ECE GY 9343 Final Exam (2022 Fall)
Version E

Name: **NetID:**

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.

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 in-person exam

Please answer all questions on the question book. You should have enough space. Since we scan all the submissions in a batch, DON'T WRITE ANSWERS ON THE BACK of any page.

Make sure you write your NetID on the bottom of each page (abc1234). Don't tear off any page. If you need extra scrape papers, we can give to you. However, it will not be graded.

Requirements for remote exam

You should only have two electronic devices on your desktop. 1) A computer where you read the exam paper from; 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 write your solutions 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 identifies 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. **Dont 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. If you need to use the restroom, send us a chat message that you need to use restroom and then you can leave. Send us another message when you come back from the restroom.

Submission **deadline is 15 minutes** once exam ends. 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 15 minutes. We can take JPEG/PDF submissions via email. DONT WAIT TILL the last minute and tell us you dont have enough time to scan!

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

(a) **T or F:** The rod-cutting problem was solved using a greedy algorithm.

(b) **T or F:** Worst case tree height for AVL tree with n nodes is $\theta(\log(n))$.

(c) **T or F:** For an undirected graph, DFS only yields either tree-edges or back-edges.

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

(e) **T or F:** Based on the white-path theorem, if there is a link $\langle u, v \rangle$ in the graph, v will become a descendant of u in a DFS tree, no matter where DFS starts.

(f) **T or F:** DFS can be used to check if a directed graph has cycles.

(g) **T or F:** The Huffman code for a set of symbols with given frequencies is unique.

(h) **T or F:** For the activity selection problem, the activity with the shortest duration will always be selected.

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

(j) **T or F:** There is only one valid topological ordering of vertices in a directed acyclic graph.

2. (**4 points**) Which of the following statements are true?

(a) If a NP-Hard problem can be solved in polynomial time, then all the NP-Complete problems can be solved in polynomial time;

(b) NP-Complete problems definitely cannot be solved in polynomial time;

(c) 2-SAT problem is NP-Complete;

(d) The time needed to check whether a path is a Hamiltonian cycle is polynomial;

(e) If problem A is reducible to problem B in polynomial time, and if problem B can be solved in polynomial time, then $A \in P$;

3. (**4 points**) Which of the following statements about BFS and DFS are true?

(a) BFS and DFS have the same time complexity;

(b) BFS and DFS can both be used to find connected components in a undirected graph.

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

(d) If u has a shorter distance to the source s than v, then BFS starting from s will find first find u before finding v;

(e) We have to use different BFS algorithms for directed and undirected graphs.

4. (**4 points**) Which of the following sets of codewords can be Huffman codes?

(a) [0, 1];

(b) [00, 001, 01, 1];

(c) [1, 01, 000, 001];

(d) [0, 10, 110];

(e) [00, 01, 10, 11];

5. (**4 points**) Let C_1 and C_2 be two strongly connected components in a directed graph, if there is a link $\langle u, v \rangle$, where $u \in C_1$ and $v \in C_2$, which of the following are true?

(a) There is no path from any node in C_2 back to any node in C_1;

(b) No matter which node DFS starts from, the finish time of C_1 is always later than the finish time of C_2;

(c) In the second DFS pass of the Kosaraju's algorithm, the finish time of any node in C_1 should be always earlier than the discovery time of any node in C_2;

(d) No matter which node DFS starts from, the finish time of u is always later than the finish time of v;

(e) None of the above

6. (**4 points**) Which of the following statements about shortest path are true?

(a) If there is a negative weight cycle, the shortest path between some pair of nodes is not well-defined;

(b) Dijkstra algorithm is a greedy-based algorithm;

(c) In Floyd-Warshall, the sub-problems are to find shortest paths with limited number of edges;

(d) For a DAG, we can find the shortest paths from a single source node by taking one pass of Bellman-Ford relaxations on all edges in any order;

(e) If there is no negative weight cycle in a graph, the shortest path lengths between any three nodes always satisfy the triangle-inequality with arbitrary link weights setting.

7. (**6 points**) When building an AVL tree from scratch, keys are inserted in the order of:

$$3 \to 12 \to 21 \to 8 \to 5 \to 9,$$

plot the AVL tree after each key is inserted, and mark the type of rotation taken, if any, at each step.

8. (**8 points**) Use dynamic programming to find the Longest Common Subsequence between strings "FBDCEGHA" and "EFDCHGBA". Show the final $c[i, j]$ and $b[i, j]$ matrices.

9. (**8 points**) For the weighted undirected graph in Figure 1, the number on each link is its weight.

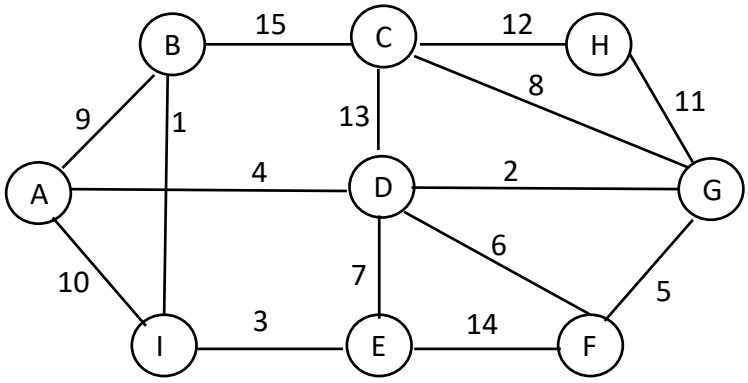

Figure 1: Weighted Undirected Graph for Question 9

(a) (**4 points**) what are the FIRST FIVE (5) edges added to the minimum spanning tree by the Prim's algorithm starting at vertex A, in the order that they are added?

(b) (**4 points**) what are the LAST FIVE (5) edges added to the minimum spanning tree by the Kruskal's algorithm, in the order they are added?

10. (**8 points**) For the directed graph in Figure 2, run DFS by considering vertices in alphabetical order, and assume the adjacent list of each vertex is sorted in alphabetical order too, show the discovery and finish time of each vertex, classify the type of each edge, tree/forward/backward/cross, plot the final DFS tree/forest.

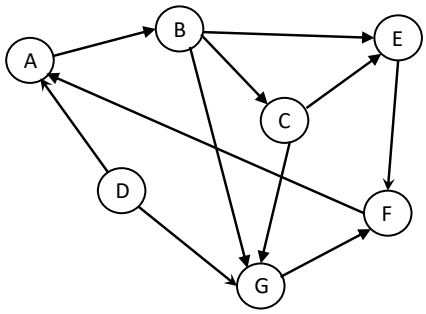

Figure 2: Digraph for Question 10

Answer:

	A	B	C	D	E	F	G
Discovery							
Finish							

Table 1: Discovery and Finish Time for Question 10

Tree Edges:

Back Edges:

Forward Edges:

Cross Edges:

Plot the DFS Tree/Forest

11. (**14 points**) There are n distinct positive numbers: $x_1, x_2, \cdots x_n$. Design a greedy algorithm with complexity $\Theta(n)$ to choose K different numbers out of $\{x_1, x_2, \cdots, x_n\}$ so that the weighted sum of the K numbers is maximized. In other words, let y_j be the j-th chosen number, we want to maximize $\sum_{j=1}^{K} w_j y_j$, where w_j is a given positive weight. (*Note:* $K = \Theta(1)$*, i.e.* K *is a constant, not increasing with* n.)

(a) (**4 points**) Write down the pseudo-code, analyze the complexity of your algorithm;

(b) (**5 points**) State and prove the *optimal substructure property*;

(c) (**5 points**) State and prove the *greedy choice property*.

12. (**16 points**) In a lossy communication network that can be modeled as a directed acyclic graph $G = (V, E)$, when a packet is transmitted on a link $e \in E$, there is a probability of p_e, $0 < p_e < 1$, that the packet will be lost. Suppose the packet losses on all the links are independent.

(a) (**8 points**) If one wants to send a packet from a source node $s \in V$ to a destination node $t \in V$, design an algorithm with complexity of $O(|V| + |E|)$ to find the most reliable path $(s \to v_1 \to v_2 \to ... \to t)$ so that the probability that the packet will be correctly received by t is maximized. Write down the pseudo-code, analyze the complexity of your algorithm. (*NOTE: you are allowed to take any path of any length from s to t, your solution should be optimal among all the possible paths*).

(b) (**8 points**) To deal with packet losses, one can implement multi-segment overlay routing with retransmissions: s first sends the packet to an overlay node o_1 (any node can be chosen as an overlay node, o_1 may not be directly connected to s, but is reachable from s), if the packet is lost on the way to o_1, s keeps retransmitting the packet until it is received by o_1; after o_1 receives the packet, it transmits it to the next overlay node o_2, retransmits the packet if it is lost; the process continues until the packet is delivered to t. Each packet transmission on each link incurs a cost of 1, and one has to pay a cost of C_v to use node v as an overlay node. Develop an algorithm to find the best overlay routing path $(s \to o_1 \to o_2 \to ... \to t)$ so that the expected total cost of transmitting a packet from s to t is minimized. Write down the pseudo-code, analyze the complexity of your algorithm. (*NOTE: your solution should be optimal among all the possible overlay routing strategies, including the one without using any overlay node; You can use any algorithm you learned in class without writing their detailed codes.*).

If you use up the space under any particular problem, you can write your answer here. On the page of the problem, tell us part of your work is here and write down the page number of this page and. Don't tear off any pages.

If you use up the space under any particular problem, you can write your answer here. On the page of the problem, tell us part of your work is here and write down the page number of this page and. Don't tear off any pages.

If you use up the space under any particular problem, you can write your answer here. On the page of the problem, tell us part of your work is here and write down the page number of this page and. Don't tear off any pages.