BLG335E

ANALYSİS OF ALGORİTHMS I

CRN:10824

PROJECT 03

INSTRUCTOR:

ZEHRA ÇATALTEPE

STUDENT:

HÜSEYİN ERDOĞAN

040100054

Introduction

Aim of this homework is learning how to build hash table. First program reads the "insert.txt" file and according to value of m(array size) and hash type(lineer, double, universal) program does insert operations. After program reads the "search.txt" file and according to value of m(array size) and hash type(lineer, double, universal) program does search operations. After termination, program gives us insert and search collision.

Development and Runtime environment

Project is developed in Microsoft Windows 7. Microsoft Visual, Dev-C++ and g++ are used for compiling.

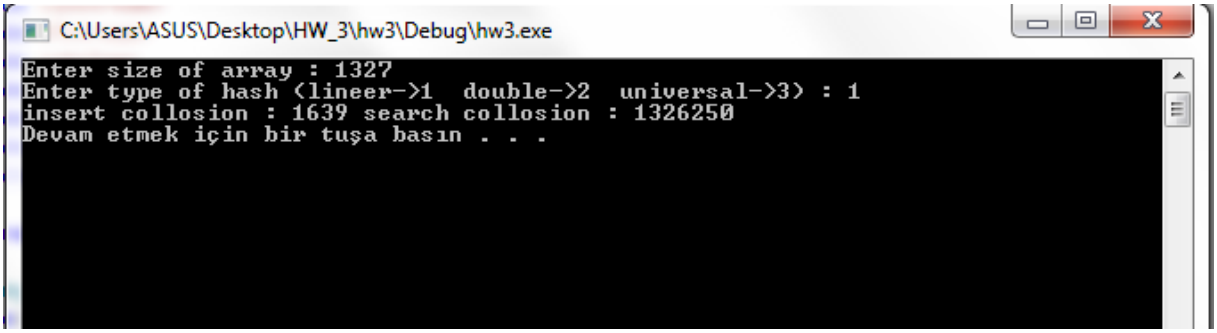
Program has one class. This class consists insert and search functions.

In my_hash class:

- `int *hash_array` **: for storing elements in array**
- `int sizeofarray` **: for storing array size**
- `int insert_collision` **: for storing insert collision**
- `int search_collision` **: for storing search collision**
- `int a0` **: used in universal hashing (between 0 and m-1)**
- `int a1` **: used in universal hashing (between 0 and m-1)**
- `int a2` **: used in universal hashing (between 0 and m-1)**
- `my_hash(int t_size)` **: for defining size of array, assigning insert and search collision to "0", assigning a0,a1,a2 to random value.**

- `~my_hash()` : **for deleting array**
- `void insert_lineer_hashing(int element)` : **for inserting element to hash table with using lineer hashing**
- `void insert_double_hashing(int element)` : **for inserting element to hash table with using double hashing**
- `void insert_universal_hashing(int element)` **: for inserting element to hash table with using universal hashing**
- `void search_lineer_hashing(char *filename)` **for searching element to hash table with using lineer hashing**
- `void search_double_hashing(char *filename)` : **for searching element to hash table with using double hashing**
- `void search_universal_hashing(char *filename)` : **for searching element to hash table with universal lineer hashing**

When program is started, user enters to size of array. After entering size of array, user selects hash type. After execution program writes insert and search collision to screen.



Required Table in Report

- **Insert Collision**

	linear	**double**	**universal**
m = 1327	1639	861	802
m = 2657	279	271	251

- **Search Collision**

	linear	**double**	**universal**
m = 1327	332976	332407	664477
m = 2657	664465	664450	1328767

Universal hashing's collision values are changing because of random numbers. But insert collision values of universal hashing are lower than linear and double hashing thaks to random values. Search collisions of linear and double hashing are almost same and lower than universal hashing. Because of random values, searching is dificult for universal hashing.