G22.3520: Honors Analysis of Algorithms

Problem Set 2+3

Due on Wed, Oct 21, after the class

Collaboration is allowed, but you must write your own solutions.

Problem 1

Suppose $G(V, E)$ is a connected graph and $h : E \mapsto \mathbf{R}$ is an assignment of costs to its edges. Let $g : E \mapsto \mathbf{R}$ be another cost assignment that satisfies:

$$\forall\, e, e' \in E, \quad h(e) \leq h(e') \Longleftrightarrow g(e) \leq g(e').$$

Prove that there exists a spanning tree of G that is a minimum cost spanning tree with respect to costs $h(\cdot)$ as well as a minimum cost spanning tree with respect to costs $g(\cdot)$.

Solve: [Kleinberg Tardos]: Chapter 4, Problem 26, on page 202.

Note: The greedy algorithm for minimum spanning tree (taught in class) works even when costs are allowed to be negative.

Problem 2

Let G be an n-vertex connected graph with costs on the edges. Assume that all the edge costs are distinct.

1. Prove that G has a unique minimum cost spanning tree.
2. Give a polynomial time algorithm to find a spanning tree whose cost is the second smallest.
3. Give a polynomial time algorithm to find a cycle in G such that the maximum cost of edges in the cycle is minimum amongst all possible cycles. Assume that the graph has at least one cycle.

Problem 3

You are given a set of n intervals on a line:

$$(a_1, b_1],\ (a_2, b_2], \ldots, (a_n, b_n].$$

Design a polynomial time greedy algorithm to select minimum number of intervals whose union is the same as the union of all intervals.

Problem 4

Solve: [Kleinberg Tardos]: Chapter 4, Problem 29, on page 203.

Hint: First prove that a non-increasing sequence $(d_1, d_2, \ldots, d_n)$ is a degree sequence of some n-vertex graph if and only if $(d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n)$ is a degree sequence of some $(n-1)$-vertex graph.

Problem 5 (Dynamic Programming)

Given a tree with (possibly negative) weights assigned to its vertices, give a polynomial time algorithm to find a subtree with maximum weight. Note that a *subtree* is a connected subgraph of a tree.

Problem 6

Let $G = (V, E)$ be a directed *acyclic* graph (i.e. it does not contain any directed cycle).

1. Prove that the graph must have a vertex t that has no outgoing edge.
2. Suppose $|V| = n$. A *topological ordering* of the acyclic graph is a labeling of its vertices by integers from 1 to n such that
   - Any two distinct vertices receive distinct labels.
   - Every (directed) edge goes from a vertex with a lower label to a vertex with a higher label.

   Give a polynomial time algorithm to find a topological ordering of the graph.
3. Fix a node t that has no outgoing edge. For every node $v \in V$, let $P(v)$ be the number of distinct paths from v to t. Define $P(v) = 0$ if no such path exists and define $P(t) = 1$ for convenience. Give a polynomial time algorithm to compute $P(v)$ for every node v.

Problem 7 (Dynamic Programming)

Solve: [Kleinberg Tardos]: Chapter 6, Problem 21, on page 330.

Problem 8 (Dynamic Programming)

Solve: [Kleinberg Tardos]: Chapter 6, Problem 22, on page 330.

Problem 9 (Dynamic Programming)

Solve: [Kleinberg Tardos]: Chapter 6, Problem 28, on page 334.

Problem 10 (Dynamic Programming)

An independent set I in a graph is called *maximal* if the graph does not contain an independent set I' such that $I \subseteq I'$ and $|I| < |I'|$.

Given a tree on n vertices, and an integer $0 \leq k \leq n$, give a polynomial time algorithm to determine whether the tree has a maximal independent set of size k. (*Hint: Design an algorithm that solves the problem for all possible values of k.*).

Problem 11

Solve: [CLRS] Problem 17-2, Page 426.

Note: In (a), SEARCH *need not run in logarithmic time. In (b), define an appropriate potential function so that insertion runs in amortized* $O(1)$ *time. In (c), one does not expect the implementation to be very efficient.*

Problem 12

Solve: [CLRS] Problem 17-3, Page 427.

Note: Assume that α *is strictly larger than* $\frac{1}{2}$ *(and strictly less than* 1*). Ignore deletions (as everything would be similar to insertions).*