

*This homework covers designing greedy algorithms. It is due in class Wednesday, April 15.*

*See the Policies page on the course website for information about revise-and-resubmit, late work, and academic integrity as it applies to homework.*

*Write your solutions carefully — your work should be neat, readable, and organized. Keep in mind that what you hand in should be a presentation of your work, not your train-of-thought scratch work with a solution mixed in somewhere.*

---

1. Do the homework #10 drill problems on Canvas. (Look for `hw10 drill` in the Quizzes section.)

For the following problems, develop greedy algorithms using the process discussed in class. This is not a research task — the point here is to understand and be able to apply the process discussed in class to develop an algorithm yourself.

Show your work — and your understanding of the steps in the process — by including each of the steps in your writeup. Don't just give an algorithm! See the posted writeups from class for examples. (Don't include the marked "commentary" parts.)

2. Your company has been hired to add closed captioning to live television broadcasts. Given a list of programs to be captioned (and the start and end times of each program), assign those programs to your employees using as few employees as possible. Note that each person can only caption one show at a time.
3. You are creating a playlist for a party. There are  $n$  songs that you are considering including; song  $i$  has a length  $l_i$  and a popularity  $p_i$ . Your goal is to create the most popular playlist possible (the playlist's popularity is the sum of the popularities of the songs included), but the playlist can't last any longer than the party does (a total time of  $L$ ) so you have to decide which songs to include and which to leave out. However, you can include parts of songs; perhaps somewhat unrealistically, the popularity scales so that, for example, half of a song is half as popular as the whole song. Which songs do you include, and how much of each?
4. Print Services has  $n$  jobs to complete, and each job  $i$  has a length  $t_i$  and a deadline  $d_i$ . A feasible schedule is one in which all of the jobs get done before their respective deadlines. Find a feasible schedule if one exists.

Hint: As with the monster problem, sometimes having too many possible solutions makes it harder to show that your algorithm finds one of them (or to come up with an algorithm in the first place) — it can be easier when there's just one to find because there are more specific criteria to aim for. In this case, instead of simply looking for a feasible schedule, try to find the *best* schedule: Develop

a greedy algorithm to find a schedule which minimizes the amount by which the most-late job misses its deadline. Then explain how finding the least-most-late schedule allows you report a feasible solution or conclude that none exists.