Marks

Table 5

## Cumulative Marks

Cumulative marks

<u>Student-Id</u>	Total marks

## 7.17 ER-diagram:

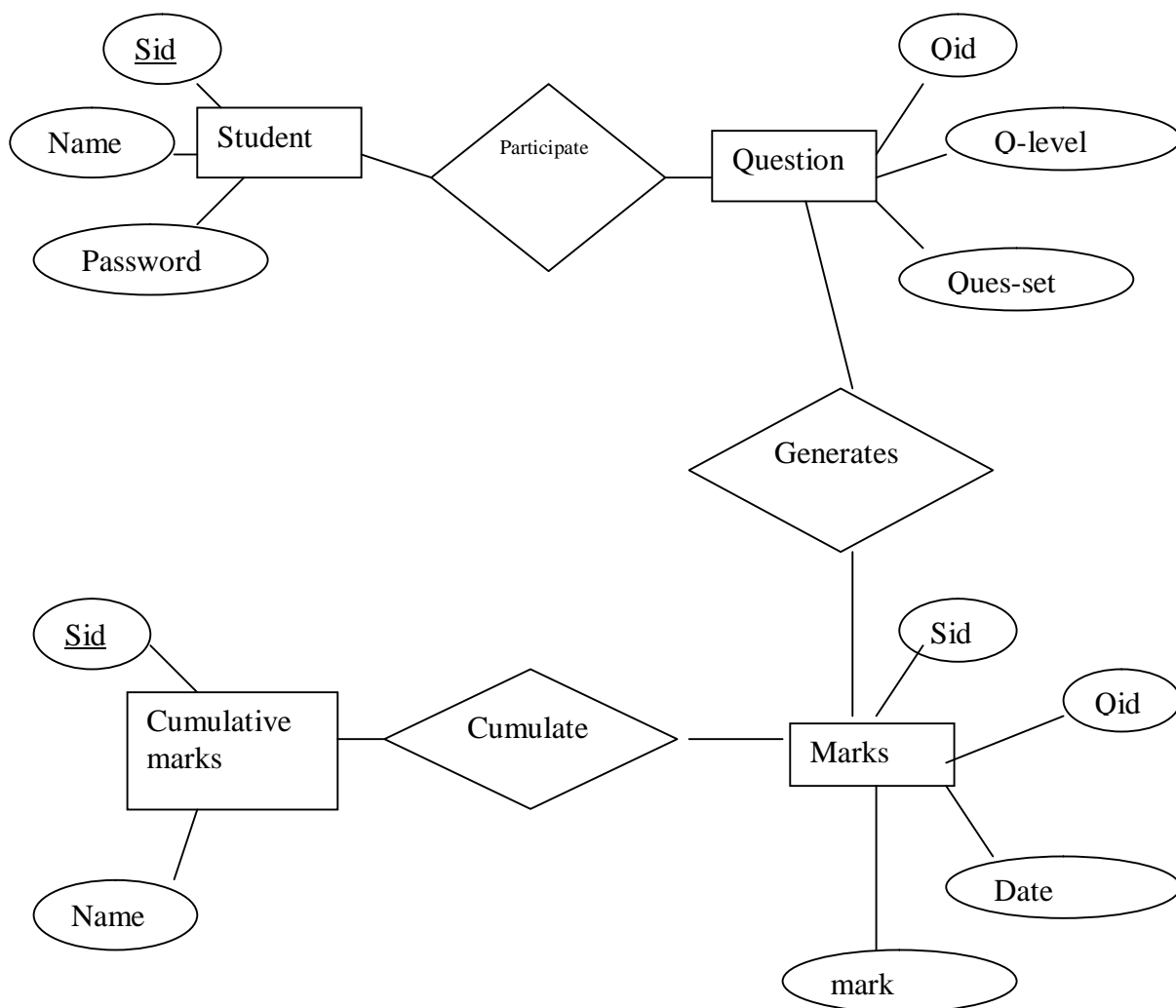
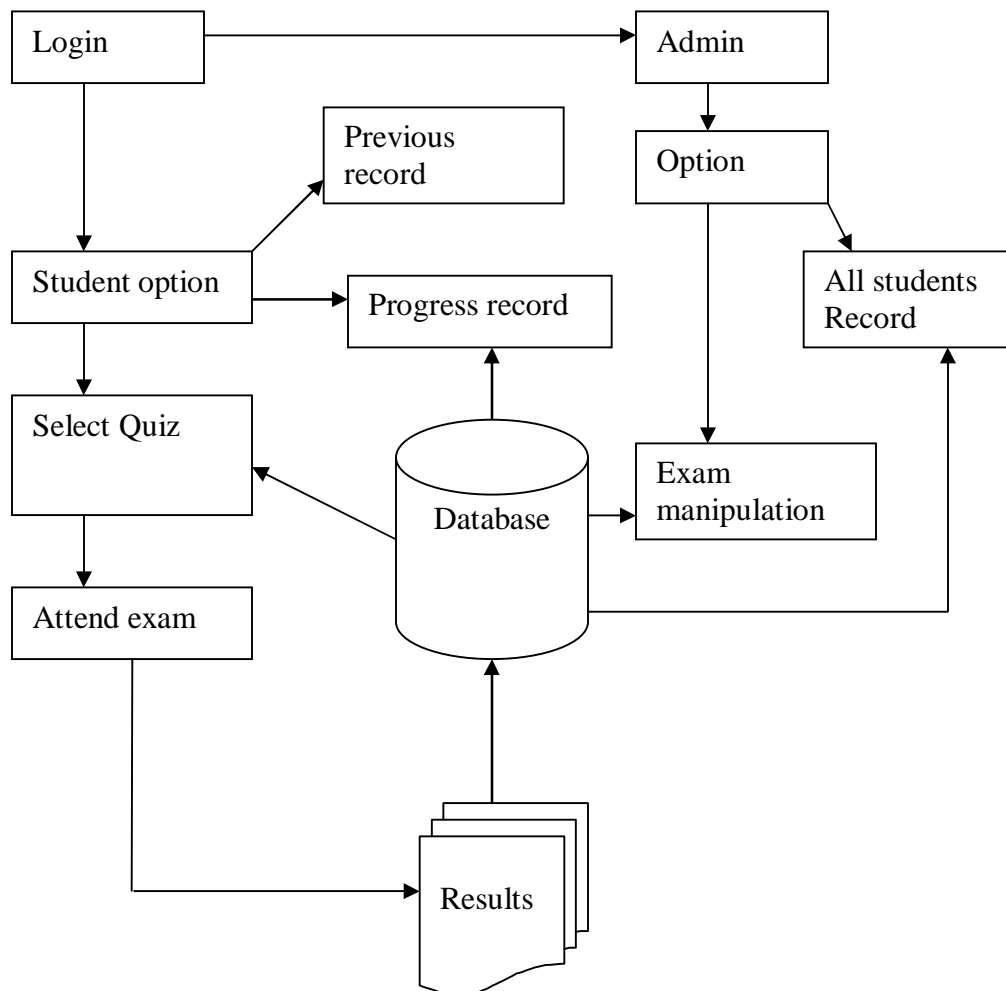


Figure 1-47 ER-diagram of the system



## 7.18 System Architecture

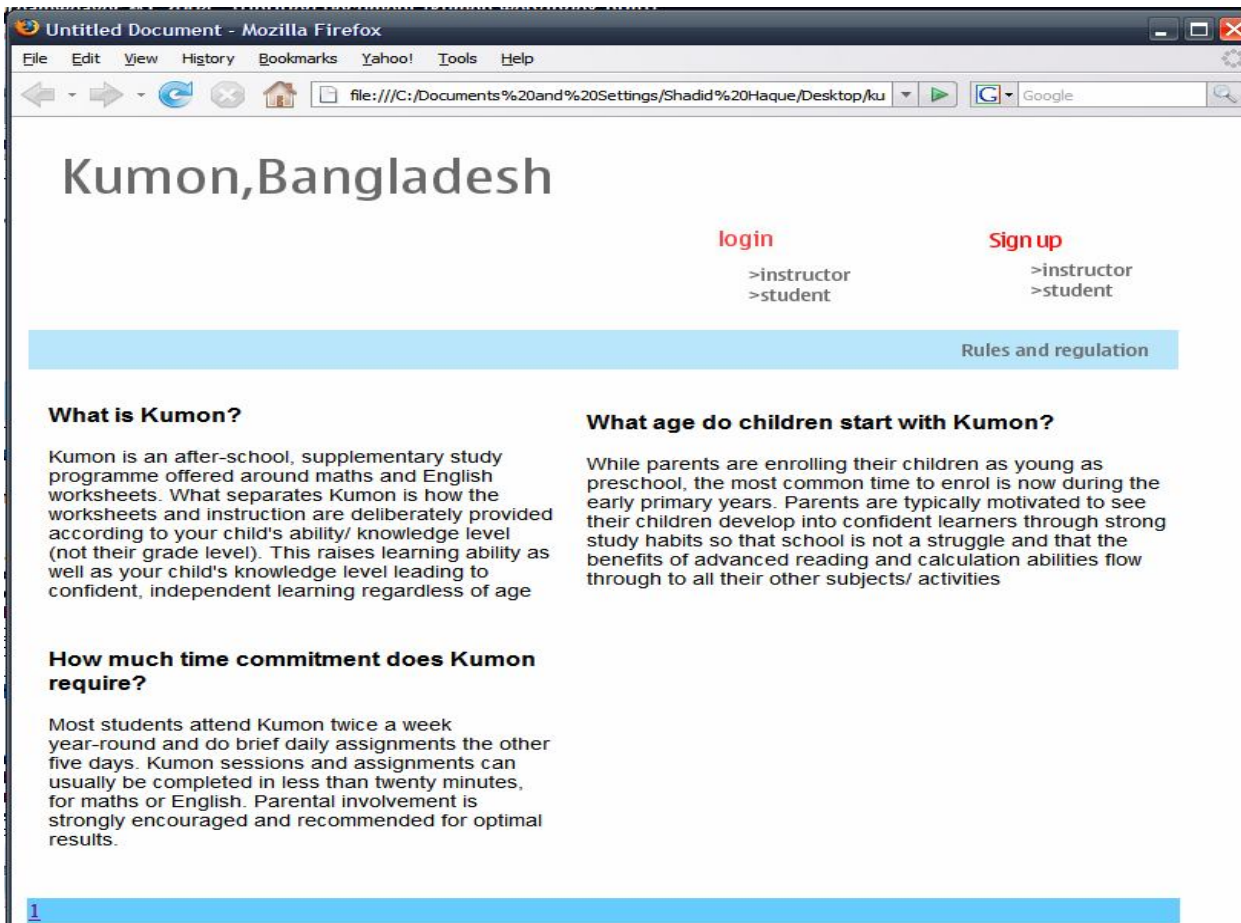


**Figure 1-48 System architecture**

A registered student will login to the site there he will get an exam schedule. After clicking on the day of the exam schedule he will enter into the quiz system. The quiz question will be called from the database. After giving exam the result will be added to the database. Student could easily see their record anytime. After login the admin will easily go to exam manipulation system,

where he could easily reorganize exam scripts. An administrator also can see students record from the database.

## 7.19 Implementation: Web site



**Figure 1-49 Home page**

## Student registration form:

The screenshot shows a web browser window titled "Untitled Document - Mozilla Firefox". The address bar contains a file path: "file:///C:/Documents%20and%20Settings/Shadid%20Haque/Deskto". The page content includes the following elements:

- Header: "Kumon, Bangladesh" with a "Sign out" link in the top right corner.
- Navigation: "login" and "Sign up" links in red text. Below "login" are links ">instructor" and ">student". Below "Sign up" are links ">instructor" and ">student".
- Section Header: "Student Registration form" in a teal bar.
- Text: "Most students attend Kumon twice a week year-round and do brief daily assignments the other five days. Kumon sessions and assignments can usually be completed in less than twenty minutes, for maths or English. Parental involvement is strongly encouraged and recommended for optimal results."
- Form Fields:
  - Student name
  - School
  - Level of education
  - Parents name
  - phone no
  - email address
- Submit Button: A blue "Submit" button.
- Message: "A mail with login information will be sent to your mail"
- Status Bar: "Done"

**Figure 1-50 student registration form**

- The registration form will fill up by the parents initially. The user login and the password will be sent to the students email address.

## Student Login

# Kumon, Bangladesh

[login](#)  
>instructor  
>student

[Sign up](#)  
>instructor  
>student

Student login

Login name

Password

**Figure 1-51 Student login page**

- Only valid users could attend to the exam. They could login to the system with the login name and password

## Exam Schedule

File Edit View History Bookmarks Yahoo! Tools Help

file:///C:/Documents%20and%20Settings/Shadid%20Haque/Desktop/kumon%20web/ Google

Untitled Document Untitled Document Untitled Document Untitled Document Untitled Document

Sign out

# Kumon, Bangladesh

**login** **Sign up**

>instructor >student >instructor >student

**Exam schedule**

Most students attend Kumon twice a week year-round and do brief daily assignments the other five days. Kumon sessions and assignments can usually be completed in less than twenty minutes, for maths or English. Parental involvement is strongly encouraged and recommended for optimal results.

Month	Week-1		Week-2		Week-3		Week-4	
January	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
february	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
March	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
April	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
May	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
June	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
july	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
August	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
September	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
October	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
november	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2
December	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2	Day-1	Day-2

**Figure 1-52 Exam schedule created for the student.**

## Auto checked answer script

## Kumon, Bangladesh

[login](#)      [Sign up](#)  
 >instructor      >instructor  
 >student          >student

Online Quiz For Child			
01. There are 2 boxes containing 43 pencils each. How many pencils are there altogether?			
(Solve by Adding)	43 43 <b>86</b> ✓	(Solve by Multiplying)	43 43 <b>45</b> ✗
02. There are 2 boxes containing 24 erasers each. How many eraser are there altogether?			
(Solve by Adding)	24 24 <b>48</b> ✓	(Solve by Multiplying)	24 2 <b>23</b> ✗
03. There are 2 boxes containing 52 nootbooks each. How many nootbooks are there altogether?			
(Solve by Adding)	52 52 <b>104</b> ✓	(Solve by Multiplying)	52 2 <b>54</b> ✗
You given <b>3</b> correct answer out of <b>6</b> Your Accuracy is <b>50%</b> <a href="#">Go Back To Quiz</a>			

**Figure 1-53 Auto checked answer script**

- All the results will be saved into database.

## Result script viewed from Student login

The screenshot shows a web browser window with the title 'Untitled Document - Mozilla Firefox'. The address bar shows a local file path. The page content includes the text 'Kumon, Bangladesh' and a 'Sign out' link. There are two columns of buttons: 'login' and 'Sign up', each with '>instructor' and '>student' options. Below this, it says 'USER: RAHAT'. A table titled 'List of Marks' displays the following data:

Transection ID	User	1	2	3	Total
1188727407	rahat	1	0	0	1
1188783093	rahat	2	4	3	9

Below the table, there are navigation links: 'First Prev 1 of 1 Next Last'.

**Figure 1-54 Result record from the database**

- The table shows day-1 and day-2 result script of one student in each row. 3rd, 4th, 5th column shows 3 quiz marks attended by the student. The 6<sup>th</sup> column shows the total marks of the student. From this table the student could aware about the improvement of his/her skill.

## Result viewed by the instructor

The screenshot shows a Mozilla Firefox browser window with the address bar displaying a local file path. The website content includes the title 'Kumon, Bangladesh', a 'Sign out' link, and two columns of links for 'login' and 'Sign up', each with sub-links for '>instructor' and '>student'. A 'Search Marks' section contains a 'User' input field and a 'Search' button. Below this is a 'List of Marks' table with the following data:

Transection ID	User	1	2	3	Total
1188727327	mishu	1	2	1	4
1188727407	rahat	1	0	0	1
1188728504	mishu	1	0	0	1
1188783093	rahat	2	4	3	9
1188784261	rahat	6	6	4	16

At the bottom of the table, there are navigation links: 'First Prev 1 of 1 Next Last'.

**Figure 1-55 All students' records**

- From instructor Login a teacher could easily see the record of all the students
- One student could not see others record



Search Marks	
User	<input type="text" value="mishu"/>
<input type="button" value="Search"/>	

List of Marks					
Transection ID	User	<u>1</u>	<u>2</u>	<u>3</u>	Total ▲
1188728504	mishu	1	0	0	1
1188727327	mishu	1	2	1	4
First Prev 1 of 1 Next Last					

***Figure 1-56 Searching record with name***

- A Teacher can search specific students result by name to see the progress.

## 8. Conclusion

Before our thesis starts we wanted to develop a web portal regarding e learning. An e learning web portal where some new learning system will be introduce for Bangladeshi students with some interesting puzzles which will help them to think differently. By solving various puzzles they can develop their logical and thinking skills.

As we mentioned earlier in this short period of time we could not include all the parts that we thought about.

While working with Sudoku puzzle we studied a lot on its generation algorithm and found that we may modify it, we can even make it simple and faster so we start working with new algorithm and coding for Sudoku puzzle. We developed new system and implement that. So in our paper Sudoku took the most of the weight.

We tried to introduce a new mathematics learning method named Kumon. Kumon is a very popular method used in outer world which is very expensive but effective. For our country this is new and it will be interesting for the younger students.

In our system we may need some modification and we may have to include some other learning parts. The site will be more interesting as we include more elements and it will grow bigger.

## 9. Bibliography

1. Bammel, Stanley E., and Jerome Rothstein. 1975. The number of 9 x 9 Latin squares. *Discrete Mathematics* 11:93-95.
2. Eppstein, David. Preprint. No repetitive paths and cycles in graphs with application to Sudoku. <http://arxiv.org/abs/cs.DS/0507053>
3. Felgenhauer, Bertram, and Frazer Jarvis. Preprint. Enumerating possible Sudoku grids. <http://www.shef.ac.uk/~pm1afj/sudoku/sudoku.pdf>
4. Pegg, Ed. 2005. Math Games: Sudoku variations. MAA Online. [http://www.maa.org/editorial/mathgames/mathgames\\_09\\_05\\_05.html](http://www.maa.org/editorial/mathgames/mathgames_09_05_05.html)
5. <http://www.sudokudaily.net/puzzle/>
6. <http://www.sudokucentral.com/how-to-solve-sudoku>
7. <http://www.sudokucentral.com/what-is-sudoku>
8. <http://www.sudoku.name/rules/en>
9. [http://www.sudokuessentials.com/sudoku\\_tips.html](http://www.sudokuessentials.com/sudoku_tips.html)
10. [http://www.sudokuessentials.com/how\\_to\\_play\\_sudoku.html](http://www.sudokuessentials.com/how_to_play_sudoku.html)
11. <http://www.csse.uwa.edu.au/~gordon/sudokumin.php>
12. Simonis, Helmut. 2005. Sudoku as a constraint problem. In *Proceedings of the Fourth International Workshop on Modeling and Reformulating Constraint Satisfaction Problems*, pp. 13-27.
13. Sudoku Programmers Forum. 2005. Discussion thread, May 5, 2005, through
14. <http://www.setbb.com/phpbb/viewtopic.php?t=27&mforum=sudoku>
15. <http://en.wikipedia.org/wiki/Sudoku>
16. <http://www-imai.is.s.u-tokyo.ac.jp/~yato/data2/SIGAL87-2.pdf>
17. [http://en.wikipedia.org/wiki/Mathematics\\_of\\_Sudoku](http://en.wikipedia.org/wiki/Mathematics_of_Sudoku)
18. [http://en.wikipedia.org/wiki/Algorithmics\\_of\\_sudoku](http://en.wikipedia.org/wiki/Algorithmics_of_sudoku)
19. <http://www.sudoku.com/>
20. <http://sudoku.com.au/>

21. [Faculty and Distance Education Services \(FDES\)](#). School of Undergraduate Studies, [University of Maryland University College](#). Retrieved December 14, 2006.
22. <http://ocw.mit.edu/OcwWeb/Global/all-courses.htm>
23. Turkish Online Journal of Distance Education-TOJDE [January 2006](#) ISSN 1302-6488 [Volume: 7 Number: 1 Article: 10](#)
24. <http://www.apress.com>
25. <http://www.westga.edu/~distance/ojdla/fall53/valentine53.html>
26. <http://www.springerlink.com/content/g6822t7p804l05gl/>
27. <http://www.educause.edu/ir/library/pdf/PUB3017.pdf>
28. [www.kumon.com/](http://www.kumon.com/)
29. <http://www.kumon.com/about/default.asp?language=USA>
30. <http://www.querycat.com/site/www.kumon.com.sg>
31. <http://www.kumon.com.ph/feedback.htm>
32. [www.kumonsantaana.com](http://www.kumonsantaana.com)
33. [http://www.kumon-olathe.com/html/why\\_kumon.html](http://www.kumon-olathe.com/html/why_kumon.html)
34. <http://en.wikipedia.org/wiki/Sudoku#Strategies>
35. <http://www.answers.com/topic/sudoku?cat=entertainment>
36. [http://davidbau.com/archives/2006/09/04/sudoku\\_generator.html](http://davidbau.com/archives/2006/09/04/sudoku_generator.html)
37. [http://en.wikipedia.org/wiki/Algorithmics\\_of\\_sudoku](http://en.wikipedia.org/wiki/Algorithmics_of_sudoku)
38. <http://www.americanscientist.org/template/AssetDetail/assetid/48550?&print=yes>
39. [http://en.wikipedia.org/wiki/Mathematics\\_of\\_Sudoku](http://en.wikipedia.org/wiki/Mathematics_of_Sudoku)
40. <http://www.di-mgt.com.au/sudoku.html>
41. [http://en.wikipedia/wiki/local search](http://en.wikipedia/wiki/local_search)