University of Washington January 21, 2020
Department of Computer Science and Engineering
CSE 417, Winter 2020

Homework 3, Due Wednesday, January 29, 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):

Suppose that an n-vertex undirected graph $G = (V, E)$ has vertices s and t with the distance from s to t strictly greater than $n/2$. Show that there must exist some vertex v, not equal to either s or t, such that deleting v destroys all $s - t$ paths. (This could be phrased as: show that a graph with *diameter* strictly greater than $n/2$ has an *articulation point.*) Give an algorithm that finds v in $O(n + m)$ time, you can assume that you are given the vertices s and t that have separation of greater than $n/2$.

Problem 2 (10 points):

Consider a directed graph on n vertices, where each vertex has exactly one outgoing edge. This graph consists of a collection of cycles as well as additional vertices that have paths to the cycles, which we call the branches. Describe a linear time algorithm that identifies all of the cycles and computes the length of each cycle. You can assume that the input is given as an array A, where $A[i]$ is the neighbor of i, so that the graph has the edge $\langle i, A[i]\rangle$.

For clarity, make sure that you describe in English the main steps of your algorithm, and don't just provide code. Justify the correctness of your algorithm.

Problem 3 (10 points):

Let $P = \{x_1, \dots x_n\}$ be points on the X-axis in increasing order, and R be a non-negative integer. Give an $O(n)$ time algorithm to determine the minimum number of intervals of length R to cover the points. Explain why your algorithm is correct. (This problem relates to Chapter 4 material on greedy algorithms, but should be doable before the material has been presented in class.)

Programming Problem 4 (10 points):

The purpose of this problem is to construct a random graph generator for use in other programming problems and to demonstrate an implementation of graphs using adjacency lists.. A random graph generator, given an input parameter n, picks "at random" a graph with n vertices. There are multiple different models of random graphs. We will consider the *edge density* model, where a parameter p gives the probability of each edge being present. This model is referred to as $\mathcal{G}_p^n$. The undirected random graph generation problem is: given an integer n and a real number p with $0 \leq p \leq 1$, construct an undirected graph on n vertices where each edge $\{u, v\}$ has probability p of being in the set of edges E, and the probability of each edge is independent. Write a generator for $\mathcal{G}_p^n$

which given inputs n and p constructs a random undirected graph in adjacency list representation.

For this problem, write the graph generator and a routine to print the edges and vertices of a graph. Print the results a creating two different random graphs using $n = 10$ and $p = 0.2$.

Programming Problem 5 (10 points):

Implement an algorithm for the Connected Components Problem for undirected graphs. The algorithm should take as input an undirected graph, and identify the connected components. The vertices should be labeled by which component they belong to. If there are d connected components, you should name the components $1, 2, \ldots, d$. You will need to determine the size of each connected component.

Using the random graph generator from problem 4, experiment with the the connected component structure of random graphs. For a fixed n, look at what happens as you vary the value of p. You should use a moderately large value of n, at least $n = 1,000$ or even $n = 10,000$. (You should not choose an n so large that it takes a long time to generate the graphs, or you run into into overflow issues.) The two outputs that will be interesting to look at are the number of connected components and the size of the connected components. The "interesting" values of p will be small, for example, with $n = 10,000$ the range of interest for p is $0.0002 \leq p \leq 0.001$.