Homework 12- Due Thursday, December 8, 2022

Page limit  You can submit at most 2 pages per problem, even if the problem has multiple parts. If you submit a longer solution for some problem, only the first sheet of paper will be graded.

Reminder  Collaboration is permitted, but you must write the solutions by yourself without assistance, and be ready to explain them orally to the instructor if asked. You must also identify your collaborators and whether you gave help, received help, or worked something out together. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Exercises  Please practice on exercises in Chapter 7 of Mitzenmacher-Upfal.

Problems

1. (Algorithmic LLL) Show how, given integers $k$ and $n$, to find, in time $n^{O(k)}$ and with high probability, a coloring of the edges of the complete graph $K_n$ with 3 colors such that no $k$-clique is monochromatic (that is, has all its edges colored the same way) as long as $8\left(\binom{k}{2}\right)(n-2)^3(\binom{k}{2}) \leq 1$.

   Hint: Use (with some modifications) the algorithmic version of the Lovasz Local Lemma.

2. (Building on the 2-SAT algorithm) A coloring of a graph $G$ is an assignment of a color to each of its vertices. If $k \geq 2$ is an integer, then $k$-coloring of $G$ is a coloring of $G$ with $k$ colors such that no two adjacent vertices have the same color. A graph $G$ is $k$-colorable if there exists a $k$-coloring of $G$. (For example, 2-colorable is the same as bipartite.)

   Let $G$ be a 3-colorable graph. Consider the following algorithm for coloring the vertices of $G$ so that no triangle is monochromatic. Start with an arbitrary coloring of vertices in $G$ with 2 colors. While there exists a monochromatic triangle in $G$, pick any such triangle $T$ and change the color of uniformly random vertex of $T$.

   (a) Derive an upper bound on the expected number of such recoloring steps needed for the algorithm to find a coloring with 2 colors with no monochromatic triangles. Your upper bound should be polynomial in the number of vertices in $G$.

   Hint: Let $C$ be a 3-coloring of $G$ with “colors” in $\{0, 1, 2\}$. Let $U$ be the set of vertices in $G$ assigned colors 0 or 1 by $C$. Keep track of the number of vertices in $U$ whose colors are the same as those assigned by $C$.

   (b) If you only used the fact that $|U| \leq n$ in the previous part, notice that we can permute the names of the colors in the 3-coloring $C$. Use this to slightly improve your bound in (b).

   (c) Now we change the algorithm to start with a uniformly random assignment of two colors to vertices (instead of an arbitrary one). How does it affect the expected number of recoloring steps?