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) = (2i + 5) mod 11 and assuming collisions are handled by chaining.

|  |  |
| --- | --- |
| **Key** | **H(K)** |
| 12 | 7 |
| 44 | 5 |
| 13 | 9 |
| 88 | 5 |
| 23 | 7 |
| 94 | 6 |
| 11 | 5 |
| 39 | 6 |
| 20 | 1 |
| 16 | 4 |
| 5 | 4 |
| **Index** | **Key** |
| 0 | ∅ |
| 1 | 20 |
| 2 | ∅ |
| 3 | ∅ |
| 4 | 16, 5 |
| 5 | 44, 88, 11 |
| 6 | 94, 83 |
| 7 | 12, 13 |
| 8 | ∅ |
| 9 | 13 |
| 10 | ∅ |

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

|  |  |  |
| --- | --- | --- |
| **Key** | **H(K)** | **Probes** |
| 12 | 7 | 7 |
| 44 | 5 | 5 |
| 13 | 9 | 9 |
| 88 | 5 | 5->6 |
| 23 | 7 | 7->8 |
| 94 | 6 | 6->10 |
| 11 | 5 | 5->0 |
| 39 | 6 | 6->1 |
| 20 | 1 | 1->2 |
| 16 | 4 | 4 |
| 5 | 4 | 4->3 |

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

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.

Quadratic probing formula: A [(i + j^2) mod N]

|  |  |  |
| --- | --- | --- |
| **Key** | **H(K)** | **Probes** |
| 12 | 7 | 7 |
| 44 | 5 | 5 |
| 13 | 9 | 9 |
| 88 | 5 | 5->6 |
| 23 | 7 | 7->8 |
| 94 | 6 | 6->10 |
| 11 | 5 | 5->3 |
| 39 | 6 | 6->1 |
| 20 | 1 | 1->2 |
| 16 | 4 | 4->2 |
| 5 | 4 | 4->??? |

Quadratic probing can't find empty slot for 5 because after j=11 it repeats as the first 11 initial value.

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

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) ?

H(k) = k mod 7

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

(2 \* 12 + 5) % 11 = 7

|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | **h(i)** | **H’(k)** | **Probes** |
| 12 | 7 |  | 7 |
| 44 | 5 |  | 5 |
| 13 | 9 |  | 9 |
| 88 | 5 | 3 | 3 |
| 23 | 7 | 5 | 1 |
| 94 | 6 |  | 6 |
| 11 | 5 | 3 | 8 |
| 39 | 6 | 3 | 4 |
| 20 | 1 | 1 | 2 |
| 16 | 4 | 5 | 0 |
| 5 | 4 | 2 | 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?

|  |
| --- |
| Algorithm removalLinearProbing**(**key**)**  Input**:** key to remove from  Output**:** remove and **return** the element    key**,** element**)** **<-** findElement**(**key**)**  If key **!=** NO\_SUCH\_KEY then  key **<-** AVAILABLE  **return** element  **return** NO\_KEY\_FOUND |
| Linear probing handles collision by putting the item in the next empty or available block. So, It is necessary to use a special marker to represent the deleted elements. If we remove it then we will not find the value that might be put in the next block after that because the search will end when it find an empty and not available block. |