CSCE 221 Cover Page
Homework Assignment #3
Due April 24 at 23:59 pm to eCampus

First Name: Pratik Last Name: Patel UIN:527004337

User Name: p.pratik99 E-mail address: p.pratik99@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more on Aggie Honor System Office website: `http://aggiehonor.tamu.edu/`

Type of sources				
People				
Web pages (provide URL)				
Printed material	Textbook			
Other Sources	Lecture Slides			

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.
On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.

Your Name: Pratik Patel Date: 04/24/2019

Homework 3 (100 points)

due April 24 at 11:59 pm to eCampus.

Write clearly and give full explanations to solutions for all the problems. Show all steps of your work.

Reading assignment.

- Hash Tables Chap. 9
- Heap and Priority Queue, Chap. 8
- Graphs, Chap. 13

Problems.

1. (10 points) R-9.7 p. 417

   Draw the 11-entry hash table that results from using the has function, $h(k) = (3k + 5)$ mod 11, to hash the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, assuming collisions are handled by chaining.

1)

$h(12) = (3(12)+5)\ \%\ 11 = 41\ \%\ 11 = 8$

$h(44) = (3(44)+5)\ \%\ 11 = 137\ \%\ 11 = 5$

$h(13) = (3(13)+5)\ \%\ 11 = 44\ \%\ 11 = 0$

$h(88) = (3(88)+5)\ \%\ 11 = 269\ \%\ 11 = 5$

$h(23) = (3(23)+5)\ \%\ 11 = 74\ \%\ 11 = 8$

$h(94) = (3(94)+5)\ \%\ 11 = 287\ \%\ 11 = 1$

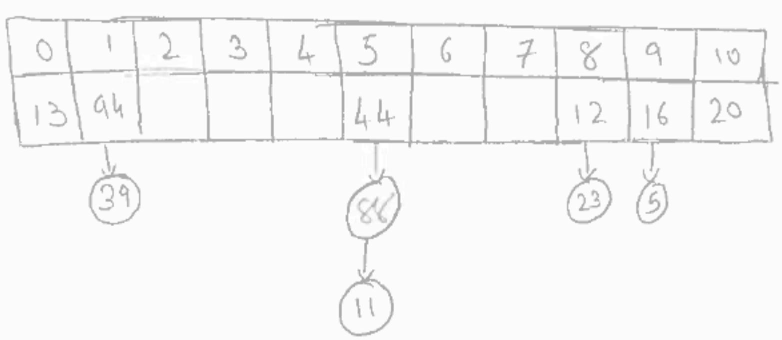
$h(11) = (3(11)+5)\ \%\ 11 = 38\ \%\ 11 = 5$

$h(39) = (3(39)+5)\ \%\ 11 = 122\ \%\ 11 = 1$

$h(20) = (3(20)+5)\ \%\ 11 = 65\ \%\ 11 = 10$

$h(16) = (3(16)+5)\ \%\ 11 = 53\ \%\ 11 = 9$

$h(5) = (3(5)+5)\ \%\ 11 = 20\ \%\ 11 = 9$



2. (10 points) R-9.8 p. 417

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

2)

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

h(12) = 8
h(44) = 5
h(13) = 0
h(88) = 5 ← Collision, so move to next available index (6)
h(23) = 8 ← Collision, so move to next available index (9)
h(94) = 1
h(11) = 5 ← Collision, so move to next available index (7)
h(39) = 1 ← collision, son move to next available index (2)
h(20) = 10
h(16) = 9 ← Collision, so move to next available index (3)
h(5) = 9 ← Collision, so move to next available index (4)

3. (10 points) R-9.10 p. 417

What is the result of Exercise R-9.7, when collisions are handled by double hashing using the secondary hash function $h_s(k) = 7 - (k \bmod 7)$?

3)

$h(12) = 8$

$h(44) = 5$

$h(13) = 0$

$h(88) = 5$ -> Collison, therefore,

$h(88) + h_s(88) = 5+3 = 8$

$\therefore h(8) = 7$

$\therefore h(88) = 7$

$h(23) = 8$ -> Collision, therefore,

$h(23) + h_s(23) = 8+5 = 13$

$h(13) = 0$

$\therefore h(23) = 0$

-> Again Collision,

$h(23) + 2h_s(23) = 8+10 = 18$

$h(18) = 4$

$\therefore h(23) = 4$

$h(94) = 1$

$h(11) = 5$ -> Collision, therefore,

$h(11) + h_s(11) = 5+3 = 8$

$h(8) = 7$

$\therefore h(11) = 7$

-> Again Collision,

$h(11) + 2h_s(11) = 5+6 = 11$

$h(11) = 5$

-> Again Collision,

$h(11) + 3h_s(11) = 5+9 = 14$

$h(14) = 4$

$\therefore h(11) = 4$

-> Again Collision.

$h(11) + 4h_s(11) = 5+12 = 17$

$h(17) = 1$

$\therefore h(11) = 1$

-> Again Collision,

$h(11) + 5h_s(11) = 5+15 = 20$

$h(20) = 10$

$\therefore h(11) = 10$

$h(39) = 1 \rightarrow$ Collision, therefore,

$$h(39) + h_s(39) = 1 + 3 = 4$$
$$h(4) = 6$$
$$\therefore h(39) = 6$$

$h(20) = 10 \rightarrow$ Collision, therefore,

$$h(20) + h_s(20) = 10 + 1 = 11$$
$$h(11) = 5$$
$$\therefore h(20) = 5$$

$\rightarrow$ Again Collision.

$$h(20) + 2h_s(20) = 10 + 2 = 12$$
$$h(12) = 8$$
$$\therefore h(20) = 8$$

$\rightarrow$ Again Collision,

$$h(20) + 3h_s(20) = 10 + 3 = 13$$
$$h(13) = 0$$
$$h(20) = 0$$

$\rightarrow$ Again Collision,

$$h(20) + 4h_s(20) = 10 + 4 = 14$$
$$h(14) = 4$$
$$\therefore h(20) = 4$$

$\rightarrow$ Again Collision.

$$h(20) + 5h_s(20) = 10 + 5 = 15$$
$$h(15) = 6$$
$$\therefore h(20) = 6$$

$\rightarrow$ Again Collision,

$$h(20) + 6h_s(20) = 10 + 6 = 16$$
$$h(16) = 9$$
$$\therefore h(20) = 9$$

$h(16) = 9 \rightarrow$ Collision, therefore

$$h(16) + h_s(16) = 9 + 5 = 14$$
$$h(14) = 4$$
$$\therefore h(16) = 4$$

$\rightarrow$ Again Collision

$h(16) + 2h_s(16) = 9 + 10 = 19$

$h(19) = 7$

$\therefore h(16) = 7$

$\rightarrow$ Again Collision,

$h(16) + 3h_s(16) = 9 + 15 = 24$

$h(24) = 0$

$\therefore h(16) = 0$

$\rightarrow$ Again Collision,

$h(16) + 4h_s(16) = 9 + 20 = 29$

$h(29) = 4$

$\therefore h(16) = 4$

$\rightarrow$ Again Collision

$\vdots$

$h(16) + 9h_s(16) = 9 + 45 = 54$

$h(54) = 2$

$\therefore h(16) = 2$

$h(5) = 9 \rightarrow$ Collision, therefore,

$h(5) + h_s(5) = 9 + 2 = 11$

$h(11) = 5$

$\therefore h(5) = 5$

$\rightarrow$ Again Collision,

$\vdots$

$\rightarrow$ Again Collision,

$h(5) + 8h_s(5) = 9 + 16 = 25$

$h(25) = 3$

$\therefore h(5) = 3$

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

4. (10 points) R-8.7 p. 361

An airport is developing a computer simulation of air-traffic control that handles events such as landings and takeoffs. Each event has a *time-stamp* that denotes the time when the event occurs. The simulation program needs to efficiently perform the following two fundamental operations:

- Insert an event with a given time-stamp (that is, add a future event)
- Extract the event with smallest time-stamp (that is, determine the next event to process)

Which data structure should be used for the above operations? Why? Provide big-oh asymptotic notation for each operation.

4) Binary Heap

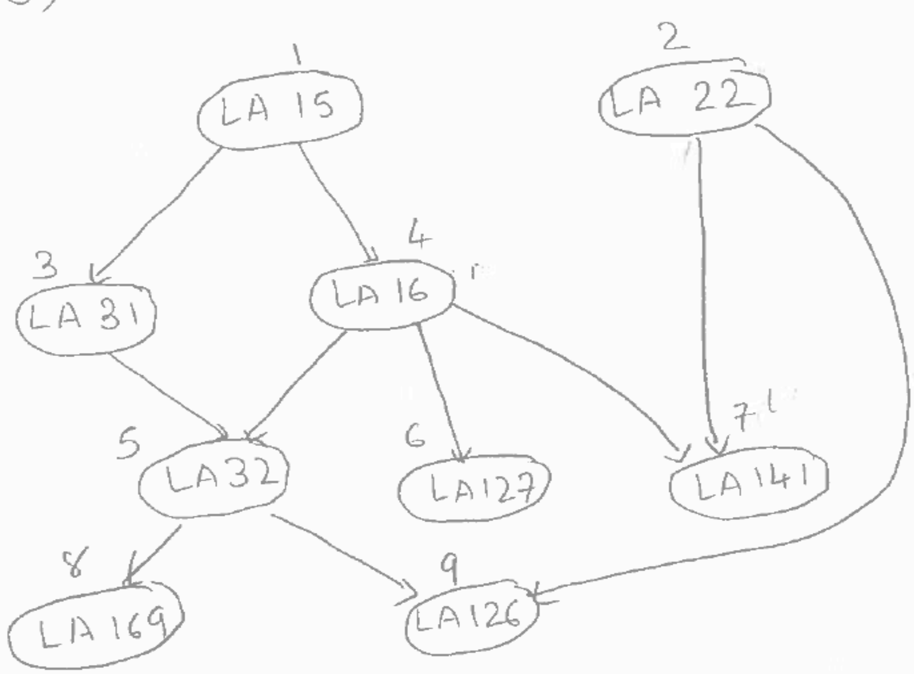
-> because we can remove the smallest element, which is the root and will take $\underline{O(1)}$ time.

-> Inserting elements will take $\underline{O(\log_2 n)}$ time

-> The keys will be the time stamp.

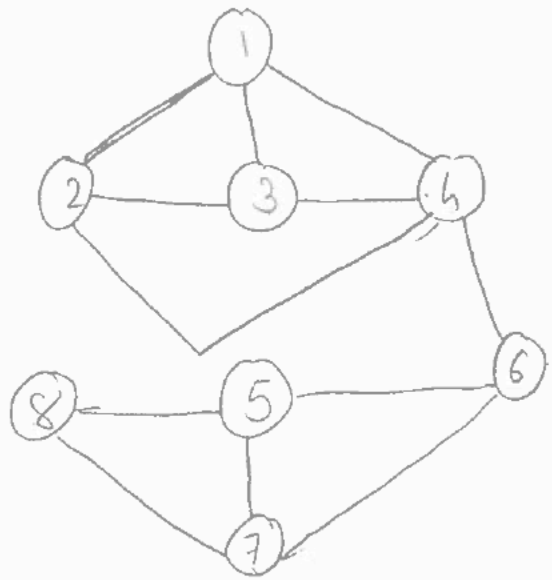
5. (10 points) R-13.5, p. 654

5)



6. (10 points) R-13.7, p. 655

6)



Starting Index-1

DFS : 1 2 3 4 6 5 8 7

BFS : 1 2 3 4 6 5 7 8

7. (10 points) R-13.16, p. 656

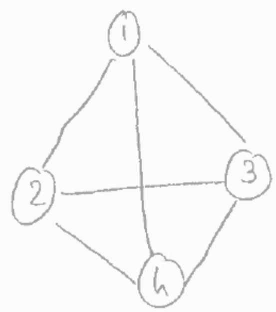
7)

```
Algorithm Dijkstra (G, v)

for all u ∈ G do    // Assume u is a vertex
    if (u = v)
        D[u] = 0
    else
        D[u] = ∞

Q ← Priority Queue
T ← tree
while (¬ Q.empty()) do
    u = Q.removeMin()
    if ( ¬ (u = v))
        T.addEdge(adj[u], u)
    for all z ∈ Q that z is adjacent to u do
        if (D[u] + w((u,z)) < D[z])
            D[z] = D[u] + w((u,z))
            change D[z] to key of vertex z in Q
            adj[z] = u

return T and the label D(u) of each vertex u.
```

8. (10 points) R-13.31, p. 657

8)

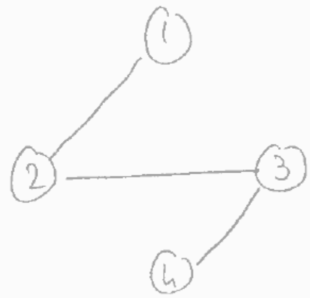
Graph



DFS starting from index 1,

1, 2, 3, 4

-> the tree will look like



∴ The search tree of a complete undirected graph looks like a path.

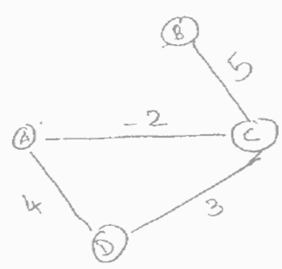
9. (10 points) C-13.10, p. 658

9)

We can use a similar algorithm to DFS. With necessory modifications we must not close circuits before all the edges are traversed. Whenever, there is a vertex with odd number of adjacent edges which aren't traversed yet, we can do a DFS and close the circuit. If we close a circuit and not all edges are traversed, we can then back track to the last vertex we saw with untraversed edges and make it a distinguished vertex. Then we try to find another edge that hasn't been traversed. In the end, we attach the old circuit to the new one. This algorithm takes $O(n+m)$ time because it is a modification of DFS. Attaching circuits would take $O(m)$ atmost.

10. (10 points) C-13.15, p. 659

10)



The shortest path from A to D = 4

→ The path $A \to C \to D$ gives us.

$$(-2) + 3 = 1$$

Hence it is incorrect, since $4 \neq 1$