General instructions: Each problem is worth 10 points. For each of the algorithm problems, design a polynomial-time approximation algorithm with a performance ratio as indicated. Give a proof of the approximation guarantee and time analysis for each algorithm. You may use any well-known algorithm or data structure, or algorithm from the texts or from class, as a sub-routine, without needing to provide details. 3. Don’t do literature searches, just use the textbooks.

**Average Finish Time** The Minimum Average Finish Time Schedule on Two Machines is as follows: You are given a list of $n$ jobs of positive real durations $d_1,..d_n$. You must assign each job to one of two identical machines, and order the jobs on each machine. The finish time of a job $i$ is the sum of all the durations of all jobs assigned before it on the same machine, plus its own duration. The objective is to minimize the sum of finish times of all jobs. Give a fully polynomial-time approximation scheme (FPTAS) for this problem.

**Heterogenous Load Balancing** Consider a heterogenous version of the load balancing problem where each of $m$ processors has a speed, $s_i$ (in cycles/unit time). Each job has a requirement $T_i$ in cycles, and needs to be performed by exactly one processor. We want to minimize the time when all jobs are finished, i.e., $\max_{1 \leq i \leq m} (\sum_{jobs } j assigned to processor i T_j)/s_i$.

Give an efficient approximation algorithm for this problem. (Hint: the following has ratio 2: Sort the jobs from largest to smallest. Assign each job to the machine that would finish it first, i.e., if machine $i$ would finish its current jobs at $F_i$, minimize $F_i + T_j/s_i$.)

**Bounding Sphere** Say that $p_1,..p_n$ are points in $\mathbb{R}^d$, the $d$-dimensional Euclidean space, i.e., each $p_i$ is a vector of $d$ real numbers. A bounding sphere $S(q, R)$ is a sphere centered at point $q$ (which is not necessarily one of the input points) of radius $R$ that contains all of the $p_i$, i.e., the distance between $q$ and every $p_i$ is at most $R$. The Minimum Radius Bounding Sphere problem is, given such a set of points, find the bounding sphere with smallest radius $R$. Give a polynomial time approximation algorithm for this problem with ratio strictly less than 2. (Note: we will return to this problem after we study convex optimization.)

**Steiner trees for $[1,2]$ -metrics** A metric space on a set of $n$ elements $x_1,..x_n$ is a distance function $d(x_i, x_j)$ satisfying $d(x, x) = 0$ and the triangle inequality $d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$ for every three points $x_i, x_j, x_k$. The Steiner problem is: given a metric space and a subset $R$ of required
elements, find a tree that contains all the required elements (and possibly some others) that minimizes the total distances along all edges. A [1,2]-metric space is one where $1 \leq d(x_i, x_j) \leq 2$ for every two distinct points $x_i, x_j$. Note that this already implies the triangle inequality. Give a polynomial-time approximation algorithm to the Steiner problem for [1,2]-metric spaces that has approximation ratio strictly less than 2.