**CHAPTER 1**

**INTRODUCTION**

Technology is important in today’s world because it serves a variety of functions in many of the most important aspects of modern society, like education, communication, business and scientific progress. In computer science, there are many branches of technology like analysis of algorithm (AA), operation research (OR), artificial intelligent (AI), compiling technique (CT), operation system (OS), software engineering (SE) and so on. Analysis of algorithms is the determination of the amount of resources (such as time and storage) necessary to [execute them](https://en.wikipedia.org/wiki/Computation). Most [algorithms](https://en.wikipedia.org/wiki/Algorithm) are designed to work with inputs of arbitrary length. Usually, the efficiency or running time of an algorithm is stated as a function relating the input length to the number of steps ([time complexity](https://en.wikipedia.org/wiki/Time_complexity)) or storage locations ([space complexity](https://en.wikipedia.org/wiki/Space_complexity)). Algorithm analysis is an important part of a broader [computational complexity theory](https://en.wikipedia.org/wiki/Computational_complexity_theory), which provides theoretical estimates for the resources needed by any algorithm which solves a given [computational problem](https://en.wikipedia.org/wiki/Computational_problem). These estimates provide an insight into reasonable directions of search for [efficient algorithms](https://en.wikipedia.org/wiki/Algorithmic_efficiency).

Storing data is important in AA. There are various forms of data structure to store data. Among them, hash table is useful for unsorted data. Hash table is a data structure used to implement an associative array, a structure that can map keys. Hashing is based on the idea of distributing keys among a one-dimensional array H[0..m-1] called hash table. The distribution is done by computing, for each of the keys, the value of some predefined function h called the hash function. This function assigns an integer between 0 and m-1, called the hash address, to a key. Every hashing scheme must have a collision resolution mechanism. This mechanism is different in the two principal versions of hashing: open hashing (also called separate hashing) and closed hashing (also called open addressing). This system applies separate chaining mechanism (Open Hashing) in which keys are stored in linked list attached to cells of a hash table. Each list contains all the keys hashed to its cell.

Everywhere, nouns participate an important part of every phrase like handbill, university names, community names, and so on. Numbers of nouns in a modern dictionary can be estimated as one third of a whole dictionary book. So it can be said nouns are very helpful for four skills in English (Reading, Writing, Speaking and Listening). The system contains many nouns vocabulary and the user can only type nouns. Initially, the system will generate a word in random order. The user must enter a word, its first character is matched with the last character of random word. The system contains minus marking scheme for duplication, mismatch, not noun words. One mark will be gained if the user enters a correct noun which is consistent with the rules of the system. The system has more than one level to test how much the user has noun vocabularies in English.

* 1. **Objectives**

Objectives of the system are as follow

* To get more interesting in English
* To be able to think nouns in a moment
* To learn noun vocabularies lightly from playing game
* To make students feeling happy in studying English
* To test the user’s ready wit and typing skill
* To understand separate chaining mechanism and how to use it.
  1. **Analysis of Algorithm**

An algorithm is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in finite amount of time. The reference to “instructions” implies that there is something or someone capable of understanding and following the instructions given. Algorithms play the central role in both the science and the practice of computing. Algorithmic is a branch of computer science that consists of designing and analyzing computer algorithms. The “design” pertains to the description of algorithm at an abstract level by means of a pseudo language, and proof of correctness that is, the algorithm solves the given problem in all cases. The “analysis” deals with performance evaluation (complexity analysis).

To analyze an algorithm:

* Develop a good implementation.
* Identify unknown quantities representing the basic operations.
* Determine the cost of each basic operation.
* Develop a realistic model for the input.
* Analyze the frequency of execution of the unknown quantities.
* Calculate the total running time:

In order to analyze an algorithm, there are two kinds of algorithm efficiency: time efficiency, indicating how fast the algorithm runs and space efficiency, indicating how much extra memory it uses. Another desirable characteristic of an algorithm is simplicity. The term “analysis of algorithms” is usually used in a narrower, technical sense to mean an investigation of an algorithm’s efficiency with respect to two resources: running time and memory space. In theoretical analysis of algorithms, it is common to estimate their complexity in the asymptotic sense, i.e., to estimate the complexity function for arbitrarily large input. Big Oh (O) notation, Big-omega (Ω) notation and Big-theta (Θ) notation are used.

**Θ-Notation** (Same order)

This notation bounds a function to within constant factors. It can be written that f(n) = Θ(g(n)) if there exist positive constants n0, c1 and c2 such that to the right of n0 the value of f(n) always lies between c1 g(n) and c2 g(n) inclusive.

**Ο-Notation** (Upper Bound)

This notation gives an upper bound for a function to within a constant factor. It can be written that f(n) = O(g(n)) if there are positive constants n0 and c such that to the right of n0, the value of f(n) always lies on or below c g(n).

**Ω-Notation** (Lower Bound)

This notation gives a lower bound for a function to within a constant factor. It can be written that f(n) = Ω(g(n)) if there are positive constants n0 and c such that to the right of n0, the value of f(n) always lies on or above c g(n).

There are three cases to analyze an algorithm: worst case, average case and best case. In the worst case analysis (usually done), upper bound is calculated on running time of an algorithm. The case that causes maximum number of operations to be executed must be known. In average case analysis (sometimes done), all possible inputs and calculate computing time for all of the inputs must be taken. Sum all the calculated values and divide the sum by total number of inputs. In the best case analysis (**bogus**), lower bound is calculated on running time of an algorithm. The case that causes minimum number of operations to be executed must be known.

* 1. **Overview of the System**

The system named “Hash Based Duplicate Nouns Checking System” uses open hashing (separate chaining). It is implemented using Java Programming Language (J2SE). The system contains many nouns vocabulary and the user can only type nouns. Initially, the system will generate a word in random order. The user must enter a word whose first character is matched with the last character of previous word. The input will be stored in linked list using open hashing (separate chaining) for avoiding collision in hashing. The system can only check the spelling of words whether it is same with the previous words and will ignore the meaning of nouns. The system contains minus marking scheme for duplication, mismatch, not noun words. One mark will be gained if the user enters a correct noun which is consistent with the rules of the system. The system will save the scores of the user. The system has more than one level to test how much the user has noun vocabularies in English.

* 1. **Organization of the Thesis**

In this thesis, there are four chapters.

Chapter 1 describes about analysis of algorithm, IT field, design and analysis of algorithm background, system overview and objectives of the thesis.

Chapter 2 displays about design and analysis of algorithm in detail and describes the algorithm used in the system.

Chapter 3 presents the design of the system, implementation of the system by using open hashing (separate chaining) based on input values.

Chapter 4 discusses the conclusion and benefits of the system.