Problem Set 9

This problem set is due at **10:00 am** on **Tuesday, May 2nd**.

Problem 9- 1: Vertex Cover

Recall that the vertex cover problem has a 2-approximation. A k-hypergraph is a graph such that each 'edge' is a set of k vertices (Since each edge so far has been between two vertices, the graphs we have seen so far have been 2-hypergraphs). A vertex cover for a k-hypergraph is a subset of vertices $C \subset V$ such that each hypergraph edge has a vertex in C. Design a k-approximation algorithm for vertex cover on a k-hypergraph. Prove the correctness of your algorithm and analyze its runtime.

Problem 9- 2: Load Balancing

In class we discussed two approximation algorithms for an NP-hard load balancing problem in which we are asked to distribute n tasks, each with a positive integer weight, over m machines in a way that minimizes the maximum load of any machine. The first algorithm that we presented guarantees a maximum load within a factor of two of optimal. The second algorithm that we presented guarantees a maximum load within a factor of $\frac{3}{2}$ of optimal. These two algorithms are also presented in Section 11.1 of the text, where they are referred to as GREEDYBALANCE and SORTEDBALANCE, respectively. The purpose of the present question is to develop a tighter analysis of algorithm SORTEDBALANCE. Specifically, we will prove that this algorithm achieves an approximation factor of $\frac{4}{3}$. Fix an instance I of the load balancing problem with m machines and n tasks. Number the tasks from 1 to n, and let w_i denote the positive integer weight of task i. Assume without loss of generality that the tasks are numbered in such a way that $w_i \geq w_{i+1}$, $1 \leq i < n$. Let L^* denote the minimum possible maximum machine load. Let k denote the number of tasks with weight greater than $L^*/3$.

(a) Explain why k is at most $2m$.

(b) Let L^{**} denote the minimum possible maximum machine load when tasks 1 through k are assigned to the machines (and the rest of the tasks are not assigned to any machine). Prove that at the point in its execution when algorithm SORTEDBALANCE has processed tasks 1 through k, the maximum load of any machine is exactly L^{**}. Hints: You may find it useful to consider the cases $0 \leq k \leq m$ and $m < k \leq 2m$ separately. In each case, begin by precisely characterizing of the way that algorithm SORTEDBALANCE maps tasks 1 through k to the machines.

(c) Prove that algorithm SORTEDBALANCE achieves an approximation ratio of $\frac{4}{3}$. Hint: Use the same overall framework as we used to establish the $\frac{3}{2}$ bound, but improve the bound by making use of the results of parts (a) and (b).