CS41 Lab 12: randomness and approximation

This week, we'll continue exploring NP-COMPLETE decision problems, and develop approximation algorithms for related versions of those problems.

1. **One-Pass Auction.** Suppose you are planning to sell an old textbook via a *one-pass auction*, in which buyers make sequential bids and each bid must be immediately (and irrevocably) accepted or refused. Specifically, the auction works as follows:

   - n buyers arrive in a random order. You know n, but no other information about the buyers.
   - When buyer i arrives, they make a bid $b_i > 0$. This bid is fixed, and you have no advance knowledge of the distribution of bids.
   - You must immediately decide whether to accept the bid or not. If you accept the bid, you are done; all future buyers are turned away. If you reject b_i, buyer i departs and the bid is withdrawn; only then do you see bids from any future buyers.

   Your goal is to design a strategy you can follow to ensure a reasonable chance of accepting the highest of the n bids. Here, a *strategy* is a rule by which you decide whether to accept each presented bid, based only on the value of n and the bids you've seen so far.

   For example, one possible strategy would be to always accept the first bid presented. Following this strategy, you will accept the highest of the n bids with probability $\frac{1}{n}$, since you only succeed if the highest bid happens to be the first one presented.

   Give a strategy under which you accept the highest of the n bids with probability at least $\frac{1}{4}$, regardless of the value of n. (For simplicity, you may assume that n is an even number.) Prove that your strategy achieves this probabilistic guarantee.

2. **Three-Coloring Revisited.** Recall the THREE-COLORING problem: Given a graph $G = (V, E)$, output YES iff the vertices in G can be colored using only three colors such that the endpoints of any edge have different colors. The optimization version THREE-COLORING-OPT is the problem: Given a graph $G = (V, E)$ as input, color the vertices in G using at most three colors in a way that maximizes the number of *satisfied* edges, where an edge $e = (u, v)$ is satisfied if u and v have different colors.

   You have already showed that:

   - THREE-COLORING is NP-HARD
   - THREE-COLORING is NP-COMPLETE
   - There is a deterministic (i.e., non-randomized) approximation algorithm for THREE-COLORING-OPT which satisfies at least $2c*/3$ edges, where the optimum possible is c^* satisfied edges.

   Can we do any better with randomization? Give randomized algorithms for THREE-COLORING-OPT with the following behavior:

   (a) An algorithm that runs in worst-case (i.e., not expected) polynomial time and produces a three-coloring such that the expected number of satisfied edges is at least $2c^*/3$.

(b) An algorithm with expected polynomial runtime that always outputs a three-coloring that satisfies at least $2c^*/3$ edges.

(c) An algorithm that runs in worst-case polynomial time, and with probability at least 99% outputs a three-coloring which satisfies at least $2c^*/3$ edges. What is the running time of your algorithm? The following inequality might be helpful: $1 - x \leqslant e^{-x}$ for any $x > 0$.

3. **Chromatic Number.** Consider the optimization problem ChromaticNumber, defined as follows. Given a graph $G = (V, E)$ as input, determine the smallest number k such that it is possible to k-color the graph.

   (a) Prove that ChromaticNumber is NP-hard.

   (b) Prove that there is no efficient $\frac{4}{3}$-approximation to ChromaticNumber unless P = NP.

   (c) Prove that for any $\epsilon > 0$ there is no efficient $1+\epsilon$-approximation to ChromaticNumber unless P = NP. Hint: recall that $\forall k > 2$, k-coloring is NP-Complete.