National University of Computer and Emerging Sciences

Lab Manual

for

Data Structure

Course Instructor	Ms. Abeeda Akram
Lab Instructor(s)	Mr. Sohaib Ahmad Ms. Ammara
Section	BCS-3G
Semester	FALL 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Lab Manual 11

Objectives:

After performing this lab, students shall be able to revise:

- ✔ Hash Table

Problem 1

Implement a Hash table class for Hashing of Integer keys, you will have a hash table array for storage with its size. Pick an odd number for hash table size.

```
class HashTable
{
private:
        int size;// Size will always be odd
        int* arr;
public:

};
```

1. Design a constructor which will take the following parameters.
    a. The boolean value flag for rehashing, if it is 0 then you will not do rehashing, and if it is 1 you will double the size of the hash table if the loading factor is above 0.5.
    b. The integer value for the selection of collision resolution methods.
2. Implement the following functions in the hash table class.
    a. Insert
    b. Delete
    c. Update
    d. Access

    NOTE: Print probe sequences generated for different keys on insertion and access.
3. Use the mod function for hashing.
4. For collision resolution you will implement the following four functions.
    a. Linear probing i = p(k, i),
    b. Linear probing with steps ci = p(k, i), where c = prime number smaller than Hash Table size.
    c. Pseudo-random probing for collision resolution. perm[i] = p(k, i)
    d. Double hashing with function hf2 = 1+ key % (size-2)

Now create 4 different objects of hash tables and place the same keys in all 4 objects. But for collision resolution.

1. Use Linear probing on the first object.
2. Use Linear probing with steps on the second object.
3. Use pseudo-random probing on the third object.
4. Use double hashing on the fourth object.
5. Calculate and compare the average access cost of data without rehashing.
6. Calculate and compare the average access cost of data with rehashing.

Problem 2

Implement a Hash table class for Hashing of Strings.

1. Implement the following functions in the hashtable class.
    a. Insert
    b. Delete
    c. Update
    d. Access
2. Polynomial function for hash code generation where a = 33.
3. Use chaining for collision resolution.

Application

Store Air pollution index of 50 Major cities of Pakistan

- **Key** City Name
- **value** air pollution index percentage.

Use hash table object functions for updating and access of data.