Topics in Approximation Algorithms 2006: Mid-Term Exam

[Answer all the questions. Each question carries $8\frac{1}{3}$ marks.]

Q1. Show that the following problem is NP-hard.

Given n positive integers $x_1, \ldots, x_n$, we want to partition these numbers into k subsets $S_1, \ldots, S_k$ so that $\sum_{i=1}^{k} \prod_{x_j \in S_i} x_j$ is minimized.

(You can use the NP-hardness of the problems discussed in the course so far.)

Q2. Show that the following problem cannot be approximated within any polynomial time computable factor α, unless P = NP.

Given a graph $G = (V, E)$ with positive weights on its edges, and a positive integer k, find subsets S of vertices of cardinality k such that the total weight of edges in the subgraph induced by S is minimized.

Q3. Give a factor f approximation algorithm for the set cover problem, where f is the frequency of the most frequent element.

Q4. Approximation algorithms are also designed for problems which can be solved exactly in polynomial time. This is motivated by the fact that the approximation algorithms are typically much simpler than the exact algorithms. One such problem is the maximum weight matching problem. There are polynomial time algorithms to compute a maximum weight matching in an arbitrary graph. However, they are not very simple and their running times are rather high.

Design a simple 1/2-approximation algorithm to compute a maximum weight matching in an arbitrary graph.

(*Hint:* Consider the greedy approach.)

Q5. Consider the following problem, which we call the MAX-3-CUT problem. Let $G = (V, E)$ be an undirected graph with non-negative edge weights. We wish to partition the vertex set V into 3 subsets V_1, V_2, V_3 so that the total weight of edges whose endpoints are in different sets V_i, V_j (call these edges cross edges) is maximized. This is an NP-hard problem.

Design a 2/3-approximation algorithm for the above problem.

(*Hint:* Again consider the greedy approach. Start with $V_1 = V_2 = V_3 = \emptyset$. Consider each vertex and add it to the V_i that causes the largest increase in the weight of cross edges.)

Q6. We wish to solve the minimum vertex cover problem on input instances that consist of a graph G, together with a valid vertex colouring of G with 3 colours. Give a 4/3-approximation algorithm for the minimum cardinality vertex cover problem on such instances.

(*Hint:* You can use the fact that we can compute a minimum cardinality vertex cover in a bipartite graph in polynomial time.)