University of Washington
Department of Computer Science and Engineering
CSE 417, Autumn 2024

October 9, 2024

Homework 2, Due Friday, October 11, 2023

Turnin instructions: Electronics submission on GradeScope. Submit as a PDF, with each problem on a separate page.

Problem 1 (10 points):

Order the following functions in increasing order by their growth rate:

1. n^3
2. $(\log n)^{\log n}$
3. $n^{\sqrt{\log n}}$
4. $2^{n/10}$

Explain how you determined the ordering.

Problem 2 (10 points):

We say that $T(n)$ is $O(f(n))$ if there exist c and n_0 such that for all $n > n_0$, $T(n) < cf(n)$. Use this definition for parts a and b.

a) Prove that $4n^2 + 3n\log n + 6n + 20\log^2 n + 11$ is $O(n^2)$. (You may use, without proof, the fact that $\log n < n$ for $n \geq 1$.)

b) Suppose that $f(n)$ is $O(r(n))$ and $g(n)$ is $O(s(n))$. Let $h(n) = f(n)g(n)$ and $t(n) = r(n)s(n)$. Prove that $h(n)$ is $O(t(n))$.

Problem 3 (10 points):

The *diameter* of an undirected graph is the maximum distance between any pair of vertices. If a graph is not connected, its diameter is infinite. Let G be an n node undirected graph, where n is even. Suppose that every vertex has degree at least $n/2$. Show that G has diameter at most 2.

Problem 4 (10 points):

Give an algorithm for efficiently computing the *number* of shortest paths in an undirected graph between a pair of vertices. Suppose that you have an undirected graph $G = (V, E)$ and a pair of vertices v and w. Your algorithm should compute the number of shortest $v - w$ paths in G. Since this graph is unweighted, the length of a path is defined to be the number of edges in the path. Your algorithm should have run time $O(n + m)$ for a graph of n vertices and m edges. If there is no path from v to w, your algorithm should report an error.

You should explain why your algorithm is correct and justify the run time of the algorithm.

Problem 5 (10 points):

Let $G = (V, E)$ be an undirected graph with n vertices such that the degree of every vertex of G is at most k. Describe an algorithm to color the edges of G with at most $2k - 1$ colors such that any pair of edges e and f which are incident to the same vertex have distinct colors. Explain why your algorithm successfully colors the edges of the graph.

You should describe your algorithm using pseudo-code, which allows you to use a mix of English language statements and control structures. For example, if you were asked to color a graph with maximum degree at most k with $k + 1$ colors you could you could give the following pseudo-code:

```
Set all vertices to uncolored
Foreach vertex v
    Select a color for v from [1,k+1] that is not used by any of v's neighbors
```

To show that the algorithm works, you would need to argue that there is always a color available for the select statement to choose.