

CS302 - Problem Set 4

1. Suppose you are hosting a music festival, and are trying to decide which bands should play in the prime-time slot. You randomly surveyed 100 of the registered attendees to have them rank the bands who will be at the festival. You have an unlimited number of stages, so you can have many different bands play at the same time. You would like to choose a subset of bands to play during the prime time spot such that (1) most attendees will be excited about at least one of the bands, and also (2) most attendees will not be excited about more than one of the bands (since this would make it so that they couldn't see both). For example, suppose some of the attendees like R&B, and some like metal, but not many people like both R&B and metal. Then you could have the most popular R&B band and the most popular metal band play the prime time slot. This would be better than having both of your R&B bands play in the prime time slot, even if they are both more popular overall than any of the metal bands, because then the metal fans would not have anyone to watch in the prime time slot, and the R&B fans would be annoyed that they would have to pick one of the bands to watch and couldn't see both.
 - (a) Describe a reduction from this problem to the Max Weight Independent Set (MWIS) problem. The input to our problem is the survey data on the bands, and the output should be a set of bands to play in the prime time slot.
 - (b) Explain why the reduction you described gives a good solution to the original problem?
 - (c) Now that you've figured out who should perform in the prime time slot, how should you pick which bands should perform in the second best slot using MWIS?
2. You have a shipping container that can hold W worth of weight (please ignore volume). You have n types of items that you can ship. Item i is w_i pounds and is worth v_i dollars. However, you can take a fraction of any item, so if you wanted, you could pack $p \times w_i$ pounds of item i , which would be worth $p \times v_i$ dollars, where $0 < p \leq 1$. You would like to pack the shipping container so that the total value is as large as possible.

Consider a greedy algorithm that evaluates a function f for each item. Then the algorithm puts as much of the largest f -valued item in as possible. If there is still room remaining, the algorithm goes to the next largest f -valued item, and so on.

 - (a) What function f should you use to rank items? (The obvious one is the one to go with!)

- (b) Let's rename the items according to their f score so that item 1 has the highest f -score, 2 the 2nd highest, and so on. (You may assume all f -values are unique.) Because of the limited size of the shipping container, suppose that with the optimal strategy we only end up putting the first m types of items into the container. This means we put all of items $1, 2, \dots, m - 1$ in the container, and we put some or all of item m into the container. Prove using a proof by contradiction that any other strategy that doesn't put all of items $1, \dots, m - 1$ and as much as possible of item m into the container is not optimal.
- (c) What is the runtime of this greedy algorithm?
3. Suppose you have n jobs, and each job i takes time $t_i > 0$ and has a deadline $d_i > 0$. Let the completion time C_i of a job be as defined as in class. Given an ordering of the jobs, we call $l_i = C_i - d_i$ the lag of a job i . We would like to minimize the maximum lag: $\max_j l_j$.
- (a) Propose function f to use to order jobs, and explain why it is reasonable given the objective.
- (b) Give a counter example to show that your greedy choice of f is not optimal.
- (c) (Challenge) Determine the function f that gives the optimal ordering. (See last page for solution.)
- (d) Prove optimality of the greedy ordering from part (c).
- (e) Briefly describe your algorithm and state its runtime.
4. Approximately how long did you spend on this assignment?

Hints

2a: $f(i) = v_i/w_i$

2b: Here is a way to start the proof:

Suppose for contradiction that the optimal a strategy does not put all of $1, 2, \dots, m-1$, and as much of m as possible into the container. There are two cases: there is some $k \in \{1, 2, \dots, m-1\}$ such that not all of item k is included, or an item with label $l \geq m$ is included. (These two cases deal with all possible differences, including putting too much of item m in instead of all of a lower numbered item, for example.)

3c: The optimal is $f(i) = d_i$.

3d: Hints for proof: The exchange is the same as in class. There are three cases to consider: that the max lag is not one of the exchanged items, that the max lag is the first of the exchanged items, or the max lag is the second of the exchanged items. Remember that, given our ordering if $i < j$, then $d_i < d_j$, and if $l_i \leq l_j$ then $C_i - d_i \leq C_j - d_j$. You'll end up with a bunch of inequalities and you'll have to show that the max lag either stays the same or decreases.