**RMIT University**

**School of Science and Technology (SST)**

**COSC2658 – Data Structures & Algorithms**

**Assignment 2**

**Algorithm Design and Implementation**

Lecturer: Dr. Minh Ngoc Dinh

Students: Ngo Duy Anh – s3695424

Duong Tuan Dat - s3636739

Ho Duy Hoang –s3672214

Date: 19/01/2021

Table of Contents

[1 Project Overview 3](#_Toc61568990)

[2 Algorithms 3](#_Toc61568991)

[3 Example 3](#_Toc61568992)

[4 References 3](#_Toc61568993)

# Project Overview

1. **Background Information**

“Guess my number” has been one of the most popular games of study in the field of data structure and algorithms due to its large but not overwhelming solution space size. The main goal of this game is to design and build a program that can deliver a guess of secret code consisting four-digit number. During the project, a competition will be held in class to figure out the most efficient algorithms among each team. Before attempting to characterize and find the solution for Guess my number, let define the game’s notation and rules.

To start a round, each team will generate a number with 4 places and 10 possible digits, denoted ‘0’- ‘9’, however, this number need to be placed on the range of 1000-9999. And it will be implemented to the program of remaining team for the guessing purpose. One each move, the program will response to the player by indicating the following manner:

* Strike: a correct digit in the guess and in the right place.
* Hit: a correct digit in the guess but out of order.
* Miss: no digit of the guess is anywhere in the number.

Hence, the basic principle is that the program of each team need to guess the other’s program delivery number and vice-versa. At the end of the game, the winner is the team who solves his opponent’s secret code with fewer guesses.

By doing this project, our team can achieve several learning outcomes such as theory-based understanding of algorithms for the fundamentals applicable purpose. An insightful understanding combining with data structure methods which had been discussed during the course to analyse effectively solutions for the real-world problems.

1. **Literature Review**

Based on the available researched specifications, “Guess my number” is nearly similar to one of the classic puzzle game – Mastermind. Therefore, there are many approaches that can use to solve the game. In 1977, Donald Knuth demonstrated an algorithm that can solved the pattern in five moves or fewer. His technique uses a greedy strategy that can progressively reduce the number of possible patterns at each step [1]. In 1993, Kenji Koyama and Tony W. Lai found a method by using an exhaustive depth-first search to reduce average number of turns needed with an average of 5625/1296 = 4.3403 turns, with a worst-case scenario of six turns [2]. Swaszek (1999-2000) had found out an algorithm that can gives a surprisingly short average game length of 4.638 [3]. His practical strategy has made a random guess from the set of remaining candidate score sequences without requires any complicated record-keeping or use of a computer. In this project, our team decided to apply the Knuth methods combining with some special data type and algorithms to overcome the game. The detail will be discussed on the section below.

# Algorithm

1. **The guessing strategy – Knut algorithm**

The algorithm of Donald Knuth works as follow:

* We will create two ArrayList that will store the possible integer number between 1000 and 9999.
* The algorithm will start with the initial guess of 1123. It will play a guess to get a response of ‘Strike’, ‘Hit’, or ‘Miss’.
* If the response returns four ‘Strike’, the game is won, the algorithm terminates.
* If it is not, we will remove from the above ArrayList any number that not given the same response if it were the code.
* Then, we will follow the Mini-Max technique to find a next guess. After eliminate the numbers that not return the same value with the secret code, it will calculate the score of remaining numbers. The score of a number is depended on the minimum number of possibilities it might eliminate from ArrayList. Then, the program will select one from the set of guesses with the maximum score. (Notice Knuth follows the convention of choosing the guess with the least numeric value e.g. 3456 is lower than 4567 [1])
* The program will execute a loop to repeat from step 3.

1. **The special algorithm**

During the task, Mini-Max Algorithm has been combined with the Knuth algorithm to deliver an efficient guess. It uses recursion and backtracking algorithm to performs a depth-first search algorithm for the exploration of the complete game tree [4]. It will go the way down to the terminal node of the tree, then backtrack the tree as the recursion. The main disadvantages of this algorithm is that it gets really slow for a complex game with has a huge branching factor, and need to deliver lots of choices to decide. That is why the execution time of the first guesses are usually sizeable. When the size of ArrayList is narrowed, the processing time will be faster.

1. **The Special data structure**

**ArrayList**

Data type ‘ArrayList’ has been used to store the possible guesses. The ‘ArrayList’ class is a resizable array, means that we can add or remove elements from an array [5]. When working with Knuth algorithm, since the system will remove any wrong numbers from an initial possible guesses array, ArrayList is an appropriate and flexible than the traditional array.

**HashMap**

The HashMap in java is a class that extends AbstractMap class and implementation of Map Interface in Collections Framework, so it will have some similar characteristics and methods with Map [6]. HashMap is used to store elements as "key / value". Key and value are any data type, and we can try to access HashMap values with a specific key. In the system, it is created to store the chance of possible number and its frequency. By iterating over HashMap, if the case has already existed in HashMap then the count of it will increase by 1. Otherwise the program will add the chance to HashMap.

# Example

Đợi bản final xong r mọi người chạy vài case đặc biệt r đưa vô

# References

[1] Knuth, Donald (1976–77). "The Computer as Master Mind" (PDF). J. Recr. Math. (9): 1–6.

[2] Koyama, Kenji; Lai, Tony (1993). "An Optimal Mastermind Strategy". Journal of Recreational Mathematics (25): 230–256.

[3] Mastermind -- from Wolfram MathWorld", *Mathworld.wolfram.com*, 2021. [Online]. Available: https://mathworld.wolfram.com/Mastermind.html. [Accessed: 14- Jan- 2021]

[4] Artificial Intelligence | Mini-Max Algorithm - Javatpoint", *www.javatpoint.com*, 2021. [Online]. Available: https://www.javatpoint.com/mini-max-algorithm-in-ai. [Accessed: 14- Jan- 2021]

[5] ArrayList in Java - javatpoint", *www.javatpoint.com*, 2021. [Online]. Available: https://www.javatpoint.com/java-arraylist. [Accessed: 14- Jan- 2021]

[6] HashMap in Java - javatpoint", *www.javatpoint.com*, 2021. [Online]. Available: https://www.javatpoint.com/java-hashmap. [Accessed: 14- Jan- 2021]