



Data Structures Project Proposal
(SOFE 2715U)
Group 37: Project 12
Power of Two-Choices Hashing

Student Names	Student ID's
Izien Iremiren	100651421
Walid Safi	100623815
Faisal Pindar	100629476

EXECUTIVE SUMMARY

Hashing is the process of converting an input of any length or size into a fixed - length text string. This is usually done using a mathematical function. This is basically the converting of any text no matter how long, into an array of numbers and letters through an algorithm. The text to be converted is referred to as the input, while the algorithm it is passed to is called a hash function and finally the output is called the hash value. The hash values have to always be unique, this means it should be impossible to produce the hash value even when inserting different inputs. The hashing speed is also an important factor. It should be able to produce a hash value in a very fast processing time. Finally, the hash function/algorithm needs to be secure, thus making it difficult and preferably impossible to determine the input through the hash value.

2-choice hashing makes use of two hash functions $h_0(x)$ and $h_1(x)$, these two hash functions should be independent and have no correlation to each other. Having two hash functions allows any integer x to have up to two potential locations to be stored based on the values of the respective outputs, $h_1(x)$ and $h_2(x)$. It is important to note that, although there are two hash functions, there is only one table; both hash functions map to locations on that table. In this project, we shall use the instance of placing balls(the index key) into 2 different bins(hash tables) $h_0(x)$ and $h_1(x)$. When placing a ball, we go according to the bins and place the ball in the bin with the fewest entries, this could be referred to as the hash function. However, the placement of a ball is independent of the placement of the previous ball. Also, as the process is running, we check the load of the bins and if 1 bin contains fewer balls, then we place the next ball to the fewer containing bin. This process can be referred to as the power of 2 choices.

Our goal in this project is to develop an algorithm that does this sorting in the fastest way and time possible. A major hurdle would be for the program to be able to go through the keys and also check the table at the same time in order to know which index key contains the fewest entries and insert the next elements inside it. In situations where both buckets have the same amount of keys, the program should know to put the next key in $h_0(x)$ which shall also be a task we need to tackle.

PROJECT PLANNING

For this project, our group will utilize Github. Each group member will be able to create a local repository of the source codes framework to develop the program. It will also allow us to organize test cases and class diagrams. Github will allow us to easily access and contribute to the project while maintaining excellent documentation as Github tracks changes in the code/repository.

- **Define the problem:** Derive a good description of the input-output relationship
 - Create a map implementation.
 - Create a class diagram representing the relations between classes.
- **Create a working algorithm:** Define the sequence of computational steps to transform the input into the output.
 - Test the algorithm based on different cases using a tester class.

- **Find a suitable data structure:** Discover an organized method of inserting and searching for data.
- **Compare algorithms:** Look through the advantages and disadvantages of different methods. And find the most optimized approach.
 - Calculate the time complexity of the program.

Group Member	Responsibility
Walid	Code, Test Cases, Diagrams, Time Complexity Calculations
Faisal	Code, Test Cases, Time Complexity Calculations
Izien	Code, Test Cases, Time Complexity Calculations