CS 5150/6150: Practice Final

Date: December 5, 2016, **Duration:** ...

NAME:

UID :

Rules: You are allowed to reference any course material that you bring with you, but using a laptop is not allowed, except for looking up the lecture slides. Please write down the solutions in the space provided below the questions. Attaching a rough sheet with your name/UID is OK, but shouldn't be necessary.

Submitting this paper implies agreement with all the rules above.

Problem 1 (10 points)

A common way to *lay out* an n-vertex graph G on a line, is to first partition the graph into $(n/2)$-vertex subgraphs G_1, G_2, lay out G_1 recursively, and then lay out G_2 to its right. Suppose the partitioning algorithm takes time $O(n \log n)$.

(a) (4 points) Write a recurrence for the total run time of the algorithm.

(b) (6 points) Find a closed form bound for the running time.

Problem 2 (5 points)

Calvin claims that the subset-sum problem [Input: a collection of integers $a_1, a_2, \ldots, a_n$ and a target T; Output: YES, if there is a subset of the a_i that add up to T and NO otherwise.] is solvable in polynomial time. His reasoning is as follows: (a) we can give a dynamic programming which runs in $O(Tn)$ time; (b) this is polynomial in T, n, and thus the overall run time is polynomial.

Do you agree with (a)? With (b)? If you disagree with either, provide reasoning in order to convince Calvin.

Problem 3 (5 points)

Let G be any directed graph with edge capacities, and let s, t be a source and a sink respectively. Consider the max flow f from s to t, and let f_e denote the flow on an edge e.

Now, consider the 'flow graph', which is a directed graph on the same set of nodes as G, and there exists the edge e iff $f_e > 0$ (and it has weight equal to f_e).

Prove that there is always a max flow f for which the flow graph is 'acyclic' (i.e., has no directed cycles).

Problem 4 (15 points)

Recall the Linear Programming 'relaxation' for the Vertex Cover problem. To recap, the input is an undirected graph $G = (V, E)$, and the goal is to pick a subset S of the vertices, such that for every edge, at least one of the end points is in S. The objective is to minimize the size of S.

In the relaxation, we introduce variables x_u for every vertex u, have the constraint $0 \le x_u \le 1$ for all u, and impose the constraints that for all $\{u, v\} \in E$, $x_u + x_v \ge 1$. The objective is to minimize $\sum_{u \in V} x_u$.

(a) (3 points) Suppose the graph G is simply the triangle (3 vertices, 3 edges). Prove that there's a solution to the above linear program (LP) in which the objective value is $3/2$.

(b) (6 points) For any graph G, consider the following strategy: first find the optimum solution to the LP above, and then, for every u, add the vertex u to S with *probability* equal to x_u (independent of all the other vertices).

What is the expected size of the set S thus obtained? (in terms of the x_u)

(c) (6 points) Say we follow the strategy in part (b) above. For any edge $\{u, v\}$, what is the probability that *neither* of u, v is added to S? Prove that this probability is $\le 1/4$. (*Hint:* use the fact that x_u's satisfy the constraints in the LP.)