University of Washington
Department of Computer Science and Engineering
CSE 417, Autumn 2023

October 12, 2023

Homework 3, Due Friday, October 20, 11:59 PM, 2023

Turn in instructions: Electronics submission on GradeScope. Submit as a PDF, with each problem on a separate page. Algorithms should be described in pseudo-code with some of the steps in English. You can use algorithms described in class such as Breadth First Search and Depth First Search at a high level, e.g., "Use Breadth First Search to . . ." or "Compute a Depth First Search Tree" without giving the code for BFS or DFS.

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):

Give an algorithm to detect whether a given undirected graph contains a cycle. If the graph contains a cycle, it should output one of the cycles. The algorithm should run in $O(n+m)$ time. You should explain why your algorithm is correct and achieves $O(n + m)$ run time. Hint: Depth First Search trees are your friend. Lemma 3.7 from the text (page 85) is relevant.

Problem 3 (10 points):

Let $P = \{x_1, \ldots x_n\}$ 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 (7 points):

LeetCode problem 1221. Split into balanced strings.

For problems from LeetCode, write a program that solves the given problem in one of the languages supported by LeetCode. Run the program in LeetCode and pass the tests. (Note that you can add your own test cases, which can be very helpful in debugging.) You should submit your source code, as well as submitting a screen shot that shows the solution has been accepted. If there is an additional question (such as for this problem), give the answer to the question.

Question: Explain why your algorithm finds the optimal solution.

Programming Problem 5 (7 points):

LeetCode problem 455. Distributing cookies.

Question: Explain why your algorithm finds the the maximum number of cookies.

Programming Problem 6 (6 points):

LeetCode problem 11. Container with the most water.

There is a brute-force approach taking $O(n^2)$ time, which tests every possible start and end choice. However, this will probably fail on test cases with a large number of elements due to a time out.

Question: Explain why your algorithm finds the maximum area in $O(n)$ time.