An Implementation of Backtracking Algorithm for Solving A Sudoku-Puzzle Based on Android

To cite this article: Herimanto *et al* 2020 *J. Phys.: Conf. Ser.* **1566** 012038

An Implementation of Backtracking Algorithm for Solving A Sudoku-Puzzle Based on Android

Herimanto[1], P Sitorus[1], and E M Zamzami[2]

[1]Master Programme in Informatics, Universitas Sumatera Utara.
[2]Faculty of Computer Science and Information Technology, Universitas Sumatera Utara

*E-mail: pardzheri@gmail.com

Abstract. Sudoku is a pretty popular number game. The goal of this game is to fill a 9x9 matrix with unique numbers, and there should not be repeated numbers in each row, column, or block. This paper proposed a solution to solve Sudoku using the Backtracking Algorithm. This algorithm is quite efficient because it does not have to check all the possibilities that exist, but which leads to only the solution that will be processed, namely by pruning every element that does not lead to the solution. Thus the time required is quite efficient and suitable for use in reasonably complex numbers games like sudoku. By implementing the backtracking algorithm in the sudoku game, the complexity of the algorithm can be as large $\boldsymbol{\theta}$ (n^3).

1. Introduction

Computers are one of the essential parts of the improvement of information technology. The computer that was initially only used by academics and the military is now widely used in various fields, such as business, health, education, psychology, games, and so on. The development of Android-based adolescent reproductive health promotion is one of the efforts to increase adolescent knowledge about adolescent reproductive health [8].

The development of computers began to spread to various fields. Various branches of science developed and were born from it, and one of them is artificial intelligence. Artificial intelligence is a field of computer science that discusses how a computer device can be programmed with a machine to be able to do what the programmer wants through specific instructions. The interesting one of this branch of artificial intelligence is where computers can be made as if they have intelligence. Contrast-enhanced is the digital manipulating carried out to increase the contrast and eliminate the noise in digital imaging [9].

One representation of artificial intelligence science is a Game. With artificial intelligence, certain games can be solved through computers. The games that can be classified into artificial intelligence games are Sudoku. Sudoku is a game that has been popular since the 1970s. This game is like a number game where the player must fill in the available empty boxes with random numbers 1-9 (standard Sudoku). The rules of the Sudoku game are straightforward. It is namely filling in a puzzle with random numbers. But it seems like not a few who fail and cannot finish this game. That's because there are more than thousands of solutions offered to solve Sudoku puzzles.

Thus came the idea to create an Android application that can generate puzzles and solve them. From the exposure, it can be determining the limitations of the problem are as follows:

- Making this Sudoku game implements the backtracking algorithm on mobile / Android applications.
- The programming language used is java. Supporting software used to make Sudoku games is Android studio.
- Sudoku puzzles used are a standard size 9 x 9 boxes.
- The initial element generated consists of a maximum of 36 numbers and a minimum of 27 puzzle numbers that have been filled in with numbers, and each row, column, and block (3x3) has been filled in at least one box and may not be empty on each of its genera.

2. Literature Review

2.1. Backtracking Algorithms

D.H. Lehmer was the first to introduce the backtracking algorithm in 1950. The backtracking algorithm is one of the problem-solving methods included in a strategy based on searching the solution space, but it does not have to examine all possibilities, only those that lead to only solutions will be processed. Algorithms backtracking is also an algorithm that works recursively, where the search process is based on the Depth-First Search (DFS) algorithm, which is to search for systematic solutions to all possible solutions and search for answers is done by tracing a tree-shaped structure rooted [2]. Therefore this algorithm is quite powerful and very good to be applied in problem-solving and to provide artificial intelligence in the game. Several types of digital games that are generally commonly known by the public, such as Chess, Math Maze, Tic Tac Toe, to Sudoku can be found a solution by implementing the backtracking algorithm.

The backtracking algorithm is an improvement of the brute force algorithm, which is to find solutions to problems among all possible solutions systematically. Backtracking is a typical form of recursive algorithm and is based on DFS (Depth-First Search) in finding the right answer. In another sense, the backtracking algorithm works like experimenting with several possibilities that lead to the solution until it finds the most appropriate one. So there is no need to check all possible solutions, but it is enough that only leads to the solution, namely by sorting pruning the nodes that do not lead to the solution. Thus the search time can be saved. The difference with the brute force algorithm is the basic concept, namely, in backtracking, all solutions are made in the form of a solution tree (tree), and then the tree will be traced in DFS (Depth-First Search) to find the best-desired solution.[1]

2.2. Related Works

The previous studies related to research conducted by the author include:

- Llyod (2019) Conduct research published by IEEE in international journals entitled “Solving Sudoku with Ant Colony Optimization.” Says that sudoku game is a famous puzzle game that is very computationally challenging, so it requires the most powerful and most sophisticated form of algorithm to solve it.[2]
- Ghosh (2017) International journals published by IEEE did the research entitled “A Simple Recursive Backtracking Algorithm for Knight’s Tours Puzzle on Standard 8×8 Chessboard” says the backtracking algorithm is not entirely the best solution in the case of a knight’s tour game because it executes too long, making it less efficient and practical.[3]
- Schottlender (2014) Conduct research published by IEEE in international journals entitled “The Effect of Guess Choices on the Efficiency of a Backtracking Algorithm in a Sudoku Solver.” Using the initial element of a randomly generated sudoku puzzle is better in the backtracking algorithm compared to using numbers that are already available as starting elements. [4]
- Szabó (2014) on International journals published by IEEE did the research entitled “Creation of the Chips Placement Game with Backtracking Method in Borland Pascal” the application of backtracking algorithm in the creation of the chips placement game is very efficient because the algorithm can calculate all combinations with a concise code [5].

- Fermuller (2014) International journals published by IEEE did the research entitled "Semantic Games with Backtracking for Fuzzy Logics." implement the backtracking algorithm in Hintikka's classical game [6].
- Li et al., (2011) International journals published by IEEE did the research entitled "The Research on Departure Flight Sequencing Based on Improved Backtracking Algorithm." The application of the backtracking algorithm in scheduling has worked optimally and produced a way that will be and as expected [7].

3. Application Architecture

Analysis process is designed to provide an overview of all processes that occur as well as provide an overview of the parts of the system being designed. In this study, the process flow shown by a flowchart, which is an algorithm flowchart. The backtracking algorithm flowchart is a data flow based on an existing algorithm. In this flowchart, the running algorithm process is explained from the beginning to the end. The backtracking algorithm's diagram is illustrated in figure 1.

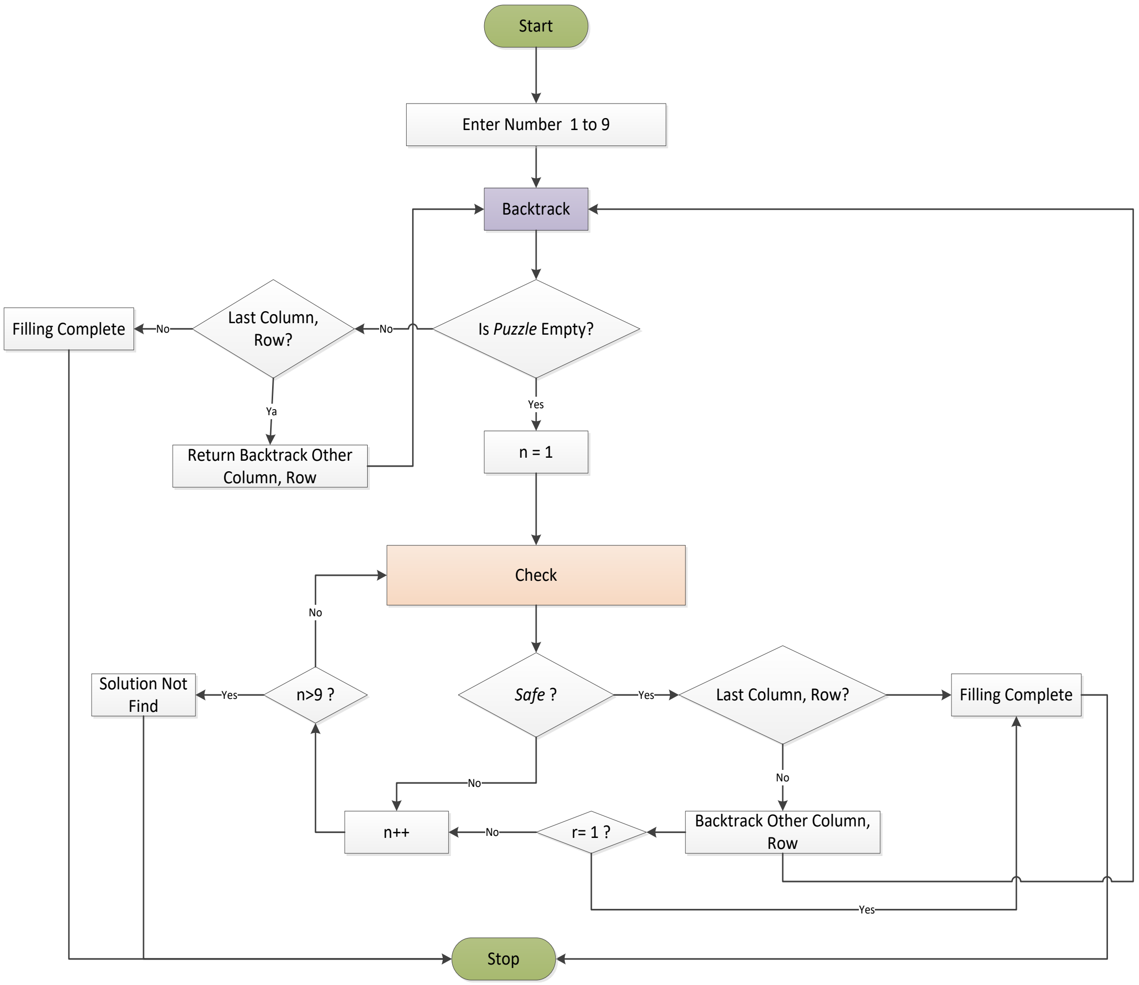


Figure 1. Algorithms Flowchart

4. Discussion

4.1. System Testing

System testing is done by running the Sudoku program using the Android platform. Several functions can be run on the system, including generating puzzles, checking the puzzle, and running the backtracking algorithm. Implementation of the operation carried out at *Smartphone android* with the following specifications:

- *Model ASUS T00F API 19.*
- *CPU Quad-core 1.2 GHz Cortex-A7*
- *Memory RAM 2048 MB*

This process is shown in figure 2, figure 3 and figure 4.

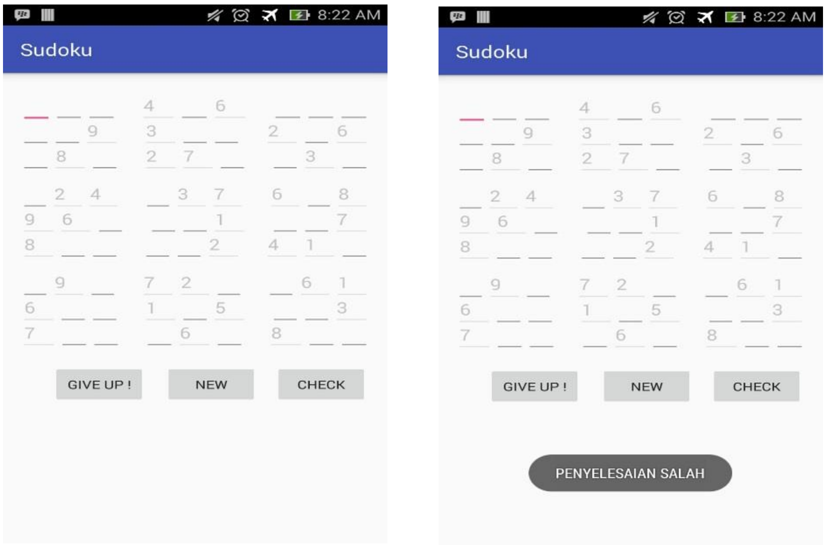


Figure 2. Generate Puzzle Sudoku Process and “CHECK” Button

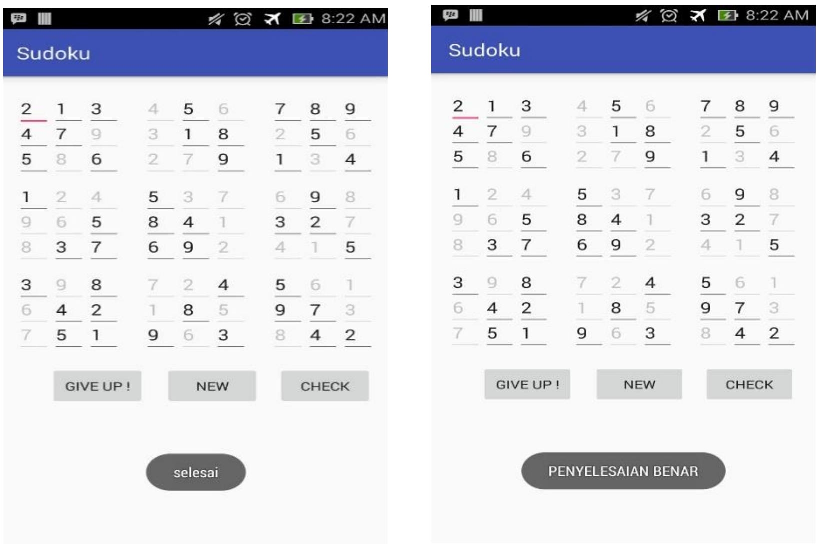
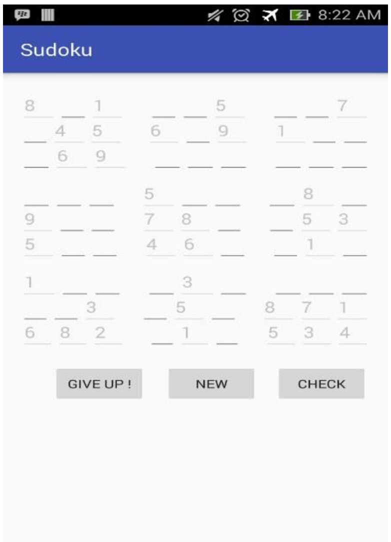


Figure 3. Backtracking Algorithm and Solve Puzzle With Manually



8:22 AM

Sudoku

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

Figure 4. Generate New Puzzle Sudoku Process

This figure above shows the process of generating the initial Sudoku puzzle number until the next figure shows the “Checking” process while the puzzle is still empty, and the system sends a Toast "PENYELESAIAN SALAH." Then in the following picture shows the system working and completing the sudoku puzzle until it is correct, and the system sends Toast "PENYELESAIAN BENAR."

4.2. Algorithm Complexity

Algorithm complexity is done by finding the big theta of the existing pseudocode. This pseudocode is part of a program that has been designed according to the backtracking algorithm. The results of calculations of the time complexity of the backtracking algorithm using the big theta notation (average case) are presented in table 1.

Table 1. Time Complexity of Backtracking Algorithms

Pseudocode	**C**	**#**	**C#**
`int temp;`	C_1	1	C_1
`for(int i=0; i<9; i++) {`	C_2	n	C_2n
`  for (int j = 0; j < 9; j++) {`	C_2	n^2	C_2n^2
`    if(check[i][j]==0){`	C_3	n^2	C_3n^2
`      temp = grid[i][j] + 1;`	C_4	n^2	C_4n^2
`      maju = true;`	C_4	n^2	C_4n^2
`      for(int k = temp; k<=10; k++) {`	C_2	n^3	C_2n^3
`      if(isSafe(i, j, k)) {`	C_3	n^3	C_3n^3
`        grid[i][j] = k;`	C_4	n^3	C_4n^3
`        break; }}`	C_5	n^3	C_5n^3
`      if(grid[i][j]>9) {`	C_3	n^2	C_3n^2
`        grid[i][j] = 0;`	C_4	n^2	C_4n^2
`        j-=2;`	C_4	n^2	C_4n^2
`        maju=false; }`	C_4	n^2	C_4n^2
`      if(j<-1 && i>0) {`	C_3	n^2	C_3n^2
`        i--;`	C_4	n^2	C_4n^2
`        j = 7; }`	C_4	n^2	C_4n^2
`      else if(j<-1 && i==0){`	C_3	n^2	C_3n^2
`        return false; }}`	C_5	n^2	C_5n^2
`    else if(check[i][j]==1 && maju==false) {`	C_3	n^2	C_3n^2
`      j-=2;`	C_4	n^2	C_4n^2
`      if(j<-1 && i>0) {`	C_3	n^2	C_3n^2
`        i--;`	C_4	n^2	C_4n^2

`        j = 7; }`	C_4	n^2	C_4n^2
`      else if(j<-1 && i==0) {`	C_3	n^2	C_3n^2
`        return false; }}}}`	C_5	n^2	C_5n^2
`return true;`	C_5	1	C_5

From the running time calculations in the table we get:

$$T(n) = C_1 + C_2n + C_2n^2 + C_3n^2 + C_4n^2 + C_4n^2 + C_2n^3 + C_3n^3 + C_4n^3 + C_5n^3 + C_3n^2 + C_4n^2 + C_4n^2 + C_4n^2 + C_3n^2 + C_4n^2 + C_4n^2 + C_3n^2 + C_5n^3 + C_3n^2 + C_4n^2 + C_3n^2 + C_4n^2 + C_4n^2 + C_3n^2 + C_5n^3 + C_5$$

$$= C_1 + C_2n + C_5 + (C_2 + 7C_3 + 10C_4 + 2C_5)n^2 + (C_2 + C_3 + C_4 + C_5)n^3$$

$$= n^3 + n^2 + n$$

$$T(n) = \boldsymbol{\theta}(n^3)$$

5. Conclusion And Future Works

The Sudoku puzzle game application with backtracking algorithm has worked and is running well, but for further research, it is better to raise more complex cases, such as a more complicated artificial intelligence game so that the algorithm used works more optimally. Likewise, the layout and appearance of the user interface are more developed. At this writing, the writer deliberately does not pay attention to the display interface because one of them is wanting to show more algorithms that work in the system instead of highlighting the game itself. The other suggestions for the next research are adding another algorithm as a comparison with the backtracking algorithm in solving the same case or different cases and comparing the running time and algorithm complexity of the two.

References

[1] T. H. Cormen, Ed., *Introduction to algorithms*, 3rd ed. Cambridge, Mass: MIT Press, 2009.

[2] H. Lloyd and M. Amos 2019 Solving Sudoku with Ant Colony Optimization," *IEEE Trans. Games*, pp. 1–1, 2019.

[3] D Ghosh and U Bhaduri 2017 A simple recursive backtracking algorithm for knight's tours puzzle on standard 8×8 chessboard *Int.Conf. on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi pp. 1195–1200.

[4] M Schottlender 2014 The effect of guess choices on the efficiency of a backtracking algorithm in a Sudoku solver *Long Island Systems, Applications and Technology (LISAT) Conference* armingdale, NY, USA pp. 1–6.

[5] R Szabo 2014 Creation of the chips placement game with backtracking method in Borland Pascal *11th International Symposium on Electronics and Telecommunications (ISETC)*, Timisoara, Romania pp. 1–4.

[6] C G Fermuller 2014 Semantic Games with Backtracking for Fuzzy Logics *IEEE 44th International Symposium on Multiple-Valued Logic*, Bremen, Germany pp. 38–43.

[7] N Li, L Liu, and X Xu 2011 The research on departure flight sequencing based on improved backtracking algorithm *Proc. Int. Conf.on Transportation, Mechanical, and Electrical Engineering (TMEE)* pp. 29–33.

[8] P Siswantara, R Riris Diana, and Muthmainnah 2019 The smart adolescent reproductive health promotion strategy based on android *Opcion*, vol. 35, no. 24, pp. 11701184

[9] R T Yunardi, Q Istiqomah, and R Latifah 2019 Contrast-enhanced Based on Abdominal Kernels for CT Image Noise Reduction *Proc. Int. Conf. Artif. Intell. Inf. Technol* pp. 293–296 doi: 10.1109/ICAIIT.2019.8834645.