Homework 7
Due 08/07/2008

Warm-Up Exercises

1. Solve the quizzes from last week and verify that you understand their solutions.[1]

2. Let A be a *minimization* problem. The decision problem Gap-$A[\alpha, \beta]$ is, given an input x, decide whether there exists a solution for A of size at most α or every solution is of size at least β (the other instances are not allowed). Recall that an algorithm is a c-approximation for A if it finds a solution of size at most c times the optimal one ($c > 1$). Prove that if there exists a polynomial-time c-approximation algorithm for A for some $c < \frac{\beta}{\alpha}$ then Gap-$A[\alpha, \beta] \in$ P.

Exercises for Submission

1. (a) Prove Markov's inequality. That is, show that if W is a nonnegative random variable whose expectation is μ then for any $a > 0$, $\Pr[W \geq a] \leq \frac{\mu}{a}$.

   (b) Let W be a random variable whose expectation is $\mu > 0$ and whose maximal value is at most 2μ. Prove that for any $0 < \varepsilon < 1$, $\Pr[W \leq (1 - \varepsilon)\mu] \leq 1 - \frac{\varepsilon}{2}$.

2. Suppose that a CNF formula has less than n^k clauses, each with at least $k \log_2 n$ distinct variables. Use the probabilistic method to show that it has a satisfying assignment.

3. Recall that in the weighted MAX-CUT problem, given a graph $G = (V, E)$ and a weight function $w : E \to \mathbb{R}^+$, the goal is to determine a cut $(S, V \setminus S)$ that maximizes the total weight of the cut's edges. Consider the following algorithm: Initially, define $V_1 = V_2 = \phi$. Then, for each vertex $v \in V$, if $\Sigma_{u \in V_2} w(v, u) \geq \Sigma_{u \in V_1} w(v, u)$ add v to V_1 and otherwise add it to V_2. Finally return the cut (V_1, V_2).
   Prove that this is a $\frac{1}{2}$-approximation algorithm for weighted MAX-CUT, and that it can be implemented in polynomial running time.

4. (a) Show an undirected connected graph $G = (V, E)$ with n vertices and $\binom{n}{2}$ minimum cuts.

   (b) Show an undirected connected graph $G = (V, E)$ with n vertices and exponentially many maximum cuts. How many minimum cuts does your graph have?

   (c) Consider running the contraction algorithm for MIN-CUT until the number of vertices is reduced to t and then using a cubic-time algorithm to find the min-cut in the contracted graph. Show that repeating this process as many times as necessary to ensure a probability of success at least $\frac{1}{2}$ leads to an algorithm with running time $\Omega(n^{8/3})$.

5. Show that if TSP can be approximated to within some constant $c > 1$ in polynomial-time then P = NP.[2] Does your proof work also for non-constant factors (i.e., factors that depend on the input size)?
   Hint: Show that given a c-approximation algorithm for TSP one can solve the Hamilton Cycle/Path problem.

[1] The quizzes and their solutions are available in the course web page.
[2] Note that we do *not* assume triangle inequality.