ECE GY 9343 Final Exam (2021 Fall)

Name: **NetID:** **Version D**

Answer ALL questions. Exam is closed book. No electronic aids. However, you are permitted two cheat sheets, two sides each sheet. Any content on the cheat sheet is permitted.

Write your solutions on your own answer sheet, write your **Name, NetID and Exam Version** on the top of the first page.

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.

Requirements for remote exam

You should only have two electronic devices on your desktop. 1) A computer where you read the exam paper from. If you use a printer to print the exam paper, you should turn the computer off after you print the exam; 2) A smartphone or a tablet with a camera that you use to join the Zoom Session. Make sure both devices are plugged into a power source all the time.

Your cheat sheets must be on paper, not on your computer. You must answer on paper, not on a tablet/iPad. If you use a pencil to answer, please use a pencil of 2B or darker.

Close all other windows/tabs on your computer. You should only use it to read the exam questions. Avoid typing on the computer once you downloaded and saved the questions.

Put your Zoom device on your side. Turn on your camera and set it up so that I have a good view of your work area, your face, your hands and computer screen and keyboard. Make sure your device is charged. Change your Zoom name to your Net ID, then your full name. Mute your microphone but leave the **speaker on** so you can hear announcements. If the proctor found something unusual, we will announce it on the speaker for three times. If you fail to respond, your grade will be zero for this exam.

Always keep your video on. **Don't move your camera** unless the device fells. You might want to use a device holder, or at least use a pile of books to hold the device. If you have questions, please unmute yourself and ask the question by voice, or you can type your question into the chat.
If you need to use the restroom, send your proctor a chat message that you need to use restroom and then you can leave. Send your procotor another message when you come back from the restroom.

The exam ends at 11:30AM. Submission **deadline on Gradescope is 11:45AM**. You can use Apple Notes App or Adobe Scan to get a single PDF document. You can temporarily turn off the camera at the end of the exam to scan the exam papers. If you miss the deadline, we CANNOT take your exam papers unless you email us right away. The timestamp for email submission should no later than 11:50AM. We can take JPEG/PDF submissions via email. DON'T WAIT TILL 11:49AM and tell us you don't have enough time to scan!

1. (**20 points**) **True or False**

   (a) **T or F:** The minimum spanning tree problem can be solved by dynamic programming.

   (b) **T or F:** The dependency graph of the subproblems in dynamic programming problem must be a directed acyclic graph(DAG).

   (c) **T or F:** The transitive closure of a strongly connected graph is a fully connected graph.

   (d) **T or F:** For a directed acyclic graph(DAG), if we reverse all the edges, the resulting graph is still a (DAG).

   (e) **T or F:** Depth-First Search of a graph is asymptotically slower than Breadth-First Search.

   (f) **T or F:** There can be only one longest common subsequence for two given strings.

   (g) **T or F:** If adjacency list is used to represent a graph, the worst case time complexity to determine whether an edge (u, v) exists is $\theta(|V|)$.

   (h) **T or F:** Kruskal's algorithm for minimum spanning tree does not work on a graph with negative weight edges.

   (i) **T or F:** If all the edge weights are distinct, there can be only one shortest path between any pair of nodes;

   (j) **T or F:** For any Huffman code, if we exchange the codewords for the two least frequent symbols, the new code is still optimal.

2. (**4 points**) Which of the following statements about breadth-first search (BFS) are true?

   (a) BFS employs a First-In-First-Out queue to store the grey nodes;

   (b) Complexity of BFS on a dense graph is $\theta(V^2)$;

   (c) BFS search starting from a single vertex s always visits all vertices in a graph;

   (d) BFS can be used to solve single-source shortest path problem in an unweighted graph;

   (e) None of the above.

3. (**4 points**) Which of the following statements about AVL trees are true?

   (a) To find the successor in an AVL tree, we can use the same procedure as in binary search tree;

   (b) After a new node v is inserted and tree is rebalanced, the node v must be a leaf node;

   (c) In AVL tree insertion, once we perform a rotation at node v, we need to check if a rotation is necessary at any ancestor of v;

   (d) Any AVL tree has a depth of $\Theta(\log(n))$, where n is the number of nodes;

   (e) None of the above.

4. (**4 points**) Assuming all edge weights are distinct, which of the following statements about minimum spanning tree (MST) are true?

   (a) Prim's algorithm can start with any vertex;

   (b) For any given graph, the same MST will be returned by any MST algorithm;

   (c) There can be more than $|V| - 1$ edges in the MST;

   (d) The shortest path between a pair of vertices is always on the MST;

   (e) None of the above.

5. (**4 points**) Which of the following statements are true?

(a) Any P problem is also a NP problem;

(b) NP-Complete problems are not in NP-Hard problem set;

(c) All NP-complete problems can be verified in polynomial time;

(d) There is no known algorithm that can solve the 3-SAT problem in polynomial time;

(e) None of the above.

6. (**4 points**) For a graph with three vertices A, B and C, what are the possible set of start (S) and finish time (F) (DFS)?

(a) $A : S = 1, F = 6 \quad B : S = 3, F = 4 \quad C : S = 2, F = 5;$

(b) $A : S = 1, F = 6 \quad B : S = 4, F = 5 \quad C : S = 2, F = 3;$

(c) $A : S = 1, F = 6 \quad B : S = 3, F = 5 \quad C : S = 2, F = 4;$

(d) $A : S = 1, F = 2 \quad B : S = 3, F = 4 \quad C : S = 5, F = 6;$

(e) None of the above.

7. (**4 points**) Which of the following statements about shortest path are true?

(a) If path $P1$ is the shortest path between u and v, and $P2$ is the shortest path between v and w, then the shortest path between u and w is $P1 + P2$.

(b) Bellman-Ford algorithm is a greedy-based algorithm;

(c) In Floyd-Warshall, the sub-problems are limited by the number of edges allowed;

(d) In a directed acyclic graph, shortest path distance between any pair of vertices is always well-defined;

(e) None of the above

8. (**6 points**) For the AVL Tree in Figure 1, when a new key with value 13 is inserted, should we use single-rotation or double-rotation to restore the AVL tree property? what is the restored AVL tree with the inserted key?

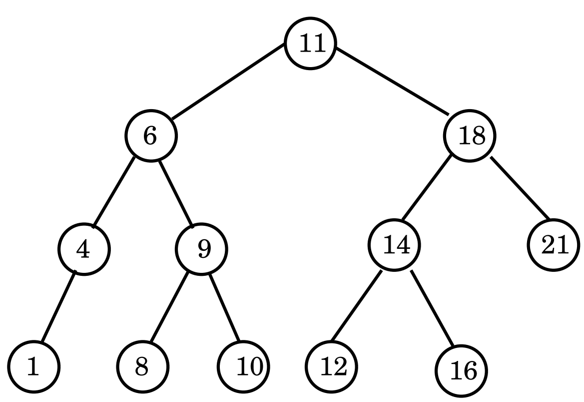

Figure 1: AVL Tree for Question 8

9. (**6 points**) Please use dynamic programming to find the Longest Common Subsequence between String "ZAYBDCAE" and "EBZYFDCA". Show the final $c[i,j]$ and $b[i,j]$ matrices.

10. (**6 points**) Construct the optimal Huffman code for the following set of symbols with associated frequencies: $\{a : 26, b : 12, c : 17, d : 28, e : 33, f : 15, g : 11, h : 34\}$

11. (**8 points**) In Homework 10, Q4: you are given an integer array `nums`, you are initially positioned at the array's index 0, and each element in the array represents your maximum jump length at that position. The following algorithm in the provided solution return *'true'* if you can reach the last index or *'false'* if not.

```
Boolean: CanJump(int[] nums)
 lastPos <-- nums.length-1;
 for (i = nums.length-1; i >=0; i--)
      if (i+nums[i]) >=  lastPos
         lastPos=i;
 return lastPos==0;
```

Prove the correctness of the algorithm using Loop-Invariant:

(a) (**2 points**) clearly state the *Loop Invariant* statement;

(b) (**1 points**) verify *Initialization* is true;

(c) (**4 points**) prove *Maintenance* can carry through between consecutive iterations;

(d) (**1 points**) verify *Termination* is true;

12. (**14 points**) Given a direct graph $G = (V, E)$, the strongly connected components form a component DAG, and the finish time $f(C)$ of a strongly connected component C is defined as the latest finish time of all vertices in C. If there is a path $C_1 \to C_2, \cdots, \to C_k$ in the component DAG, prove or disprove the following statements regarding DFS in G:

(a) (**4 points**) $f(C_1) > f(C_2) > \cdots f(C_k)$, no matter where the DFS starts.

(b) (**4 points**) Given $1 \leq i < j \leq k$, for any vertex $u \in C_i$, and any vertex $v \in C_j$, we always have $f(u) > f(v)$, no matter where the DFS starts.

(c) (**6 points**) If DFS starts from some vertex $u \in C_i$, for any vertex $v \in C_m$ $(m < i)$, and any vertex $w \in C_l$ $(l \geq i)$, we always have $f(v) > f(w)$.

Hint: if you think the statement is true, prove it is true; if you think the statement is wrong, you just need to come up with a counter-example to disprove it. You can use any theorem/statement we used in lectures.

13. (**16 points**) Given a directed acyclic graph $G = (V, E)$, each node $v \in V$ has a reward value of R_v, when an agent passes through v, it can collect the reward of R_v.

(a) (**12 points**) If an agent starts with a given node s, design a dynamic programming algorithm to find the optimal path $(s \to v_1 \to v_2 \to ... \to v_k)$ for the agent to collect the maximum total reward of $\sum_{i=1}^{k} R_{v_i}$ with complexity of $O(|E|)$. Write down the pseudo-code, prove the correctness of your algorithm, analyze the complexity of your algorithm. (*NOTE: k is NOT a given constant, intermediate nodes $\{v_i\}$ and destination node v_k are NOT given, the agent is allowed to take any path of any length to any destination node, your solution should be optimal among all possible paths to all possible destinations*).

(b) (**4 points**) In addition to (a), if the agent has to pay a cost of $C_{\langle v_i, v_{i+1}\rangle}$ when traversing each link $\langle v_i, v_{i+1}\rangle$, update your algorithm in (a) to find the optimal path that maximizes the net profit of the agent $\sum_{i=1}^{k} R_{v_i} - \sum_{i=0}^{k-1} C_{\langle v_i, v_{i+1}\rangle}$, where $s = v_0$, with complexity of $O(|E|)$. (No need to prove the correctness nor analyze the complexity for your updated algorithm.)

+++++++++++++ **End of Exam** +++++++++++++++