CSCE 221 Cover Page
Homework #3
Due August 4 at 23:59 pm to eCampus

First Name McKenzie Last Name Burch UIN 225005240

User Name mburch13gig-em E-mail address mburch13gig-em@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.

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

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 McKenzie Burch Date July 31, 2018

Homework 3 (120 points)

due August 4 at 11:59 pm.

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

Reading assignment.

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

   - $h(k) = (3k + 5) mod 11$

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

$h(12) = (3 * 12 + 5) mod 11 = (36 + 5) mod 11 = 41 mod 11 = 8$
$h(44) = (3 * 44 + 5) mod 11 = (132 + 5) mod 11 = 137 mod 11 = 5$
$h(13) = (3 * 13 + 5) mod 11 = (39 + 5) mod 11 = 44 mod 11 = 0$
$h(88) = (3 * 88 + 5) mod 11 = (264 + 5) mod 11 = 269 mod 11 = 5$
$h(23) = (3 * 23 + 5) mod 11 = (69 + 5) mod 11 = 74 mod 11 = 8$
$h(94) = (3 * 94 + 5) mod 11 = (282 + 5) mod 11 = 287 mod 11 = 1$
$h(11) = (3 * 11 + 5) mod 11 = (33 + 5) mod 11 = 122 mod 11 = 5$
$h(39) = (3 * 39 + 5) mod 11 = (117 + 5) mod 11 = 122 mod 11 = 1$
$h(20) = (3 * 20 + 5) mod 11 = (60 + 5) mod 11 = 65 mod 11 = 10$
$h(16) = (3 * 16 + 5) mod 11 = (48 + 5) mod 11 = 53 mod 11 = 9$
$h(5) = (3 * 5 + 5) mod 11 = (15 + 5) mod 11 = 20 mod 11 = 9$

2. (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 $h_s(k) = 7 - (k \bmod 7)$?

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

$h(k) = (3k + 5) mod 11$	$h_s(k) = 7 - (k * mod 7)$	$h(k, i) = (h(k) + ih_s(k)) mod 11$
$h(12) = (3 * 12 + 5) mod 11 = 41 mod 11 = 8$		
$h(44) = (3 * 44 + 5) mod 11 = 137 mod 11 = 5$		
$h(13) = (3 * 13 + 5) mod 11 = 44 mod 11 = 0$		
$h(88) = (3 * 88 + 5) mod 11 = 269 mod 11 = 5$	$h(88) = 7 - (88 mod 7) = 3$	$h(88, 1) = (5 + 3) mod 11 = 8$
		$h(88, 2) = (5 + 6) mod 11 = 0$
		$h(88, 3) = (5 + 9) mod 11 = 3$
$h(23) = (3 * 23 + 5) mod 11 = 74 mod 11 = 8$	$h(23) = 7 - (23 mod 7) = 5$	$h(23, 1) = (8 + 5) mod 11 = 2$
$h(94) = (3 * 94 + 5) mod 11 = 287 mod 11 = 1$		
$h(11) = (3 * 11 + 5) mod 11 = 122 mod 11 = 5$	$h(11) = 7 - (11 mod 7) = 4$	$h(11, 1) = (5 + 4) mod 11 = 9$
$h(39) = (3 * 39 + 5) mod 11 = 122 mod 11 = 1$	$h(39) = 7 - (39 mod 7) = 3$	$h(39, 1) = (1 + 3) mod 11 = 4$
$h(20) = (3 * 20 + 5) mod 11 = 65 mod 11 = 10$		
$h(16) = (3 * 16 + 5) mod 11 = 53 mod 11 = 9$	$h(16) = 7 - (16 mod 7) = 5$	$h(16, 1) = (9 + 5) mod 11 = 3$
		$h(16, 2) = (9 + 10) mod 11 = 8$
		$h(16, 3) = (9 + 15) mod 11 = 2$
		$h(16, 4) = (9 + 20) mod 11 = 7$
$h(5) = (3 * 5 + 5) mod 11 = 20 mod 11 = 9$	$h(5) = 7 - (5 mod 7) = 2$	$h(5, 1) = (9 + 2) mod 11 = 0$
		$h(5, 2) = (9 + 4) mod 11 = 2$
		$h(5, 3) = (9 + 6) mod 11 = 4$
		$h(5, 4) = (9 + 8) mod 11 = 6$

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

3. (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.

- A minimum priority queue should be used because you can insert and remove the minimum element

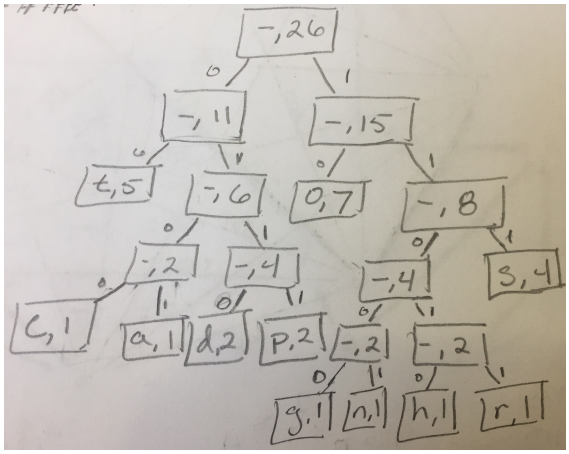
Priority Queue					
	Array		Linked List		Heap
	Sorted	Unsorted	Sorted	Unsorted	
insertElement()	$O(n)$	$O(1)$	$O(n)$	$O(1)$	$O(log_2 n)$
removeMin()	$O(1)$	$O(n)$	$O(1)$	$O(n)$	$O(log_2 n)$
minElement()	$O(1)$	$O(n)$	$O(1)$	$O(n)$	$O(1)$

4. (10 points) R-12.14 p. 588

Draw the frequency array. Use the minimum priority queue based on sorted array to build the Huffman tree for the string below. What is the code for each character and the compression ratio for this algorithm?

"dogs do not spot hot pots or cats".

Char	g	n	h	r	c	a	d	p	s	t	o
Freq	1	1	1	1	1	1	2	2	4	5	7
Code	11000	11001	11010	11011	0100	0101	0110	0111	111	00	10



— © Teresa Leyk

5. (10 points) R-13.15, p. 656

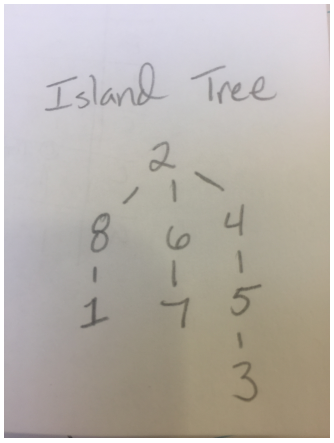
```
input: digraph
output: shortest path
set D[v] = 0 and D[u] = ∞
Q is priority queue of all vertices
S is queue of shortest path
S.insert(v);
while Q is not empty do:
    u = Q.removeMin();
    for each vertex adjacent to u in Q do:
        if D[z] > D[u] + w((u,z)){
          D[z] = D[u] + w((u,z));
          S.insert(z);
        }
return D[u];
```

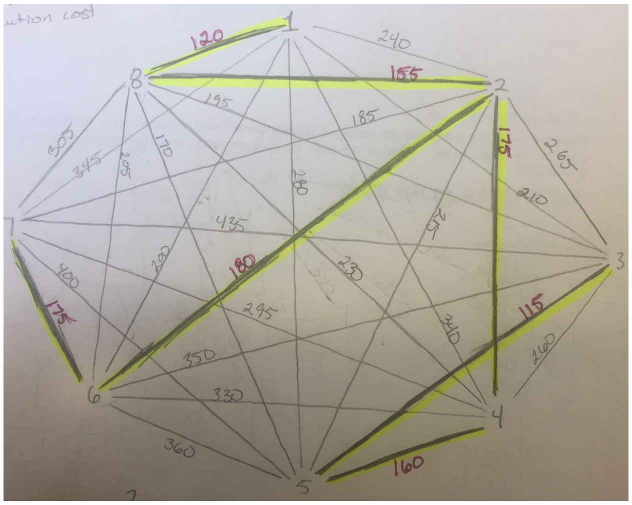
6. (10 points) R-13.16, p. 656

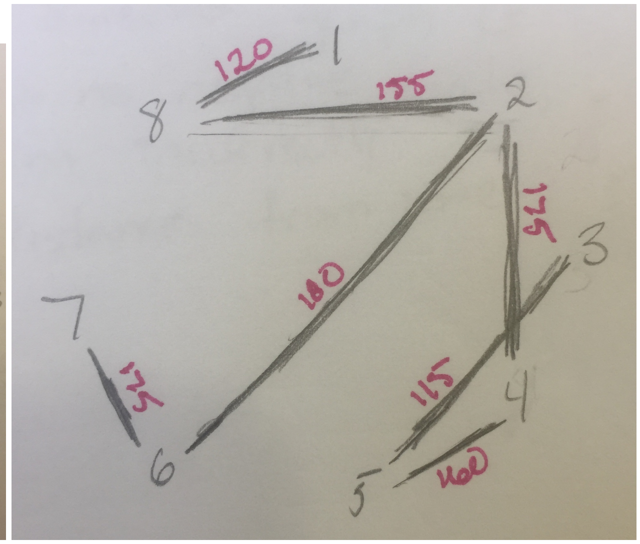
```
input: digraph
output: shortest path
set D[v] = 0 and D[u] = ∞
Q is priority queue of all vertices
S is queue of shortest path
S.insert(v);
while Q is not empty do:
    u = Q.removeMin();
    for each vertex adjacent to u in Q do:
        if D[z] > D[u] + w((u,z)){
          D[z] = D[u] + w((u,z));
          S.insert(z);
        }
print S;
return D[u];
```

7. (10 points) R-13.17, p. 656

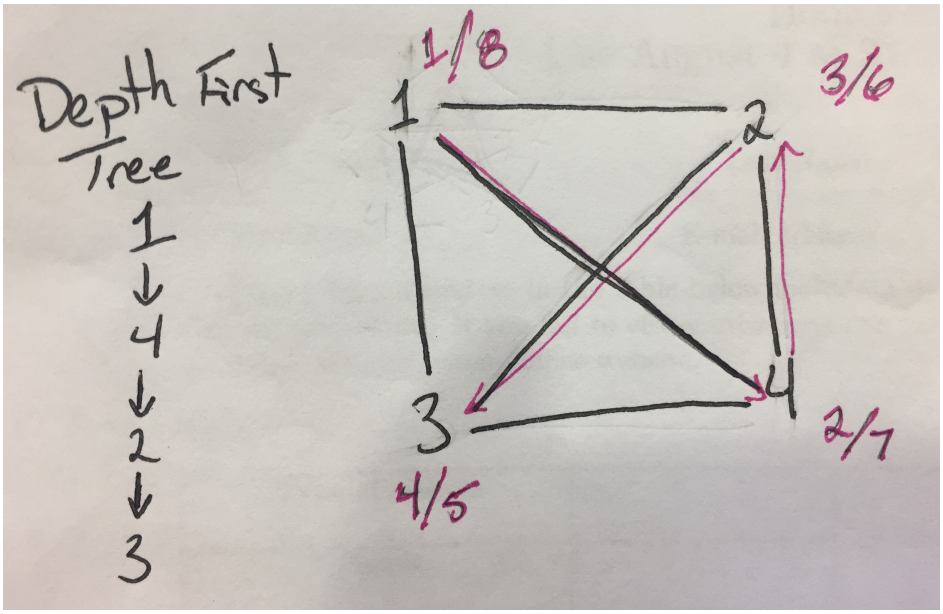
Island Distance								
	1	2	3	4	5	6	7	8
1	-	240	210	340	280	200	345	120
2	-	-	265	175	215	180	185	155
3	-	-	-	260	115	350	435	195
4	-	-	-	-	160	330	295	230
5	-	-	-	-	-	360	400	170
6	-	-	-	-	-	-	175	205
7	-	-	-	-	-	-	-	305
8	-	-	-	-	-	-	-	-




1
2
3
4
5
6
8
120
155
240
185
265
210
305
180
115
160
175
350
330
360


1
2
3
4
5
6
7
8
120
155
175
180
115
160
175

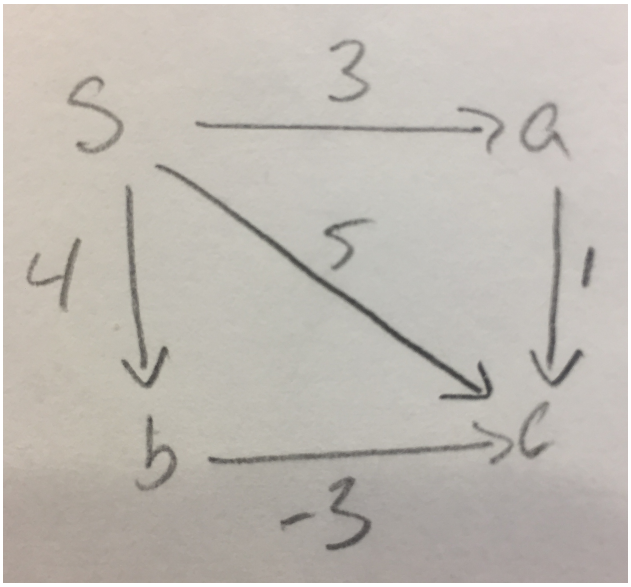
8. (10 points) R-13.31, p. 657



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

   Depth first search has a running time of $O(n+m)$ and visits each edge and vertex.

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



iter	P[v]	d[s]	d[a]	d[b]	d[c]
0	-	0	∞	∞	∞
1	s		3(s)	4(s)	5(s)
2	a	↓	↓	↓	4(a)
3	b				↓
4	c				

```
R(s,a,3)
if(d[a]>d[s]+w(s,a))
    d[a]=d[s]+w(s,a)
    P[a]=s
R(s,b,4)
if(d[b]>d[s]+w(s,b))
    d[b]=d[s]+w(s,b)
    P[b]=s
R(s,c,5)
if(d[c]>d[s]+w(s,c))
    d[c]=d[s]+w(s,c)
    P[c]=s
R(a,c,1)
if(d[c]>d[a]+w(a,c))
    d[c]=d[a]+w(a,c)
    P[c]=s
```