CSC 212 Homework # 6
Heaps & Graphs
Due date: 08/01/2017 (5:00AM)

Guidelines:	This is an individual assignment. The homework must be **submitted electronically through LMS**. **Hard copy submissions are not accepted**.

Problem 1

1. Draw the max-heap (tree form) that results upon inserting the following keys: 4, 12, 16, 21, 45, 33, 10 and 60. The heap is initially empty.
2. Draw the previous heap after removing two elements.
3. Apply heap sort to sort the following array in increasing order:
   (a) If you want to sort the array in-place (without using an additional array) what type of heap will you use?
   (b) Draw the resulting array after converting it into a heap.
   (c) Draw the array after removing each element (the array must be sorted in-place, so show the complete content of the array at every step).

0	1	2	3	4	5	6
-	12	20	18	23	28	15

Problem 2

Consider the undirected graph shown in its adjacency matrix representation below.

-	a	b	c	d	e	f	g
a	0	1	1	1	0	0	0
b	1	0	0	0	1	0	0
c	1	0	0	0	0	1	0
d	1	0	0	0	1	1	0
e	0	1	0	1	0	1	1
f	0	0	1	1	1	0	1
g	0	0	0	0	1	1	0

1. Draw this graph.
2. Give its adjacency list representation.
3. Give its BFS and DFS traversals starting at node a (order the children in alphabetical order).

Problem 3

Use a min heap in the following.

1. Describe how to remove an element located at an arbitrary position i (not necessarily the root). What is the running time (big oh notation) of this procedure?
2. Describe how to update a key located at an arbitrary position i (the key is replaced by a new value). What is the running time (big oh notation) of this procedure?

Problem 4

Suppose you want to represent graphs where the number of nodes is unknown in advance, and the nodes IDs are arbitrary (not contiguous). You want the following operations to be performed efficiently:

- Add a node.
- Remove a node.
- Add an edge.
- Remove an edge.
- Find the degree of a node.
- Find all neighbours of a node.

Describe your data structure and give the performance of each of the above operations.