MCS 401 – Computer Algorithms I
Spring 2016
Problem Set 6

Lev Reyzin

Due: 4/27/16 by the beginning of class

Instructions: Atop your problem set, write your name and whether you are an undergraduate or graduate student. Also write the names of all the students with whom you have collaborated on this problem set.

1. **[10 pts]** We define the decision version of the Minimum Cut problem as follows: given an undirected, unweighted graph $G = (V, E)$ two distinct vertices $s, t \in V$, and a number k, does there exist an $s-t$ cut of value at most k?

For each of the two questions below, decide whether the answer is (i) "Yes", (ii) "No", or (iii) "Unknown, because it would resolve the question of whether $\mathcal{P} = \mathcal{NP}$." Give an explanation of your answers.

(a) Question: Is it the case that Minimum Cut $\leq_p$ Vertex Cover?

(b) Question: Is it the case that Independent Set $\leq_p$ Minimum Cut?

2. **[10 pts]** Consider a set of monotone clauses $C_1, \ldots, C_k$ over a set of Boolean variables $x_1, \ldots, x_n$. A clause is monotone if each term in the clause consists of an unnegated variable, for example $(x_1 \vee x_3 \vee x_4 \vee x_7)$ is a monotone clause. Satisfying a collection of monotone clauses is easy – simply set all the variables appearing in the clauses to 1.

However, it is natural to ask how few variables need to be set to 1 in order to satisfy a collection of monotone clauses. The Miserly Satisfiability problem asks the following: given a collection of monotone clauses and a number k, is it possible to satisfy all of the clauses without setting more than k variables to 1? Prove that Miserly Satisfiability is NP-complete.

3. **[10 pts]** Suppose someone gives you a black-box B that takes in any undirected graph $G = (V, E)$ and a number k, and in unit time it returns "yes" if G has an independent set of size at least k and "no" otherwise. Design an algorithm with the power to access B as many times as it wishes that returns an independent set of maximum cardinality in a given graph in polynomial time.

4. **[10 pts]** Earlier, we considered the following clustering problem: given objects $p_1, \ldots p_n$, and distances $d(\cdot, \cdot)$ on them (with $d(p_i, p_i) = 0$, $d(p_i, p_j) = d(p_j, p_i)$, and $d(p_i, p_j) > 0$ for $i \neq j$), divide objects into k sets so as to maximize the minimum distance between any pair of objects in distinct clusters. This turned out to be efficiently solvable by a nice application of minimum spanning trees.

A different but seemingly related way to formalize the clustering problem would be as follows: divide the objects into k sets so as to minimize the maximum distance between any pair of objects in the same cluster. Whereas the formulation in the previous paragraph sought clusters so that no two were "close," this new formulation seeks clusters so that none of them is too "wide."

Given the similarities, it is perhaps surprising that this new formulation is computationally hard to solve optimally. First, lets write it first as a yes/no decision problem: given n objects $p_1, \ldots, p_n$, with distances on them as before, a number k, and a bound B, we can ask: can the objects be partitioned into k sets, so that no two points in the same set are at distance greater than B from one another?

Prove that this new formulation is NP-complete.