EL9343 Final Exam (Summer, 2020)

Name: **ID:**

August 21, 2020

- 2.5 hours exam, close-book, close-notes;
- Write all answers on your own answer sheets;
- Multiple choice questions may have multiple correct answers. You will get partial credits if you only select a subset of correct answers, and will get zero point if you select one or more wrong answers.

---

1. (**24 points, 3 points each**) **True or False**

(a) **T or F** Depth-first search of a graph is asymptotically slower than breadth-first search.

(b) **T or F** Dijkstra's algorithm is an example of a greedy algorithm, and the Floyd-Warshall algorithm solves the all-pairs shortest-paths problem using dynamic programming.

(c) **T or F** Let T be a minimum spanning tree of G. Then, for any pair of vertices s and t, the shortest path from s to t in G is the path from s to t in T.

(d) **T or F** Given a directed graph $G = (V, E)$, if you reverse all the edges in G, the resulting graph G^T (called the *reverse* or the *transpose* of G) has the same strongly connected components as the original graph G.

(e) **T or F** Dijkstra's algorithm and Prim's algorithm have the same asymptotical time complexity.

(f) **T or F** For the activity selection problem, the greedy algorithm that always chooses the activity with the shortest duration at each step is optimal.

(g) **T or F** To save time on determining whether an edge (u, v) exists in a graph, one should use adjacency matrix, instead of adjacency list, to represent the graph.

(h) **T or F** Fractional knapsack problem can be solved using greedy algorithm.

2. (**3 points**) Which of the following code can be huffman code for four symbols? (Select all that apply)
A: 00, 01, 10, 11
B: 111, 110, 10, 1
C: 0, 10, 110, 1110
D: 0, 11, 100, 101
E: None of the above

3. (**3 points**) Which of the following algorithms can be used to find shortest paths in a graph where some links have negative weights? (Select all that apply)
A: Floyd-Warshall; **B:** Dijkstra; **C:** Bellman Ford; **D:** BFS; **E:** Topological Sort.

4. (**10 points**) For the AVL Tree in Figure 1, two keys are inserted one after the other:

(a) After the first key with value of 30 is inserted, where is the violation, if any, of AVL tree property? how many rotations are needed to restore the AVL tree property? Plot the restored AVL tree.

(b) After the first key is inserted, and AVL tree is restored, a second key with value 11 is inserted, where is the violation, if any, of AVL tree property? how many rotations are needed to restore the AVL tree property? Plot the restored AVL tree.

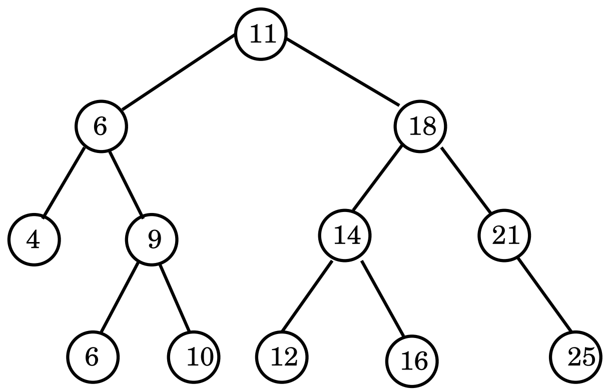

Figure 1: AVL Tree for Question 4

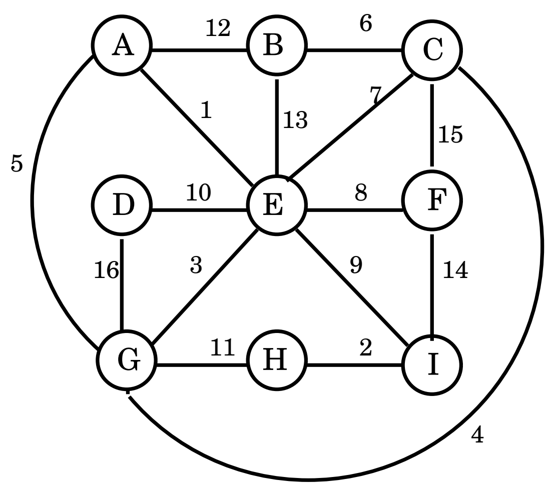

Figure 2: Weighted Undirected Graph for Question 5

5. (**10 points**) For the weighted undirected graph in Figure 2,

(a) (**5 points**) what are the FIRST FIVE (5) edges added to the minimum spanning tree by the Kruskal's algorithm, in the order they are added?

(b) (**5 points**) what are the FIRST FIVE (5) edges added to the minimum spanning tree by Prim's algorithm, started at vertex G, in the order they are added?

6. (**10 points**) Use Bellman Ford algorithm to find the shortest paths from A to all the other vertices in Figure 3, show your steps by plotting a graph similar to Figure 3 for each step on your answer sheet, put d values in vertices and use shaded edges to illustrate predecessors. The order for edge relaxation is AB, AC, BD, CS, DA, DB, DC, SA, SD

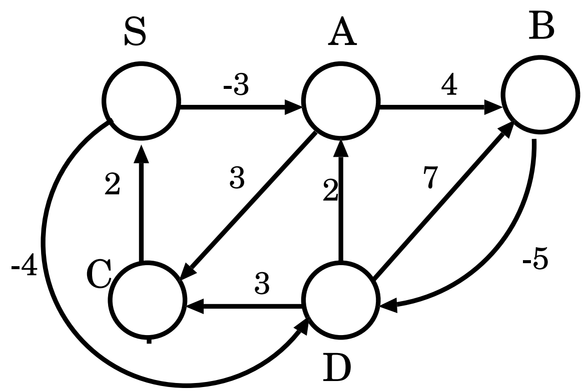

Figure 3: Bellman Ford Algorithm for Question 6

7. (**10 points**) In a directed graph, we define a vertex v to be a source, if there is no edge coming into v, and a sink if there is no edge going out of v. Please prove that in a directed acyclic graph (or DAG), there is at least one source and at least one sink.

8. (**12 points**) Design a dynamic programming approach to finding a longest weighted simple path from a vertex s to another vertex t in a weighted directed acyclic graph $G = (V, E)$. Write down the pseudo-code, and analyze the complexity of your algorithm.

9. (**18 points**) Transitive closure of a directed graph $G = (V, E)$ is a graph $G^* = (V, E^*)$, where E^*={(i,j): there is a path from vertex i to vertex j in G}. We can use Floyd-Warshall algorithm to compute the transitive closure of a graph in $\theta(|V|^3)$ time.

   (a) (**6 points**) Develop a more efficient algorithm to find transitive closure of an undirected graph in $\theta(|V| + |E|)$ time.

   (b) (**6 points**) Show how to use breadth-first-search to find transitive closure of a directed graph, and what is the time complexity?

   (c) (**6 points**) Show how to leverage on Kosaraju's algorithm to find transitive closure of a directed graph, and what is the time complexity?