CS180 Final Exam

For all the algorithms you design, in addition to describe your algorithm clearly, please also (a) briefly justify the correctness of the algorithm; (b) present the time complexity of the algorithm and briefly justify the reason. Partial credits will be given if your algorithm has complexity slightly worse than the solution for all the problems.

1. (25 pt) Answer **true or false** for the following questions (you do not need to justify your answer):

   (a) If $P \neq NP$ then every problem in NP requires exponential time.

   (b) If G is a weighted graph, adding a constant to all its weights won't change the shortest path between two nodes (u, v).

   (c) If the input graph has negative weight but is a Diected Acyclic Graph (DAG), Dijkstra's algorithm can output the correct shortest paths.

   Solve the following recurrences below to give tight bounds $T(n) = O(f(n))$ for an appropriate function f.

   (d) $T(n) = 7T(\frac{n}{8}) + n \log n$

   (e) $T(n) = 9T(\frac{n}{3}) + n^2$

2. (25 pt) There's a length n ($n \geq 3$) array and we'd like to fill in each element of the array with an integer. The integers should be in $\{1, 2, \ldots, k\}$ and any 3 adjacency entries cannot have the same number. Given n and k , design a polynomial time algorithm to compute the number of ways to fill this array. Please also justify the correctness and the time complexity of your answers.

3. (25 pt) Assume there are n courses offered by the university, where each course has one or no prerequisites. If course j is the prerequisite for course i, then we are only allowed to take course i after course j. And after taking each course i, we can get a reward r_i. Given the reward and prerequisite for each course, design an algorithm to find the maximum total reward we can get by taking m courses. The time complexity should be $O(nm^2)$.

4. (25 pt) Consider the following problem:

   Problem A: Given a directed graph G and an integer k, is it possible to remove $\leq k$ nodes (along with edges associated with those nodes) such that the resulting graph has no cycle?

   Prove this problem is NP-complete. (Hint: can you reduce a vertex cover problem to this Problem A?)