15-854 Approximation and Online Algorithms

Homework # 1 **due: February 7, 2000**

Groundrules:

- Homeworks will generally consist of easier *exercises* and harder *problems*. You should do the exercises by yourself, but you may work with a friend on the problems if you want — just write down who you worked with. One exception: no fair working with someone who has already figured out (or already knows) the answer.
- If you've seen a problem before (sometimes I'll give problems that are "famous"), then say that in your solution (it won't affect your score, I just want to know). Also, if you use any sources other than those on the web page, write that down too. It's fine to look up a complicated sum or inequality, but don't look up an entire solution.

Definitions: A vertex coloring of a graph is an assignment of "colors" to the nodes so that no two adjacent nodes get the same color. A graph is *k-colorable* if you can do this using at most k different colors. A set of nodes in a graph is an *independent set* if no two of them are connected to each other by an edge. So, saying that a graph is k-colorable is equivalent to saying that the vertices can be partitioned into k independent sets. There are fast algorithms that will 2-color any 2-colorable graph, but the 3-coloring problem (given a graph G, is it 3-colorable) is NP-complete.

Exercises:

1. Give a polymonial-time algorithm that, given a graph G, will color it using at most $d(G) + 1$ colors, where $d(G)$ is the largest degree of any node in the graph.
2. What is the relationship between a vertex cover and an independent set? Suppose a graph has a vertex cover that that itself is also an independent set: what does that tell you about the graph?

Problems:

3. (a) Give an approximation algorithm that will color any 3-colorable graph with $O(\sqrt{n})$ colors, where n is the number of nodes in the graph.

   Hint: use exercise 1 and the fact that the neighborhood of any node in a 3-colorable graph must be 2-colorable.

   (b) Extend your result from part (a) to color any 4-colorable graph with $O(n^{2/3})$ colors.
4. Prove that if we could approximate the Vertex Cover problem to a factor of $1.5 - \epsilon$ for some constant $\epsilon > 0$, then we could color any 3-colorable graph with $O(\log n)$ colors.
5. Here is a greedy algorithm for the vertex cover problem: pick the vertex of highest degree and put it into your cover. Then, delete all edges that have been covered and repeat. Come up with a counterexample showing that, strangely enough, this does *not* guarantee a factor of 2 approximation ratio.