Module 9 Problems

1. [10 pts] Explain why O(log(E) = O(log(V)), but O(E) is not necessarily O(V).

2. [6 pts] The following courses: LA15, LA16, LA22, LA31, LA32, LA126, LA127, LA141, and LA169 have these prerequisites.
LA15: (none)
LA16: LA15
LA22: (none)
LA31: LA15
LA32: LA16, LA31
LA126: LA22, LA32
LA127: LA16
LA141: LA22, LA16
LA169: LA32

Find a viable course plan that meets the prerequisites.

3. [6 pts] Here is a graph given by the adjacency list:

vertex adjacent vertices

0: [1, 2, 3]
  1: [0, 2, 3]
  2: [0, 1, 3]
3: [0, 1, 2, 5]
  4: [5, 6, 7]
  5: [3, 4, 6]
  6: [4, 5, 7]
  7: [4, 6]

You always get the adjacent list for a node in the shown order.
Problem: Starting at node 0, Give the order that DFS visits the nodes

4. [6 pts] For the same graph as problem 3, starting at node 0, give the order that BFS visits the nodes.

5. [15 pts] Recall the unit on Priority Queues. Give Python code for the arbitrary update of priority for any element. Your algorithm should be able to update any element of a priority queue in O(log(N)), where N is the number of items in the priority queue

6. [10 pts] Modify Dijkstra's Algorithm as given in the notes to actully give a shortest path, not just the distance. Your Modified Dijkstra's Algorithm should be in Python.

7. [13 pts] Give a counterexample of what goes wrong when Dijkstra encounters negative edge weights
(even with no negative cycles)

8. [13 pts] Give an example to show that adding a positive offset to all negative edge weights will not succeed in computing shortest paths

9. [15 pts] In a graph G, a vertex is a universal sink if it has in-degree |V|-1, and out-degree 0. Give an algorithm that finds a universal sink if there is 1, or reports that there is no such node. Your algorithm, however, must work with the ADJACENCY MATRIX representation.

The complexity should be O(|V|).

10. [6 pts] Give a step-by-step execution trace of Kruskal's algorithm on the following graph:
A,C 7
A,D 5
B,C 8
B,E 4
C,D 9
C,E 10
D,E 15
D,F 6
E,F 12
G,F 11
G.E 13