University of Washington
Department of Computer Science and Engineering
CSE 417, Winter 2020

January 28, 2020

Homework 4, Due Wednesday, February 5, 9:29 am, 2020

Turn-in instructions: Electronic submission using the CSE 417 canvas site. Each numbered problem is to be turned in as a separate PDF.

Problem 1 (10 points):

Let S be a set of intervals, where $S = \{I_1, \ldots, I_n\}$ with $I_j = (s_j, f_j)$ and $s_j < f_j$. A set of points $P = \{p_1, \ldots, p_k\}$ is said to be a *cover* for S if every interval of S includes at least one point of P, or more formally: for every I_i in S, there is a p_j in P with $s_i \leq p_j \leq f_i$.

Describe an algorithm that finds a cover for S that is as small as possible. Argue that your algorithm finds a minimum size cover. You algorithm should be efficient. In this case $O(n \log n)$ is achievable but it is okay if your algorithm is $O(n^2)$. You may assume that the intervals are sorted in order of finishing time.

Problem 2 (10 points):

Let $G = (V, E)$ be a directed graph with lengths assigned to the edges. Let $\delta(u, v)$ denote the shortest path distance from u to v. Prove that for all vertices $u, v, w \in V$:

$$\delta(u, w) \leq \delta(u, v) + \delta(v, w).$$

You may assume that the graph is strongly connected, so that there is a path between every pair of vertices.

Problem 3 (10 points):

Let $G = (V, E)$ be a directed graph with integral edge costs in $\{1, 2\}$. Give an $O(n + m)$ time algorithm that given vertices $s, t \in V$ finds a shortest path from s to t.

Programming Problem 4 (10 points):

The degree of a vertex is the number of neighbors that the vertex has. Write a program to compute the distribution of vertex degrees for an undirected graph. For an input graph G determine the number of vertices of degree d for each d between 0 and n.

Test your program on random graphs using the generator from HW 3. Use values of n of 1000 (or larger). You should report results for values of p in the range 0.002 and 0.02.

Programming Problem 5 (10 points):

Implement the greedy algorithm for graph coloring discussed in class (Lecture 9). Run the algorithm on random graphs. Use values of n of 1000 (or larger). You should report results for values of p in the range 0.002 and 0.02. How many colors are needed on the average.

Programming Problem 6 Extra Credit (10 points):

Implement a graph coloring algorithm that performs better than the simple greedy coloring algorithm. Compare your results with the maximum degree of the graph, along with the simple greedy algorithm. You should report results for values of p in the range 0.002 and 0.02. How many colors are needed on the average.