University of Washington — January 13, 2023
Department of Computer Science and Engineering
CSE 417, Winter 2023

Homework 2, Due Friday, January 20, 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):

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 3 (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.

Programming Problem 4 (10 points):

The *Coupon Collector* problem is: There are n types of coupons. Each time you get a coupon, you are given a coupon of a random type (with equal probability of receiving each coupon). The question is how many coupons do you expect to receive, on the average, before you have collected the full set of coupons.

Your programming assignment is to write a simulator of the Coupon Collector Problem, and run simulations to see how long how many coupons are needed to complete the set. You should run your program for values of n up to $4,000$. Determine the average number of coupons required to complete the set.

Submit a table of values C the number of coupons collected, and C/n for $n = 200, 400, \ldots, 4,000$. Also, submit a PDF of your source code.

A reason for looking at this process is that it can be used to analyze the performance of the M-Proposal Algorithm for stable matching. The total number of coupons collected C will correspond with the M-rank, and C/n will correspond with the average m-rank.

Programming Problem 5 (10 points):

We now consider a variation of the Coupon Collector Problem where there are n types of coupons, and each coupon has a value associated with it. The value of a coupon is a random integer between 1 and n. You want to put together a complete set of coupons of *minimum value*, so that of all of the coupons you receive of a certain type c_i, you keep the one of minimum value. You collect coupons until you have a full set of coupons and then you determine the value of the set of coupons.

Your programming assignment is to write a simulator of the Coupon Collector problem with values, and run simulations to see what the average value is for a complete set of coupons. You should run this up to $n = 4,000$.

Submit a table of values V the value of coupons collected, and V/n for $n = 200, 400, \ldots, 4,000$. Also, submit a PDF of your source code.

A reason for looking at this process is that it can be used to analyze the performance of the M-Proposal Algorithm for stable matching. The total value of coupons collected V will correspond with the W-rank, and V/n will correspond with the average w-rank.