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

First Name Joseph Last Name Martinsen UIN 323009961

User Name E-mail address

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				
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.

Your Name Date

Homework 3 (100 points)

due April 23 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.

- Heap and Priority Queue, Chap. 8
- Graphs, Chap. 13

Problems.

1. (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.

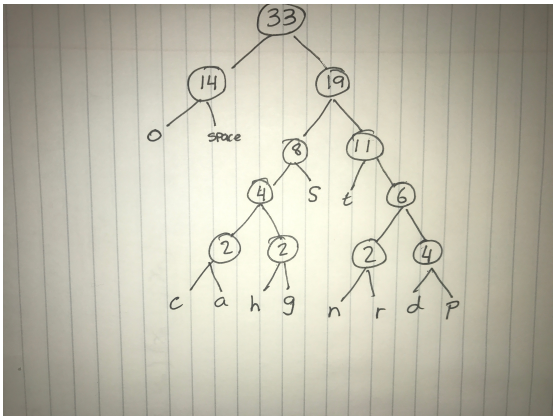
   Binary heap would the best choice in this case. Insert is O(n) but getting max and min would be O(1)

2. (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’’.

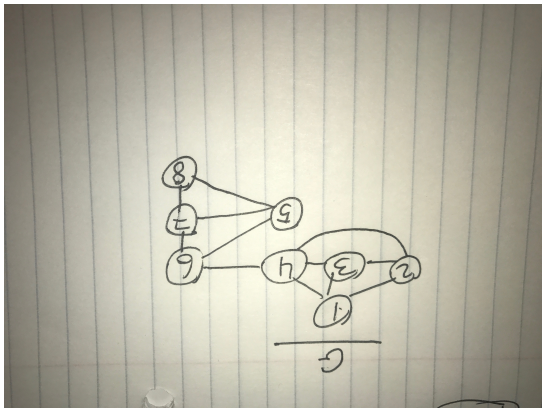
Char.	space	a	c	g	h	n	r	d	p	s	t
Freq.	7	1	1	1	1	1	2	2	4	5	7



3. (10 points) R-13.5, p. 654

   LA15,LA22,LA16,LA31,LA32,LA126,LA127,LA141, and LA169

4. (10 points) R-13.7, p. 655



   (a) DFS (using stack): 1,2,3,4,6,5,7,8

   (b) BFS (using que): 1,2,3,4,6,5,7,8

5. (10 points) R-13.8, p. 655

   (a) list needs: $10000 + 20000 = 3 * 10^4$ entries matrix needs : $10000^2 = 10^8$ entries, clearly list is the better option since space is an issue.

   (b) list needs: $10000 + 20000000 = 20010000$ entries matrix needs : $10000^2 = 100000000$, clearly list is the better option since space is an issue.

   (c) In a matrix the query can be in constant time and since space is not an issue, matrix is the clear choice.

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

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

   2 to 6 - 180
   5 to 3 - 115
   5 to 4 - 160
   6 to 7 - 175
   8 to 1 - 120
   8 to 2 - 155
   8 to 5 - 170

   1 to 8 to 2 to 6 to 7
   to
   3 to 5 to 4

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

   The depth-first search tree of a complete graph looks like a path.

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

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