CSCE 221 Cover Page
Homework 3
Due April 24 at midnight to eCampus

First Name: Cristian Last Name: Avalos UIN: 627003137

Username: avalos672918 E-mail address: avalos672918@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: Aggie Honor System Office

Type of Sources			
People	Friends (discussion)		
Web pages (provide URL)	http://www.cplusplus.com/reference/list/list/		
Printed material			
Other Sources			

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."

Cristian Avalos April 24, 2020

1. (10 points) For the following statements about red-black trees, provide a justification for each true statement and a counterexample for each false one.

   (a) A subtree of a red-black tree is itself a red-black tree.
   False: A red-black tree with a red root is not a red-black tree.

   (b) The sibling of an external node is either external or red
   True: A red-black tree leaves must have the same black depth. So if the external node was black, then the sibling must be black as well. If not, then it must be red because the leaves of that one must be null externals, keeping the same black depth.

   (c) There is a unique 2-4 tree associated with a given red-black tree.
   True: A node in a red-black tree with no children is uniquely represented as a 2-node in a 2-4 tree. A node in a red-black tree with one child is uniquely represented as a 3-node in a 2-4 tree. Lastly, a node in a red-black tree with two children is uniquely represented as a 4-node in a 2-4 tree.

   (d) There is a unique red-black tree associated with a given 2-4 tree.
   False: A 3-node in a 2-4 tree has two possible different representations in a red-black tree.

2. (10 points) Modify this skip list after performing the following series of operations: erase(38), insert(48, x), insert(24, y), erase(42). Provided the recorded coin flips for x and y. (boxes, subscripts, and arrows represent the order in which I go)

$-\infty$	—	—	—	—	—	$+\infty$
$-\infty$	—	17	—	—	—	$+\infty$
$-\infty$	—	17	—	—	42	$+\infty$
$-\infty$	—	17	—	—	42	$+\infty$
$-\infty$	12	17	—	38	42	$+\infty$
$-\infty$	12	17	20	38	42	$+\infty$

---

erase(38)

$\boxed{-\infty}_1$	—	—	—	—	—	$+\infty$
$\boxed{-\infty}_{2\rightarrow}$	—	$\boxed{17}_{3\downarrow}$	—	—	—	$+\infty$
$-\infty$	—	$\boxed{17}_4$	—	—	42	$+\infty$
$-\infty$	—	$\boxed{17}_5$	—	—	42	$+\infty$
$-\infty$	12	$\boxed{17}_{6\rightarrow}$	—	$\boxed{\cancel{38}}_{7\downarrow}$	42	$+\infty$
$-\infty$	12	17	20	$\boxed{\cancel{38}}_8$	42	$+\infty$

$-\infty$	—	—	—	—	$+\infty$
$-\infty$	—	17	—	—	$+\infty$
$-\infty$	—	17	—	42	$+\infty$
$-\infty$	—	17	—	42	$+\infty$
$-\infty$	12	17	—	42	$+\infty$
$-\infty$	12	17	20	42	$+\infty$

---

insert(48, x)

Coin flips: heads, tails: height = 1

$\boxed{-\infty}_1$	—	—	—	—	$+\infty$
$\boxed{-\infty}_{2\rightarrow}$	—	$\boxed{17}_{3\downarrow}$	—	—	$+\infty$
$-\infty$	—	$\boxed{17}\ _{4\rightarrow}$	—	$\boxed{42}_{5\downarrow}$	$+\infty$
$-\infty$	—	17	—	$\boxed{42}_6$	$+\infty$
$-\infty$	12	17	—	$\boxed{42}_7$	$+\infty$
$-\infty$	12	17	20	$\boxed{42}_{8\rightarrow}$	$+\infty$

$-\infty$	—	—	—	—	—	$+\infty$
$-\infty$	—	17	—	—	—	$+\infty$
$-\infty$	—	17	—	42	—	$+\infty$
$-\infty$	—	17	—	42	—	$+\infty$
$-\infty$	12	17	—	42	48	$+\infty$
$-\infty$	12	17	20	42	48	$+\infty$

---

insert(24, y)

Coin flips: tails: height = 0

$\boxed{-\infty}_{1\downarrow}$	—	—	—	—	—	$+\infty$
$\boxed{-\infty}_{2\rightarrow}$	—	$\boxed{17}_{3\downarrow}$	—	—	—	$+\infty$
$-\infty$	—	$\boxed{17}_{4}$	—	42	—	$+\infty$
$-\infty$	—	$\boxed{17}_{5}$	—	42	—	$+\infty$
$-\infty$	12	$\boxed{17}_{6}$	—	42	48	$+\infty$
$-\infty$	12	$\boxed{17}_{7\rightarrow}$	$\boxed{20}_{8\rightarrow}$	42	48	$+\infty$

$-\infty$	—	—	—	—	—	—	$+\infty$
$-\infty$	—	17	—	—	—	—	$+\infty$
$-\infty$	—	17	—	—	42	—	$+\infty$
$-\infty$	—	17	—	—	42	—	$+\infty$
$-\infty$	12	17	—	—	42	48	$+\infty$
$-\infty$	12	17	20	24	42	48	$+\infty$

---

erase(42)

$\boxed{-\infty}_{1\downarrow}$	—	—	—	—	—	—	$+\infty$
$\boxed{-\infty}_{2\rightarrow}$	—	$\boxed{17}_{3\downarrow}$	—	—	—	—	$+\infty$
$-\infty$	—	$\boxed{17}_{4\rightarrow}$	—	—	$\boxed{\cancel{42}}_{5\downarrow}$	—	$+\infty$
$-\infty$	—	17	—	—	$\boxed{\cancel{42}}_{6}$	—	$+\infty$
$-\infty$	12	17	—	—	$\boxed{\cancel{42}}_{7}$	48	$+\infty$
$-\infty$	12	17	20	24	$\boxed{\cancel{42}}_{8}$	48	$+\infty$

$-\infty$	—	—	—	—	—	$+\infty$
$-\infty$	—	17	—	—	—	$+\infty$
$-\infty$	—	17	—	—	—	$+\infty$
$-\infty$	—	17	—	—	—	$+\infty$
$-\infty$	12	17	—	—	48	$+\infty$
$-\infty$	12	17	20	24	48	$+\infty$

3. (10 points) Draw the 17-entry hash table that results from using the has function: $h(k) = ((3k + 5) \bmod 11)$, to hash the keys: 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5, assuming collisions are handled by double hashing using the secondary hash function: $h_s(k) = (7 - (k \bmod 7))$.
If there is still a collision after double hashing, I will use $h(i, k) = (h(k) + i \cdot h_s(k)) \bmod 17$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
								12								

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
					44			12								

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13					44			12								

$$h(88) = ((3(88) + 5) \bmod 11) = 269 \bmod 11 = 5$$
$$h_s(88) = (7 - (88 \bmod 7)) = 7 - 4 = 3$$
$$h(1, 88) = (5 + (1)3) \bmod 17 = 8$$
$$h(2, 88) = (5 + (2)3) \bmod 17 = 11$$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13					44			12			88					

$$h(23) = ((3(23) + 5) \bmod 11) = 74 \bmod 11 = 8$$
$$h_s(23) = (7 - (23 \bmod 7)) = 7 - 2 = 5$$
$$h(1, 23) = (8 + (1)5) \bmod 17 = 13$$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13					44			12			88		23			

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	94				44			12			88		23			

$$h(11) = ((3(11) + 5) \bmod 11) = 38 \bmod 11 = 5$$
$$h_s(11) = (7 - (11 \bmod 7)) = 7 - 4 = 3$$
$$h(1, 11) = (5 + (1)3) \bmod 17 = 8$$
$$h(2, 11) = (5 + (2)3) \bmod 17 = 11$$
$$h(3, 11) = (5 + (3)3) \bmod 17 = 14$$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	94				44			12			88		23	11		

$$h(39) = ((3(39) + 5) \bmod 11) = 122 \bmod 11 = 1$$
$$h_s(39) = (7 - (39 \bmod 7)) = 7 - 4 = 3$$
$$h(1, 39) = (1 + (1)3) \bmod 17 = 4$$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	94			39	44			12			88		23	11		

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	94			39	44			12		20	88		23	11		

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	94			39	44			12	16	20	88		23	11		

$$h(5) = ((3(5) + 5) \bmod 11) = 20 \bmod 11 = 9$$
$$h_s(5) = (7 - (5 \bmod 7)) = 7 - 5 = 2$$
$$h(1, 5) = (9 + (1)2) \bmod 17 = 11$$
$$h(2, 5) = (9 + (2)2) \bmod 17 = 13$$
$$h(3, 5) = (9 + (3)2) \bmod 17 = 15$$

Final Result:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	94			39	44			12	16	20	88		23	11	5	

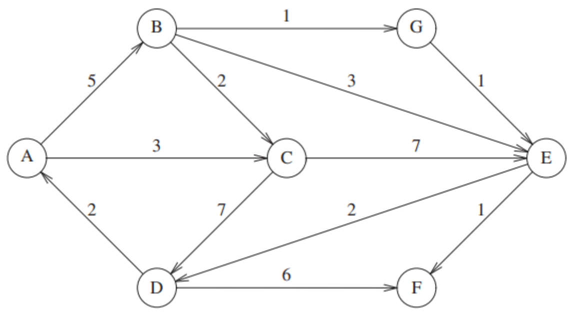
4. (10 points) An airport is developing a computer simulation of air-traffic control that handles events such as landings and takeoffs. Each event has a time-stamp that denotes the time when the event occurs. The simulation program needs to efficiently perform the following two fundamental operations:

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

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

   I would say that the best data structure for this task is a binary search tree. This is because it has an average Big-O of $O(log(n))$ for both inserting into a tree and retrieving data from a tree. In the worst case it will be $O(n)$. A tree will easily store the data nicely and organized.

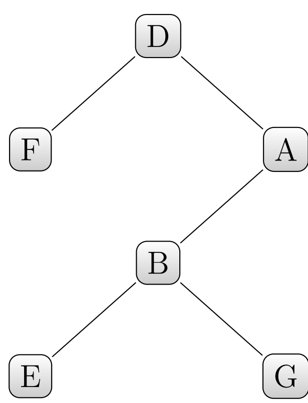
5. (15 points) Find the shortest path from D to all other vertices for the graph below.



(a) Illustrate the minimum priority queue at each iteration Dijkstra's algorithm.

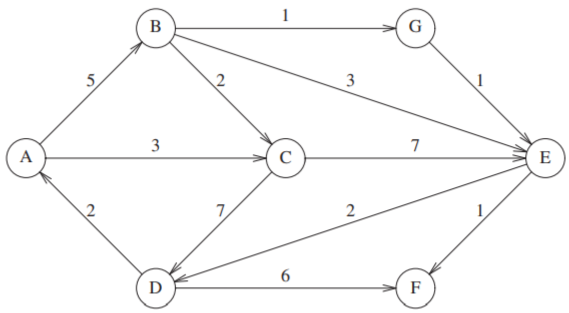
	D	A	B	C	E	F	G
1	$\boxed{0}$	∞	∞	∞	∞	∞	∞
2		$\boxed{D/2}$	∞	∞	∞	D/6	∞
3			∞	∞	∞	$\boxed{D/6}$	∞
4			$\boxed{A/7}$	A/5	∞		∞
5				$\boxed{A/5}$	∞		∞
6					B/10		$\boxed{B/8}$
7					$\boxed{B/10}$		

(b) Draw the Shortest Path Tree.



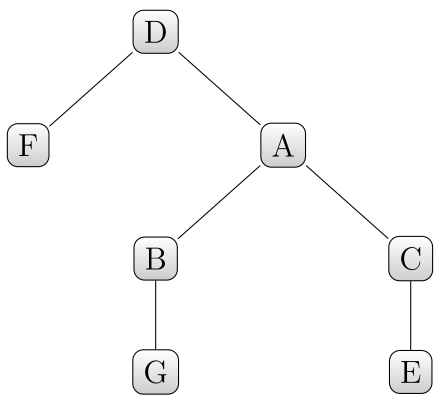
(c) What is the running time of the Dijkstra's algorithm under the assumption that the graph is implemented based on an adjacency list and the minimum priority queue is implemented based on a binary heap?
Under these assumptions, the running time would be $O(mlog(n))$.

6. (15 points) Find the shortest unweighted path from D to all other vertices for the graph below. You can measure the distance from D by number of edges.



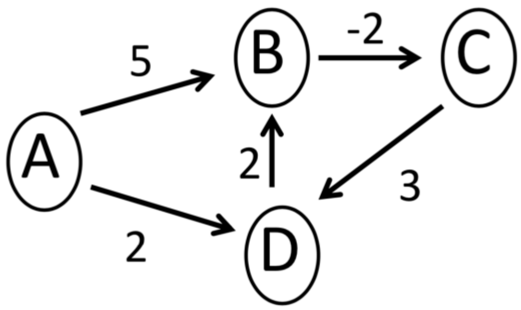
(a) Which graph algorithm can solve the problem?
Dijkstra's algorithm can solve this problem.

(b) Draw the Shortest Path Tree.



Here, B can also path to E with the same length cost as C, I just couldn't connect them.

7. (10 points) Apply the Dijkstra's algorithm to find the shortest path from the vertex A to all the vertices in the graph below. Does the algorithm return a correct output? Justify your answer using the Dijkstra's Theorem.



	A	B	C	D
1	$\boxed{0}$	∞	∞	∞
2		A/5	∞	$\boxed{\text{A/2}}$
3		$\boxed{\text{D/4}}$	∞	
4			$\boxed{\text{B/2}}$	

Normally, Dijkstra's algorithm does not work with graphs with negative cycles. For this particular graph, it seems that the algorithm does output to correct shortest path.
A $\rightarrow$ D $\rightarrow$ B $\rightarrow$ C

8. (20 points) There are eight small island in a lake, and the state wants to build seven bridges to connect them so that each island can be reached from any other one via one or more bridges. The cost of bridge construction is proportional to its length. The distance between pairs of islands are given in the following table.
(the subscripts represent the order in which I solved them in)

(a) Illustrate the Prim's algorithm using the graph below. Draw the Minimum Spanning Tree. What is the length of the bridges?

	1	2	3	4	5	6	7	8
1	-	240	210	340	280	200	345	$\boxed{120}_4$
2	-	-	265	175	215	$\boxed{180}_2$	185	$\boxed{155}_3$
3	-	-	-	260	$\boxed{115}_6$	350	435	195
4	-	-	-	-	$\boxed{160}_7$	330	295	230
5	-	-	-	-	-	360	400	$\boxed{170}_5$
6	-	-	-	-	-	-	$\boxed{175}_1$	205
7	-	-	-	-	-	-	-	305
8	-	-	-	-	-	-	-	-

I first began at vertex 7. I then added a new vertex at each comparison. This resulted in:
7 → 6 (175)
6 → 2 (180)
2 → 8 (155)
8 → 1 (120)
Here, I notices that the connections for 1 were not as low as the others, so I stayed at 8.
8 → 5 (170)
5 → 3 (115)
Here, I noticed that the connections for 3 were not as low as the last connections for 4, so I then connected 4 to 5.
5 → 4 (160)
This results in:
1 – 8 – 2 – 6 – 7
|
3 – 5 – 4

(b) Illustrate the Kruskal's algorithm using the graph below. Draw the Minimum Spanning Tree. What is the length of the bridges?

	1	2	3	4	5	6	7	8
1	-	240	210	340	280	200	345	$\boxed{120}_1$
2	-	-	265	175	215	$\boxed{180}_7$	185	$\boxed{155}_2$
3	-	-	-	260	$\boxed{115}_3$	350	435	195
4	-	-	-	-	$\boxed{160}_4$	330	295	230
5	-	-	-	-	-	360	400	$\boxed{170}_5$
6	-	-	-	-	-	-	$\boxed{175}_6$	205
7	-	-	-	-	-	-	-	305
8	-	-	-	-	-	-	-	-

I began by starting each vertex independently and connecting to the shortest island.
This resulted in:
$1 \rightarrow 8$ (120)
$2 \rightarrow 8$ (155)
$3 \rightarrow 5$ (115)
$4 \rightarrow 5$ (160)
$5 \rightarrow 8$ (170)
$6 \rightarrow 7$ (175)
Now we need to connect 6 and 7 to the rest of them. 6 seems to have the lowest connection with 2.
$6 \rightarrow 2$ (180)
This results in:

```
1 – 8 – 2 – 6 – 7
    |
3 – 5 – 4
```