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

October 27, 2023

Homework 5, Due Friday November 3, 11:59 PM, 2023

This assignment is completely optional. Just for fun. No credit for doing it.

Turn in instructions: Submit a PDF on Gradescope with each problem on a separate page.

Programming Problem 1 (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 2 (10 points):

Implement the greedy algorithm for graph coloring discussed in class (Lecture 9, Slide 10, Coloring Algorithm version 2). 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? Since you are generating random graphs, taking several graphs with the same value of p will give more interesting results. Averaging over 10 graphs (per value of p) is probably sufficient.

Programming Problem 3 (10 points):

Create two new versions of your graph coloring algorithm from Problem 2 that process the vertices in order of increasing vertex degree, and in order of decreasing vertex degree. Compare your results with the maximum degree of the graph, along with the algorithm from Problem 2. You should report results for values of p in the range 0.002 and 0.02. How many colors are used on the average.

Programming Problem 4 (10 points):

Develop an algorithm that does better than "Highest Degree First", which appears to be the best of the algorithms above. For a particular challenge, I have been experimenting with $n = 1000$ and $p = 0.1$, the HDF algorithm seems to find solutions of 29 or 30 colors while I have an algorithm that seems to average about 27 colors.