Due Date: November 4, 2014

Problem 1: Given a finite set $A = \{a_1, \ldots, a_n\}$ of natural numbers and two positive integers K and J, show that it is NP-complete to decide whether there exists a partition $A_1, A_2, \ldots, A_K$ of A such that $\sum_{i=1}^{K} \left(\sum_{a \in A_i} a\right)^2 \leq J$.

Problem 2: Describe a linear-time greedy algorithm to find the maximum independent set in a tree.

Problem 3: Given an undirected graph $G(V, E)$, we need to color its vertices so that the two endpoints of any edge do not receive the same color.

(i) Describe a linear-time algorithm to determine whether G can be colored with two colors.

(ii) Give a greedy linear-time algorithm for coloring G with $\Delta + 1$ colors, where Δ is the maximum degree of the graph.

(iii) Give a polynomial-time algorithm for coloring a 3-colorable graph with $O(\sqrt{n})$ colors. (**Hint:** *For any vertex v, the induced subgraph on its neighbours, $N(v)$, is 2-colorable. If v has degree at least $\sqrt{n}$, then color $\{v\} \cup N(v)$ with 3 colors, delete them from G and continue till every vertex has degree $\leq \sqrt{n}$.*)

Problem 4: [This problem is from the midterm.] Consider the following fractional LP for computing the minimum spanning tree (this is the same LP given in the midterm examination).

$$\begin{aligned} \sum_{e \in E(S)} x_e &\leq |S| - 1 \quad \forall S \subseteq V \\ \sum_{e \in E} x_e &= |V| - 1 \\ x_e &\geq 0 \end{aligned}$$

Show that a primal-dual algorithm to solve this LP is the same as running Kruskal's algorithm, i.e., the sequence of edges picked up by Kruskal's algorithm is the same as the one picked by the primal-dual algorithm. Describe how will you set/update the dual variables at each step, and then show that the order in which the constraints become tight is the same as the edge sequence chosen by Kruskal's algorithm.

Problem 5: The *maximum clique* problem asks to find the largest clique in an input graph $G = (V, E)$. Show that if there is a polynomial-time $(1/2)$-approximation algorithm for this problem, then you can get a polynomial-time $(1/\sqrt{2})$-approximation algorithm for the same. (**Hint:** *Make $|V|$ copies of G, connect appropriate nodes between the copies by edges, and show that a $(1/2)$-approximation algorithm on this graph leads to a $(1/\sqrt{2})$-approximation algorithm for G.*)