**Assignment 7**

R-2.19 Draw the 11-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, using the hash function *h(i)* = (2*i* + 5) mod 11 and assuming collisions are handled by **chaining**.

R-2.20 What is the result of the previous exercise, assuming collisions are handled by **linear probing**?

R-2.21 Show the result of Exercise R-2.19, assuming collisions are handled by **quadratic probing**, up to the point where the method fails because no empty slot is found.

R-2.22 What is the result of Exercise R-2.19 assuming collisions are handled by **double hashing** using a secondary hash function *h’(k)* = 7 – (*k* mod 7)?

Answer:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12 | 44 | 13 | 88 | 23 | 94 | 11 | 39 | 20 | 16 | 5 |

h(k) = (2k+ 5) mod 11

Chaining:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 20 |  |  | 16-5 | 44-88-11 | 94-39 | 12-23 |  | 13 |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Linear Probing:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 39 | 20 |  | 16 | 44 | 88 | 12 | 23 | 13 | 94 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Quadratic Probing:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 20 | 16 | 11 | 39 | 44 | 88 | 12 | 23 | 13 | 94 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

5 --> No empty slot is found because after j=11 it repeats as the first 11 initial value.

Double Hashing:

|  |  |  |  |
| --- | --- | --- | --- |
| k | h(k) = (2k + 5) mod 11 | d(k) = 7 – (k mod 7) | Probes |
| 12 | 7 | 2 | 7 |
| 44 | 5 | 5 | 5 |
| 13 | 9 | 1 | 9 |
| 88 | 5 | 3 | 5\_8 |
| 23 | 7 | 5 | 7\_1 |
| 94 | 6 | 4 | 6 |
| 11 | 5 | 3 | 5\_8\_0 |
| 39 | 6 | 3 | 6\_9\_1\_4 |
| 20 | 1 | 1 | 1\_2 |
| 16 | 4 | 5 | 4\_9\_3 |
| 5 | 4 | 2 | 4\_6\_8\_10 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 23 | 20 | 16 | 39 | 44 | 94 | 12 | 88 | 13 | 5 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Give the pseudo-code description for performing a removal from a hash table that uses linear probing to resolve collisions. Why is it necessary to use a special marker to represent deleted elements?

Answer:

Algorithm removalLinearProbling(k)

item 🡨 findItem(k)

if item = NO\_SUCH\_KEY then

return NO\_SUCH\_KEY

o 🡨 item.element()

updateItem(k, AVAILABLE)

return o

Linear probing handles collisions by placing the colliding item in the next (circularly) available table cell. To avoid losing track of items that have been deleted, a special marker is used to represent them. Removing this marker can cause problems because the search will stop once it reaches an empty and unavailable block, preventing the discovery of any values that may have been placed in subsequent blocks.