CS 38 Introduction to Algorithms **Spring 2014**

Problem Set 7

Out: May 27 *Due:* **June 3**

Reminder: you are encouraged to work in groups of two or three; however you must turn in your own write-up and note with whom you worked. You may consult the course notes and the optional text (CLRS). The full honor code guidelines can be found in the course syllabus.

Please attempt all problems. **To facilitate grading, please turn in each problem on a separate sheet of paper and put your name on each sheet. Do not staple the separate sheets.**

1. We are given subsets $S_1, S_2, \ldots, S_n$ of an m-element universe U. Recall that a cover is set I such that
$$\bigcup_{i \in I} S_i = U.$$
In lecture 18, we saw a $(\ln m + 1)$-approximation algorithm for this problem. In this problem you are asked to give a k-approximation algorithm for SET COVER in the special case that no element $u \in U$ appears in more than k subsets. Hint: solve an LP relaxation, and round.

2. In the MAX CUT problem, we are given an undirected graph $G = (V, E)$ and we seek a *cut* with the maximum number of edges crossing it. An edge (u, v) crosses a cut $S \subseteq V$ iff one of u, v is in S and the other is not in S. This problem is NP-complete. Formulate an LP relaxation and rounding scheme that yields a 2-approximation algorithm for this problem. Hint: your LP should have variables for each vertex and for each edge.

3. Faster maximum matching in a general graph. We saw in lecture 12 how to find a maximum bipartite matching in time $O(mn^{1/2})$ time, and we asserted that a more complex algorithm achieved a similar time bound for general graphs. In this problem you will show that if one is satisfied with a 2-approximation, then an $O(m)$ running time can be achieved by a very simple algorithm.
   (a) A *maximal* matching is a matching that cannot be enlarged by adding any edge. Show that one can find a maximal matching in $O(m)$ time in a general undirected graph.
   (b) Prove that every maximal matching has cardinality at least one-half of the cardinality of a maximum matching. Hint: relate both quantities to the cardinality of a minimum vertex cover.

4. Bin packing. We are given a list of real numbers $a_1, a_2, \ldots, a_n$ and we wish to *pack* them into as few unit-capacity bins as possible. A unit-capacity bin can accommodate any subset of the a's whose sum is at most 1. This problem is NP-hard (it is even NP-complete to determine if 2 bins suffice).
A simple strategy is to place each a_i in turn into the first bin into which it can fit. Prove that this strategy is a 2-approximation algorithm for bin-packing.