**CS 201 Data Structures II – Spring 2024**

**Instructor: Dr. Muhammad Mobeen Movania**

**Quiz 4 - Solution**

**Name:** _____ **Date:** _____

**Regn. No. :** _____

**There are two questions in this quiz. Each question carries 5 marks.**

Q1) Let us consider the quadratic probing scheming for a hash table of size 7 with the hash function as:

$h(x) = (x + i^2) \ mod \ 7$, where i = 0, 1, 2, 3… in case of a collision.

Using the properties of a good hash function, prove that the above function do not form a good hash function. (5 marks)

To prove that the given hash function does not map to all addresses within the hash table, we can analyze its behavior for a few values of x.

Lets start with x=0 and i=0, h(0) gives 0. When i=1, the hash value becomes $(0+1^2)$ mod 7=1. Similarly, when i=2, the hash value becomes $(0+2^2)$ mod 7=4. We can see that for x=0, the hash values only cover a subset of the hash table, not all addresses.

Similarly, for x=1, the hash values would be ,3,6,3,1,1,…, and for x=2, the hash values would be 2,5,3,2,2,…, which again do not cover all addresses within the hash table.

We can continue this analysis for other values of x, and we will find that the hash function does not map to all addresses within the hash table for any x. Hence, it violates the property that a hash function must map to all addresses within a hash table.

Q2) Consider the quadratic probing scheming for a hash table of size 7 with the hash function as:

$h(x) = (x \pm i^2) \ mod \ 7$, where i = 0, 1, 2, 3… in case of a collision.

Note that this function alternates between adding $i^2$ for first probe and subtracting $i^2$ for second probe and so on. Populate the given data into the appropriate location in the given hashtable.

Ans.

Data = {13,7,2,14,8,32}

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Key | 7 | 14 | 2 | | 8 | 32 | 13 |

For 13, 7, 2, no collision

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Key | 7 | | 2 | | | | 13 |

For 14, 14% 7=0 we have collision as at 0 we have 7, so we probe by using i=1, i^2=1 so we add 1 to index 0, the initial index we got from 14%7. We get 1 so 14 is stored at index 1.

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Key | 7 | 14 | 2 | | | | 13 |

Next value is 8, 8%7==1, we have collision since we have 14 at index 1, we again take i=1, i^2=1, so we add 1 to initial index we got from 8%7==1 to get 2. We see at index 2, there is collision due to 2, so we hash with i=2 and i^2=4 which will be subtracted from the initial index of 1 to get -3. We mod 7 to get 4 which is where we store 8.

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Key | 7 | 14 | 2 | | 8 | | 13 |

For final value 32, we % 7 to get 4, at 4 there is 8 so we get collision, we add i^2=1 since i=1 to get 5. We then mod 5 with 7 to get 5. This is where 32 is stored.

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Key | 7 | 14 | 2 | | 8 | 32 | 13 |