CS 2501 Exam 1

Name _____

You MUST write your e-mail ID on **EACH** page and put your name on the top of this page, too.

If you are still writing when "pens down" is called, your exam will be ripped up and not graded – sorry to have to be strict on this!

There are 5 pages to this exam. Once the exam starts, please make sure you have all the pages. Questions are worth different amounts of points.

Answers for the short-answer questions should not exceed about 20 words; if your answer is too long (say, more than 30 words), you will get a zero for that question!

This exam is CLOSED text book, closed-notes, closed-calculator, closed-cell phone, closed-computer, closed-neighbor, etc. Questions are worth different amounts, so be sure to look over all the questions and plan your time accordingly. Please sign the honor pledge below.

_____

_____

_____

_____

_____

In theory, there is no difference between theory and practice.
But, in practice, there is.

Page 2: Miscellaneous

1. [20 points] For each of the following statements, circle whether it is true (always true) or false.

- **True** or **False**: Depth-first search is typically less memory intensive than breadth-first search.
- **True** or **False**: On a sparse graph, an adjacency matrix uses less memory than an adjacency list.
- **True** or **False**: It takes E time to find the incoming edges to a given node when using an adjacency matrix.
- **True** or **False**: Some graphs have more than one valid topological sort.
- **True** or **False**: Topological sort has a worse time-complexity than BFS and DFS.
- **True** or **False**: Dijkstra's algorithm works when there are negative cost edges in a graph, but not if there are negative cost cycles.
- **True** or **False**: Given a flow network, the net flow across a given cut is always less than or equal to f, the current amount of flow moving through the network.
- **True** or **False**: The amount of flow going through a flow network can never exceed the capacity of any cut C.
- **True** or **False**: If there exists a cut in a flow network whose capacity is equal to the max-flow f of the network, then it is still possible that an augmenting path exists, but the path will be unable to be used to increase the overall flow.
- **True** or **False**: The Ford-Fulkerson algorithm runs in $\Theta(E * f)$ time.
- **True** or **False**: If provided with a fast reduction from problem A, to problem B, then one can conclude that the problem A is equally hard or harder than B.
- **True** or **False**: A reduction is not valid if the time it takes to actually reduce the problem is slower than the algorithm used on the target problem because then the reduction time dominates the algorithm.
- **True** or **False**: Max-Flow can be reduced to bi-partite matching.
- **True** or **False**: Prim's algorithm and Kruskal's algorithm have the same runtime, so it doesn't matter which you choose (in terms of runtime).
- **True** or **False**: All the operations on a find-union data structure are contant time or amortized constant time.
- **True** or **False**: Greedy algorithms require optimal substructure.
- **True** or **False**: When solving the continuous knapsack problem, we use profit-to-weight ratio and add items in decreasing order, but have to be careful about items with the same ratio and always take the lightest item first.
- **True** or **False**: Interval scheduling problems always have a unique solutin, and our greedy algorithm from class will always find it.
- **True** or **False**: Dijkstra's algorithm is a greedy algorithm.
- **True** or **False**: When making change with standard coins (quarter, dime, nickel, penny), the optimal solution is always unique.

Page 3: Graphs

2. [8 points] In the movie *Die Hard 2*, Bruce Willis and Samuel L. Jackson open up a briefcase with a bomb. To disarm it, they must take one 3 Liter and one 5 Liter water jug and somehow measure 4 Liters (they have access to a running fountain of water). Once they have done this, they can place those 4 Liters on a scale connected to the bomb to disarm it. The solution is to fill the 5L jug, pour all of it into the 3L jug (leaving 2L in the bigger jug). Empty the 3L jug. Put the 2L in the 3L jug. Refill the 5L jug to the rim and pour as much of it into the 3L leaving 4L in the 5L jug. For this question, solve a more general form of this problem: Given a list of jugs and their capacities $J = \{j_1, j_2, ..., j_n\}$ and a target amount T, find a solution to using those jugs to measure the target amount T. Describe your algorithm in a couple of sentences. You should use *Graphs* to solve this problem.

3. [2 points] How many nodes does your graph that your solution uses have? Make sure to define any variables you use as precisely as you can.

Page 4: Find-Union and MST

4. [10 points] Prove that the minimum-spanning tree problem has optimal substructure. More formally, suppose T is an MST for graph G. Now suppose we remove an arbitrary edge e from T, producing two disjointed spanning trees T_1 and T_2. Prove that if T is an MST for G, then T_1 must be an MST for $G' \subset G$ (the subset of nodes connected by T_1 only). *Use a proof by contradiction!*

Page 5: Greedy Algorithms

5. [8 points] You have an nxn square matrix filled with signed integers. You want to change signs on elements so each row and column has a positive sum. However, you must change the sign of an entire row or column all at once. Describe a greedy algorithm that solves this problem.

6. [2 points] In a few sentences, explain why your algorithm above is guaranteed to work. This does not need to be a formal proof, just a intuitive explanation of why your algorithm will terminate with the correct solution.