Name: ____

15-351 / 15-650 / 02-613 (Fall 2019): Midterm #2

Note: Please solve each of the following problems. This is a closed-notes and closed-book exam. You also should not use your laptops and cell phones. If you need additional space, use the back of the exam pages and indicate that you did so.

Problem 1. *(25 points) Short answ*

i. What is the worst possible insertion order in a binary search tree?

ii. What is the worst-case running time to insert a key into a skip list?

iii. After we insert a value

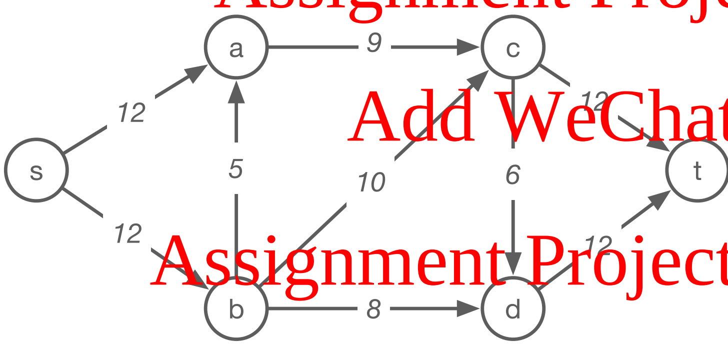
iv. Recall the dynamic programming recurrence for RNA folding. In what or OPT(i, j) subproblems? Why?

v. We can use network flow to solve maximum bipartite matching. Suppose there are $2n$ nodes and m edges in the bipartite graph G. What is the runtime to find maximum bipartite matching in G?

vi. Provide a short proof: Let f be an $s-t$ flow and (A, B) be an $s-t$ cut. Then $v(f) = f^{out}(A) - f^{in}(A)$, where $v(f)$ is the value of the flow being sent out from s.

Problem 2. *(25 points)* Use Ford-Fulkerson algorithm to solve the max-flow problem based on the following network where the capacity of each edge has been labeled.

- (15 pts) Draw, separately, the residual graph when you *cannot* find any augmenting path any more (i.e., when Ford-Fulkerson algorithm stops).
- (5 pts) Draw the max-flow on th
running Ford-Fulkerson. S
- (5 pts) Draw/Indicate the cut t



Problem 3. *(25 points)* A subsequence is a sequence that can be derived from another sequence by deleting characters without changing the relative ordering of the remaining characters. For example, "ABD" is a subsequence of "$ACBFDG$". The longest common subsequence (LCS) problem asks for the longest subsequence that is common to both input strings. For example, let $s_1 =$ "$ACBFDG$" and $s_2 =$ "$CAXBFWG$". The longest common subsequence of s_1 and s_2 is "$CBFG$".

Design a dynamic programming algorithm to find the LCS between two input strings. Briefly explain why your algorithm is correct and provide runtime analysis.

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Problem 4. *(25 points)* You are designing an exam for a class. You have a collection of problems $P = p_1, \ldots, p_n$. Each problem has an estimated time $t(p_i)$ in minutes that you think it will take a prepared student to answer. Each problem also has a quality score $q(p_i)$ that is your estimation of how good a problem it is (higher $q(p_i)$ means a better problem). Your class will have K minutes to take the exam.

i. (25 pts) Design a dynamic programming algorithm to select a subset of problems from P such that: (1) the total time to take the test is K, and (2) the sum of the qualities of the selected problems is as large as possible.

ii. (Extra credit - 15 pts) Now suppose t set of "concepts" $C = \{ {}_1, \ldots, {}_m \}$ ${}_1 =$ namic programming"), and each problem tests one concept $v(p_i) \in C$ (for example, problem p_1 might test concept $v(p_1) =$ "network flow"). Design a dynamic programming recurrence that selects a set of problems from P that (1) can be completed in $\leq K$ minutes, (2) tests every concept in C at least once (it can test a concept more than once), and (3) maximizes the sum of the qualities of the selected proble $|C|$ is a small constant.

For both questions, briefly describe why the algorithm is correct and provide run