**Question 1:**

Given a hash table with m=11 entries and the following hash function h1 and step function h2:

h1(key) = key mod m

h2(key) = {key mod (m-1)} + 1

Insert the keys {22, 1, 13, 11, 24, 33, 18, 42, 31} in the given order (from left to right) to the hash table using each of the following hash methods:

a. Linear-Probing with h1 ⇒ h(k,i) = (h1(k)+i) mod m

b. Quadratic-Probing with h1 ⇒ h(k,i) = (h1(k)+i2) mod m

c. Separate Chaining with h1 ⇒ h(k) = h1(k)

c. Double-Hashing with h1 as the hash function and h2 as the step function ⇒ h(k,i) = (h1(k) + ih2(k)) mod m

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Linear Probing | Quadratic Probing | Double Hashing | Separate Chaining |
| 0 | 22 | 22 | 22 | 33->11->22 |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 13 | 13 | 13 | 24->14 |
| 3 | 11 | 24 |  |  |
| 4 | 24 | 11 | 11 |  |
| 5 | 33 |  | 18 |  |
| 6 |  |  | 31 |  |
| 7 | 18 | 18 | 24 | 18 |
| 8 |  |  | 33 |  |
| 9 | 42 | 33 | 42 | 31->42 |
| 10 | 31 | 42 |  |  |
| 11 |  | 31 wont be added |  |  |

For the previous h1 and h2, is it OK to use h2 as a hash function and h1 as a step function?

No, since h1 can return 0 and h2 skips the entry 0. For example, key 11-> h1(11) = 0 , and all the steps are of length 0, h(11,i) = h2(11) + i\*h1(11) = h2(11) = 2. The sequence of index addressing for key 11 consists of only one index, 2.

**Question 2:**

Given a hash table with the following properties:

Key: Point(x, y, z)

H(key) = (x +y – z) mod m

Initial table size = 5

### λ = 0.5 Compute hash code here

add( new Point(5, 6, 2), "C") --------------🡪 9

add( new Point(4, 5, 0), "D") --------------🡪 9

add( new Point(4, 0, 0), "F") --------------🡪 4

getValue(new Point(4, 0, 0)) --------------🡪 4

a. Linear-Probing with h1 ⇒ h(k,i) = (h1(k)+i) mod m

b. Quadratic-Probing with h1 ⇒ h(k,i) = (h1(k)+i2) mod m

c. Separate Chaining with h1 ⇒ h(k) = h1(k)

**After the second add**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Linear Probing | Quadratic Probing | Separate Chaining |
| 0 | ([x=5, y=6, z=2], C) | ([x=5, y=6, z=2], C) | ([x=5, y=6, z=2], C) -> ([x=4, y=5, z=0], D) |
| 1 | ([x=4, y=5, z=0], D) | ([x=4, y=5, z=0], D) |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |

**After the third add, λ = 3/5> 0.5 so we double the size of the array 10-> then we find the next prime number 11 then we re-add the values**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Linear Probing | Quadratic Probing | Separate Chaining |
| 0 |  |  |  |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 | ([x=4, y=0, z=0], F) | ([x=4, y=0, z=0], F) | ([x=4, y=0, z=0], F) |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 | ([x=5, y=6, z=2], C) | ([x=5, y=6, z=2], C) | ([x=5, y=6, z=2], C)->([x=4, y=5, z=0], D) |
| 10 | ([x=4, y=5, z=0], D) | ([x=4, y=5, z=0], D) |  |
| 11 |  |  |  |