EC 504
Spring, 2021
HW 4

Due Sunday, March 21, 8PM on Gradescope.

1. (15 pts) Consider an undirected graph $G = (V, E)$.

   (a) Describe a linear-time algorithm $O(|V| + |E|)$ for determining whether the graph is bipartite.

   (b) Describe an $O(|V|)$-time algorithm for determining whether the graph contains a cycle. .

   (c) Describe an algorithm to find the minimum number of edges that must be removed from a given undirected graph in order to make it acyclic. Note the graph may not be connected.

2. (15 pts) The two arrays below contain a description of a directed, graph, stored as a forward star. The vertices V are numbered $1, \ldots, 7$, and the edges are numbered $1, \ldots, 10$.

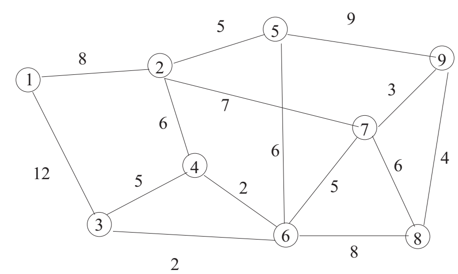
```
Array Firstout:  {1,3,5,7,8,10,11}
Array Endvertex:   {2,3,6,4,4,5,6,4,7,7}
```

   (a) Draw the graph.

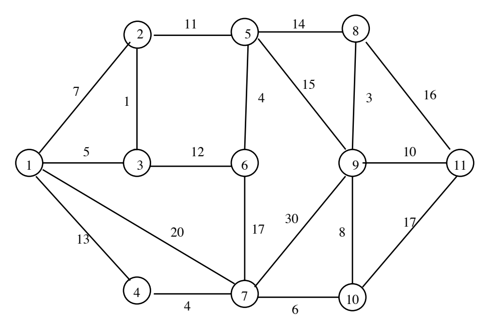
   (b) Is the graph acyclic?

   (c) Use depth first search, implemented by recursion, examining the arcs in the order in which they appear in the array Endvertex whenever there is a tie, to generate the order in which the vertices will be visited. Draw also the spanning tree constructed by the relation that the parent vertex spawns the recursive function which examines the child vertex.

3. (10 pts) Consider the undirected, weighted graph on the right. List the order in which the nodes will be visited in a Breadth-First Search, starting from node 5. Assume that the arcs out of a node will be visited in clockwise order, starting from straight up direction. Show the tree which is generated by the relation that node i's parent is the node which puts node i into the ~~stack~~. queue



4. (10 pts) For the graph on the right, compute the minimum spanning tree. Identify the algorithm you are using, and show some of the partial work. Compute the total weight of the minimum spanning tree.



5. (10 pts) Consider a path between two vertices s and t in an undirected, *connected* weighted graph G where no two edges have the same weight. The bottleneck length of this path is the maximum weight of any edge in the path. The bottleneck distance between s and t is the minimum bottleneck length of any path from s to t. Describe an algorithm to compute the bottleneck distance between every pair of vertices in an arbitrary undirected weighted graph.

6. (10 pts) Suppose you have computed a minimum spanning tree T for a graph $G(V, E)$. Assume you now add a new vertex n and undirected arcs $E_n = \big\{\{n, v_i\}$ for some $v_i \in V\big\}$, with new weights $w(n, v_i)$. Provide the pseudocode for an algorithm to find the minimum weight spanning tree in the augmented graph $G_a(V \cup \{n\}, E \cup E_n)$. Estimate the complexity of this algorithm.

7. (10 pts) Consider the undirected weighted graph in the right. Show each step of Boruvka's algorithm for finding a minimum spanning tree in the graph.