COMS 511 - Homework 11

Due: April 28 11:59 PM

Guidelines

- When proofs are required, you should make them both clear and rigorous. Do not hand-waive.
- Your assignment needs to be submitted via Canvas.
  - You **must** type your solutions. Please submit a PDF version.
  - Please make sure that the file you submit is not corrupted and that its size is reasonable (e.g., roughly at most 10-11 MB).

    If we cannot open your file, your homework will not be graded.
- The following are examples of activities that are prohibited:
  - Sharing solutions or fragments of solutions (e.g., via email, whiteboard, handwritten, or printed copies).
  - Post solutions or fragments of solutions in a location accessible to others.
  - Using solutions or fragments of solutions provided by other students (including students who had taken the course in the past).
  - Using solutions or solution fragments obtained on the Internet or from solution manuals for textbooks.
  - Using material from textbooks, reference books, or research articles without properly acknowledging and citing the source.
- Concerns about grading should be expressed within one week of returning the homework.
- **No late homework is accepted** with the exception of at most one late submission up to 12 hours late.

Problem

Problem 1. (50 points)

Let $G = (V, E)$ be a graph with nonnegative edge costs. S and R are two disjoint subsets of V, where S is a set of *senders* and R is a set of *receivers*. The problem is to find a minimum cost subgraph of G that has a path connecting each receiver to a sender (any sender suffices).

(1) If $S \cup R = V$, show that this problem is in P.

(2) If $S \cup R \neq V$, show that this problem is NP-hard. Give a 2-approximation algorithm for this case.

(Hint: Consider adding a new vertex connected to each sender by a zero cost edge and transforming to a problem of finding a minimum cost Steiner tree.)

Problem 2. (50 points)

Suppose there are m identical machines and n jobs with processing times $t_1, ..., t_m$. Let $S[i]$ be the subset jobs assigned to machine i. The *load of machine* i is $L[i] = \sum_{j \in S[i]} t_j$. The *makespan* of an algorithm is the maximum load on any machine $L = max_i L[i]$. The *Load Balancing* problem is to find an assignment of jobs to machines so as to minimize the makespan.

(1) Show that this problem is NP-hard by reduction from Subset-Sum.

(2) Give a 2-approximation algorithm for Load-Balancing.

(3) Further improve the algorithm in (2) and give a 3/2-approximation algorithm.