

Submission instructions: As in previous homework.

## 25 (100 PTS.) Approximate coloring

Consider a graph  $G = (V, E)$ , with  $n$  vertices and  $m$  edge. Let  $k > 0$  be a parameter, and consider a  $k$ -coloring  $\psi : V \rightarrow \llbracket k \rrbracket$ , where  $\llbracket k \rrbracket = \{1, \dots, k\}$ . The set of *failed edges* for  $\psi$ , is

$$F(\psi) = \{uv \in E \mid \psi(u) = \psi(v)\}.$$

The size of  $F(\psi)$  is the *damage* of  $\psi$ . Of course computing the coloring with minimum damage is **NP-HARD** even for  $k = 3$ .

- 25.A.** (20 PTS.) Consider the deterministic algorithm that starts with an arbitrary  $k$ -coloring of  $G$ , but repeatedly changes the color a vertex if it strictly decreases the damage. The algorithm stops when no such improvement is possible (here, one has to perform this check for every vertex in the graph, and any possible color change). This is a *local-search* algorithm. Describe how to implement this algorithm so that it runs as quickly as possible. What is the running time of your algorithm. Furthermore, if  $\psi$  is the final coloring computed, prove that its damage is “small”. Formally, prove that  $|F(\psi)| \leq m/k$ .
- 25.B.** (30 PTS.) Provide a randomized algorithm, as fast as possible, that computes a random  $k$ -coloring  $\psi$  with  $\mathbb{E}[|F(\psi)|] \leq m/k$ . Prove the correctness of your algorithm, and bound its running time.
- 25.C.** (50 PTS.) Using the previous part, and only the previous part, provide a randomized algorithm, *as fast as possible* (in expectation), that outputs a valid coloring of  $G$  with  $3 \lceil \sqrt{m} \rceil$  colors. That is, all the edges are legally colored (i.e., the damage is zero).

## 26 (100 PTS.) Diamonds are forever.

A *mongrel* in the plane, is defined by its center point  $(x, y)$ , and is the union of three (closed) interior-disjoint segments. These segments are each of length one, and they are horizontal or vertical, and have  $(x, y)$  as one of their endpoints. A set of mongrels is *independent* if no two mongrels in it intersect. See [Figure 1](#).



Figure 1: Left: A set of mongrels (colored for clarity). For example, all the red mongrels form an independent set (depicted on the right).

**26.A.** (20 PTS.) Let  $S$  be a set of  $n$  mongrels in the plane that is contained inside an axis parallel square  $\square$  of sidelength  $\Delta > 0$  (assume  $\Delta$  is an integer), and let  $I \subseteq S$  be the largest independent set in  $S$ . Prove that  $|I| \leq 2\Delta^2$ . Present an algorithm that computes the largest independent set in  $S$  in  $n^{O(\Delta^2)}$  time.

**26.B.** (30 PTS.) Let  $M$  be any set of mongrels provided in advance. Let  $\varepsilon \in (0, 1)$  be a parameter, and consider  $\Delta = \lceil 10/\varepsilon \rceil$ . Let  $p = (\alpha, \beta)$  be a random point picked uniformly at random in the square  $[0, \Delta]^2$ , and consider the sets of lines

$$\mathcal{X} = \{x = \alpha + i\Delta \mid i \in \mathbb{Z}\} \quad \text{and} \quad \mathcal{Y} = \{y = \beta + i\Delta \mid i \in \mathbb{Z}\},$$

where  $\mathbb{Z}$  is the set of all integer numbers. This partitions the plane into a randomly translated (by  $p$ ) grid  $G$ , where every cell has sidelength  $\Delta$ . Let  $M'$  be the set of all the mongrels in  $M$  that do not intersect any of the lines of  $\mathcal{X}$  or  $\mathcal{Y}$  (i.e., these are all the mongrels of  $M$  that are fully contained inside the grid cells of  $G$ ). Prove that  $\mathbb{E}[|M'|] \geq (1 - \varepsilon)|M|$ .

**26.C.** (50 PTS.) You are given a set  $S$  of  $n$  mongrels in the plane, and a parameter  $\varepsilon \in (0, 1)$ . Using the previous two parts, present an algorithm, as fast as possible, that outputs (**in expectation**) an independent set of mongrels of size  $\geq (1 - \varepsilon)\text{opt}$ , where  $\text{opt}$  is the size of the largest independent set in  $S$ . For credit, your algorithm should have a polynomial running time if we consider  $\varepsilon$  to be a constant.

**27** (100 PTS.) Silly fail.

In the silly Traveling Salesman algorithm, the algorithm starts from a starting vertex  $v_1 = s$ , and in  $i$ th stage, it goes to the closest vertex to  $v_i$  that was not visited yet. The algorithm repeats this process till all vertices are exhausted, and then it returns to the start vertex.

Show an example, that proves that the silly traveling salesman does not provide any constant factor approximation to the TSP even with the *triangle inequality*.

Formally, for any constant  $c > 0$ , provide a complete graph  $G$ , and positive weights on its edges, such that the weights obey the triangle inequality, and the length of the silly TSP computed is by a factor of (at least)  $c$  longer than the length of the shortest TSP of  $G$ . (In particular, *prove* that the triangle inequality holds for the weights you assign to the edges of  $G$ .)