**Assignment 11**

**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.**

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

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

12 (7), 44 (5), 13 (9), 88 (5), 23 (7), 94 (6), 11 (5), 39 (6), 20 (1), 16 (4), and 5 (4)

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

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

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

12 (7), 44 (5), 13 (9), 88 (5), 23 (7), 94 (6), 11 (5), 39 (6), 20 (1), 16 (4), and 5 (4)

**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.**

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

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

12 (7), 44 (5), 13 (9), 88 (5), 23 (7), 94 (6), 11 (5), 39 (6), 20 (1), 16 (4), and 5 (4)

Search A[(i + j2) mod N for j = 0, 1, 2, … until an 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)?**

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

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

12 (7), 44 (5), 13 (9), 88 (5), 23 (7), 94 (6), 11 (5), 39 (6), 20 (1), 16 (4), and 5 (4)

Double hask = 7 –(k mod 7)

**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?**

After deletion of an item, we introduce a special object called “Available” to help the linear probing and insert the new item if necessary. This will avoid accessing “null” while probing.

**C-4.10 Suppose we are given an n-element sequence S such that each element in S  
represents a different vote in an election, where each vote is given as an integer  
representing the ID of the chosen candidate. Without making any assumptions about  
who is running or even how many candidates there are, design an efficient algorithm to  
see who wins the election S represents, assuming the candidate with the most votes  
wins. Handle the possibility of multiple winners and do this using a Dictionary.  
Today specify your solution using pseudo code (tomorrow we will implement in  
JavaScript after discussing today’s pseudo code solution).**

Algorithm voteCount(S)

L:= new lookupTable

count :=0

for i=0 to i=L.size()-1 do

item:= findValue(L[i])

if item = “null” then

L.insertItem(L[i],1)

else

L.insertItem(L[i],count+1)

return getMax(L)

Algorithm getMax(L)

max:= L.first()

maxVal:= L.first().value()

p:= L.after(max)

if p.value() > maxVal then

max = p

maxVal = p.value()

p:= L.after(p)

return (max,maxVal)