Computational Complexity

Exam #2-Take Home
Start Date: 31/2/1398 7:00 a.m.
Due Date: 3/3/1398 24:00 p.m.

Some Rules.
1. Please hand in your worksheet prior to 24:00 p.m. Friday 3 Khordad 1398. Overdue papers are not acceptable at all.
2. You are allowed to use the Internet and the textbooks. You are NOT allowed to talk with your classmates about the problems and the solutions.
3. Partial answers are appreciated, so write down your thinking even though you may think it is insufficient or imperfect.

Problem 1.
Let G be a graph on the vertex set $V(G) = \{1, 2, \ldots, n\}$ with e edges. Let m_{ij} be the expected number of steps that a random walk starting from i takes until first visit of j. We proved in the class that for every vertex i, $m_{ii} = 2e/\deg(i)$ and for every $i \neq j$, when G is regular, $m_{ij} = O(n^3)$. Prove that for general graph, we have $m_{ij} = O(n^5)$.
(Hint: Add some loops on vertices to get a regular graph G'. Then, simulate each step of a random walk on G' with a random walk on G.)
Can you prove that $m_{ij} = O(n^4)$?

Problem 2.
Let A be a minimization problem. For every instance x, we have a lot of solutions y. For every solution y we have a cost $c(y)$ and we aim to minimize the cost $c(y)$ over all solutions y. We denote the minimum by $OPT(x)$.

$$OPT(x) = \min_{y \text{ is a solution for } x} c(y).$$

Let A and B be two minimization problems with cost functions c_A and c_B respectively. We say that A is reducible to B (denoted by $A \leq B$) if there is polynomial-time computable functions f and g and two constants $\alpha, \beta > 0$, which satisfy the following properties:

1. For any instance x of A, $f(x)$ is an instance of B such that $OPT_B(f(x)) \leq \alpha OPT_A(x)$.
2. For any solution y of instance $f(x)$ with cost $c_B(y)$, $g(y)$ is a solution of x with cost $c_A(g(y))$ such that
$$|c_A(g(y)) - OPT_A(x)| \leq \beta |c_B(y) - OPT_B(f(x))|.$$

(a) Prove that if $A \leq B$ and $B \leq C$, then $A \leq C$.

(b) Prove that if $A \leq B$ and for some constant ρ, B admits a ρ-approximation algorithm, then there is a constant ρ' such that A admits a ρ'-approximation algorithm.

Problem 3.
Here we are going to prove that if $\mathbf{PCP}(o(\log n), 1) = \mathbf{NP}$ then $\mathbf{P} = \mathbf{NP}$. So, PCP Theorem is probably optimal up to constant factors.

(a) Prove that if $L \in \mathbf{PCP}(r(n), q(n))$, then for every instance x of length n, one may construct a 3-CNF φ_x such that

(1) $x \in L$ iff φ_x is satisfiable.

(2) $|\varphi_x| \leq O(q(n)2^{q(n)}r(n)2^{r(n)})$.

(3) φ_x can be computed in time $O(2^{r(n)}2^{q(n)}n^c)$ for some constant c.

(b) Prove that if $\mathtt{3-SAT} \in \mathbf{PCP}(o(\log n), 1)$, then there is a constant $d > 1$ and an integer n_0 such that for every 3-CNF φ, one may construct in polynomial-time a 3-CNF ψ_φ where

(1) φ is satisfiable iff ψ_φ is satisfiable.

(2) If $|\varphi| \geq n_0$, then $|\psi_\varphi| \leq d|\varphi|^{1/2}$.

(c) Applying (b) several times, prove that there is an integer n_0 such that for every 3-CNF φ, one may construct in polynomial-time a 3-CNF ψ_φ where

(1) φ is satisfiable iff ψ_φ is satisfiable.

(2) $|\psi_\varphi| \leq n_0$.

(d) Prove that if $\mathbf{PCP}(o(\log n), 1) = \mathbf{NP}$, then $\mathbf{P} = \mathbf{NP}$.

Good Luck!