CS6170: Randomized Algorithms
Problem Set #3

Name: Your name | Marks: 25

Roll No: Your roll number | Due: Oct 25, 23:59

Problem 1 — 5 marks

Consider the following deterministic paging algorithms.

- Longest-Time First: Whenever there is a cache-miss evict the item that has been in the cache for the longest period of time.
- Least Frequently Used: Whenever there is a cache-miss, evict the item that has been requested least often.

(a) (3 marks) Show that the LTF paging scheme is k-competitive where k is the size of the cache.

> **Solution:** Type your solution here.

(b) (2 marks) Show that the competitive ratio for LFU is unbounded.

> **Solution:** Type your solution here.

Problem 2 — 7 marks

An *edge coloring* of a graph is an assignment of colors to the edges of a graph such that no two edge that share a vertex are assigned the same color. Let us look at the online version of edge coloring where the number of vertices in the graph are fixed, and the edges arrive in an online fashion. We will assume that the degree of every vertex in the graph is at most Δ.

(a) (3 marks) Show that there exists an online algorithm that can edge color the graph with at most $2\Delta - 1$ colors.

The best that an online algorithm can be do is $\Delta + 1$, and follows from Vizing's theorem in graph theory.

> **Solution:** Type your solution here.

(b) (4 marks) Show that there is no deterministic algorithm that uses fewer than $2\Delta - 1$ colors in the worst case.

To that end, consider a graph consisting of disjoint stars with Δ vertices, and edges connecting the center vertex of each of the stars to another fixed vertex. Show that for any deterministic algorithm, there is an adversarial order that can force $2\Delta - 1$ colors.

> **Solution:** Type your solution here.

Problem 3 — 7 marks

Our goal in this problem is to show that there does not exist an online algorithm (randomized or deterministic) for the bipartite matching problem that gives a competitive ratio better than $1 - 1/e$. We will use Yao's minimax principle to achieve this.

To that end, let us construct a probability distribution over the input instances of the online bipartite matching problem. First, let us assume that the n vertices in R are known, and let π be a permutation of these vertices chosen uniformly at random. The n vertices in L arrive one-by-one and the i^{th} vertex in L has edges to the last $n-i+1$ vertices in R according to the permutation π.

(a) (1 mark) Show that every bipartite graph sampled via the process above has a perfect matching.

> **Solution:** Type your solution here.

(b) (3 marks) Let A be any deterministic online algorithm for bipartite matching. Prove that for every $i \in \{1, 2, \ldots, n\}$, the probability that A matches the i^{th} vertex of R to some vertex in L is at most

$$\min\left\{\sum_{j=1}^{n} \frac{1}{n-j+1}, 1\right\}.$$

> **Solution:** Type your solution here.

(c) (3 marks) Use the part above to conclude that for any deterministic online algorithm for bipartite matching, the expected size of the matching computed by it is at most $n(1-1/e)$.

> **Solution:** Type your solution here.

Problem 4 — 6 marks

A coloring of a graph is an assignment of colors to the vertices in the graph. A graph is said to be k-colorable, if there is a coloring of the graph with k colors such that no two adjacent vertices have the same color. Let G be a 3-colorable graph.

(a) (2 marks) Show that there exists a coloring of the graph with 2 colors such that no triangle is monochromatic. (A monochromatic triangle is one that has all vertices of the same color.)

Do not use Part (b) to solve this question.

> **Solution:** Type your solution here.

(b) (4 marks) Consider the following algorithm for coloring G with two colors so that no triangle is monochromatic. The algorithm begins with an arbitrary 2-coloring. While there are monochromatic triangles in G, the algorithm chooses one such triangle and changes the color of a randomly chosen vertex of the triangle. Derive an upper bound on the expected number of such recoloring steps before the algorithm finds a 2-coloring with the desired property.

Solution: Type your solution here.