CMSC 858S: Randomized Algorithms

Fall 2001

Homework Assignment #2 **(will be graded)**

Due date: Beginning of class on November 9, 2001

Please note: Please submit only a final, polished set of answers, and do not include unnecessary material. (For instance, please don't include your preliminary work/calculations/intuition that led to the final answer.) Partial credit will be given where appropriate: if you think you have some ideas that deserve partial credit, please *itemize them concisely*, so that the grading process can be accurate.

1. Recall that in the $G(n,p)$ random graph model, we start with n vertices labeled with distinct labels, and *independently* put an edge between each (unordered) pair of distinct vertices with probability p. Suppose we draw a random graph G using this random process, for some given n and p. Let t and k be some given integers. Let $\mathcal{A}$ be the event "every vertex of G has degree at most t", and let $\mathcal{B}$ be the event "there are at least k cycles in G". Is it true that $\Pr[\mathcal{A} \wedge \mathcal{B}] \leq \Pr[\mathcal{A}] \cdot \Pr[\mathcal{B}]$? (**Hint:** Can some correlation inequality be of help?) **(4 points)**

2. Suppose we are given a finite set V and a collection of subsets $S_1, S_2, \ldots, S_m$ of V. Suppose further that $\sum_{i=1}^{m} 2^{1-|S_i|} < 1$.

(i) Show that there is a two-coloring of V in which no S_i is monochromatic. **(1 point)**

(ii) Use some derandomization approach to derive a deterministic polynomial-time algorithm to two-color V in a way that leaves no S_i monochromatic. **(4 points)**

3. Suppose $G = (V, E)$ is an undirected graph. Recall that a *proper coloring* of V with some given set of colors (if such a coloring exists) is an assignment of one color to each vertex such that no two adjacent vertices get the same color. A related useful notion which arises in various settings such as timetabling, is *list-coloring*. Here, we are also given a set $L(u)$ of possible colors for each vertex u. Now, we must (if possible) choose a color *from* $L(u)$ *for each vertex* u in such a way that no two adjacent vertices get the same color.

In this problem, we want to show that in certain situations, there always exists some legal list-coloring. Suppose, for some positive integer ℓ, $|L(u)| = 8\ell$ for each u; moreover, suppose that for each vertex u and each $c \in L(u)$, u has at most ℓ neighbors v such that $c \in L(v)$. Show that there exists a legal list-coloring in such a case. (**Hint:** Start with a natural randomized coloring algorithm. Now carefully define a suitable set of "bad" events, and use the Lovász Local Lemma to show that we can avoid all of these events with positive probability.) **(6 points)**