CMPSCI 311: Introduction to Algorithms
Second Midterm Exam: Practice Exam

Name: ____ ID: __________________________

Instructions:

- Answer the questions directly on the exam pages.
- Show all your work for each question. Providing more detail including comments and explanations can help with assignment of partial credit.
- If the answer to a question is a number, *unless the problem says otherwise*, you may give your answer using arithmetic operations, such as addition, multiplication, "choose" notation and factorials (e.g., "$9 \times 35! + 2$" or "$0.5 \times 0.3/(0.2 \times 0.5 + 0.9 \times 0.1)$" is fine).
- If you need extra space, use the back of a page.
- No books, notes, calculators or other electronic devices are allowed. Any cheating will result in a grade of 0.
- If you have questions during the exam, raise your hand.

Question	Value	Points Earned
1	10	
2	10	
3	20	
5	10	
Total	50	

Question 1. (*10 points*) Indicate whether each of the following statements is TRUE or FALSE. No justification required.

1.1 (*2 points*): *Consider the following subset sum instance. The items have weights* $w_1 = 2, w_2 = 5, w_3 = 3, w_4 = 7, w_5 = 10$ *and the target weight is* $W = 14$. *This instance has a subset whose weight is exactly 14.*

1.2 (*2 points*): *In a flow network* $G = (V, E)$ *with capacities* c*, if we remove the edge with smallest capacity we always decrease the maximum flow.*

1.3 (*2 points*): *If a flow network has a* $s \to t$ *flow of capacity* M *then it also has a* $t \to s$ *flow of capacity* M*.*

1.4 (*2 points*): *Suppose we compute two optimal sequence alignments between two strings* x, y *using cost matrix* C*, but one alignment uses gap penalty* δ_1 *and the other uses* δ_2*. Let* M_1 *be the pairs of matched characters in the first and* M_2 *be the pairs of matched characters in the second.* ***True or False:*** *If* $\delta_1 \leq \delta_2$ *then* $M_1 \subset M_2$*.*

1.5 (*2 points*): *For the subset sum, the greedy algorithm that takes the element with largest weight subject to not exceeding capacity finds the optimal subset.*

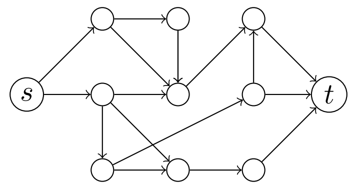
Question 2. (*10 points*) In this problem we'll revisit the strategy game Russell and Jesse played on the homework. Recall that they found a n bills of various denominations and arranged them into a list $S[1 \dots n]$ where $S[i]$ denotes the value of the ith bill. On each player's turn, they may take a single bill from either the front or the end of the list and then it becomes the next player's turn. The goal is to collect as much money as possible.

2.1 (*2 points*): *Suppose the list of bills is* $S = [1, 2, 10, 5, 17, 8, 6, 4]$*. If Russell goes first and both players play optimally, how much money can Russell collect?*

2.2 (*3 points*): *They decide to change the rules so that on a player's turn, they may take any bill. Describe briefly what the optimal strategy is with this twist.*

2.3 (*5 points*): *Suppose that the bills come in both positive and negative denominations. Along with this twist, they change the rules so that on a player's turn they may take any prefix or suffix of the current list. Describe how you would compute the optimal strategy in this new game?*

Question 3. (*20 points*) In this problem we will be interested in computing the number $s-t$ paths in a directed graph.



3.1 (*1 points*): *In the above graph, what is the length of the shortest path between s and t?*

3.2 (*1 points*): *How many distinct shortest paths between s and t are there?*

3.3 (*8 points*): *Describe an algorithm for computing the number of shortest $s-t$ paths? You may assume all edges have weight* 1*.*

A set of $s-t$ paths are edge disjoint if each edge is used in at most one path.

3.4 (*2 points*): *How many edge-disjoint $s-t$ paths are there in the graph on the previous page?*

3.5 (*3 points*): *More generally, if there are k edge-disjoint paths from x to y in a graph, and there are k edge-disjoint paths from y to z in a graph, are there k edge-disjoint paths from $x-z$? Remember that the graph may have cycles.*

3.6 (*5 points*): *Describe an algorithm for computing the number of edge disjoint $s-t$ paths in a graph.*

Question 4. (*10 points*) Suppose we have a complete k-ary tree with n leaves (suppose $n = k^d$ for some integer d). Each leaf v is associated with a weight $w(v)$. The weight of an internal node is defined to be the sum of the weights of all leaves that are descendants of this node. So the weight of the root r is $w(r) = \sum_{\text{leaves } v} w(v)$. Design and analyze an algorithm to compute the weight of every internal node.