

# CS 270 COMBINATORIAL ALGORITHMS & DATA STRUCTURES — Spring 2023

## PROBLEM SET 6

Due: 11:59pm, Friday, March 24th  
Solution maximum page limit: 4 pages

See homework policy at <https://cs270.org/spring23/syllabus/#homework-policies>

**Problem 1:** In weighted vertex cover we have an undirected graph with  $n$  vertices and  $m$  edges. Each vertex  $v$  has a cost  $c_v > 0$ . We must choose a subset  $S$  of the vertices such that

- Each one of the  $m$  edges is incident to at least one vertex in  $S$ .
  - $\sum_{v \in S} c_v$  is minimized amongst all subsets  $S$  of vertices satisfying the previous bullet.
- (a) (3 points) Modify the primal LP formulation from Lecture 18 to handle costs, and write the new dual. (This should be a minor modification of what was done in class.)
- (b) (7 points) Give a greedy algorithm for weighted vertex cover achieving a 2-approximation. **Hint:** Maintain a feasible dual solution and build up a feasible primal solution (i.e. as long as an edge is not satisfied, cover it using one or both of its endpoints). The primal solution you maintain should be fractional, and only take a vertex into  $S$  once its variable becomes big enough.

**Problem 2:** In this and the next problem, we will work toward a PTAS for a scheduling problem. There are  $n$  jobs that we would like to schedule on  $m$  machines. Job  $i$  requires processing time  $p_i$ , and each job must be assigned to exactly one machine. The completion time of a machine is then the sum of processing times of jobs assigned to it, and the “load” of an assignment is the maximum completion time of any machine in that assignment. We want to minimize load.

- (a) (2 points) Let  $p_{max}$  be the maximum processing time of any job. Show that the quantity  $\max\{p_{max}, (1/m) \sum_{i=1}^n p_i\}$  is a lower bound on OPT.
- (b) (3 points) Consider a greedy algorithm which loops over the jobs from 1 to  $n$ , and for job  $i$  assigns it to the currently least loaded machine (the one whose completion time when considering jobs assigned so far is smallest). Show that the greedy algorithm achieves completion time at most  $(2 - 1/m)\text{OPT}$ . **Hint:** Use (a).

**Problem 3:** Now suppose we want a PTAS for the scheduling problem from Problem 2, i.e. to achieve load at most  $(1 + \varepsilon)\text{OPT}$  with an algorithm whose running time is polynomial in the input size (the exponent of this polynomial can be any function of  $1/\varepsilon$ ). Define the “big” jobs to be  $B = \{i \in [n] : p_i \geq \varepsilon\text{OPT}\}$  and the “small” jobs to be  $S = \{i \in [n] : p_i < \varepsilon\text{OPT}\}$ .

- (a) (2 points) Show that if the jobs in  $B$  are assigned to machines to achieve load  $L$ , then greedily assigning the remaining jobs in  $S$  achieves load at most  $\max\{L, (1 + \varepsilon)\text{OPT}\}$ .
- (b) (4 points) Show that if there are at most  $k$  distinct job processing times across all jobs, then for any  $T$  there is a dynamic programming algorithm which finds a schedule achieving load at most  $T$  in time  $O(n^{2k})$  (or reports if no such schedule exists).
- (c) (5 points) Conclude that, if we know  $\text{OPT}$ , we can schedule all jobs with load at most  $(1 + \varepsilon)\text{OPT}$  in time  $n^{O(\log(1/\varepsilon)/\varepsilon)}$ . **Hint:** For  $i \in B$ , round  $p_i$  to  $\varepsilon(1 + \varepsilon)^j$  for integer  $j$ .
- (d) (4 points) In actuality we don't know  $\text{OPT}$ . Show, nonetheless, that there is a PTAS for our scheduling problem.

**Problem 4:** (1 point) How much time did you spend on this problem set? If you can remember the breakdown, please report this per problem. (sum of time spent solving problem and typing up your solution)