

General instructions. Answer the problems given in the rest of this document. Edit this LaTeX template to add your answers, as described in the file named `00_README.txt`, then compile it to produce `homework_4.pdf`.

Don't know? Say so. If you don't know the answer to a problem or any part of the problem, we encourage you to simply write “*I don't know*” instead of giving an answer. (If you write “I don't know” you can also add any questions you might have.)

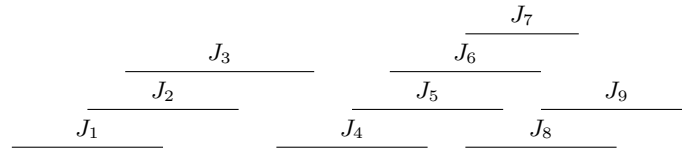
Gentle reminder: homework is for learning. As noted in [Lecture Note 1](#), engaging fully with the homeworks is the best way to learn the material. *The primary role of homeworks is to help you learn the material, and the primary role of feedback on homeworks is to help you improve your ideas.*

Please try to explain your ideas as clearly as you can. The clearer your explanations are, the more useful the feedback on them can be.

Collaboration and citation policy. If you are doing this homework as part of a course, follow the course's collaboration and citation policy. In any case, at the end of each answer, list the people you collaborated with, and cite every source that your answer uses ideas from (other than the lecture notes and the sources listed in the lecture notes). If there are none, write “none”.

Problem 1. (Section 5.3) Consider executing the Activity Selection algorithm `rselect` (as defined in Section 5.3) on the following instance of the Activity Selection problem:

$$I = \{[0, 4], [2, 6], [3, 8], [7, 11], [9, 13], [10, 14], [12, 15], [12, 16], [14, 18]\}.$$



- When `rselect(I)` executes, what jobs are in the set D that it computes in Line 3, before recursing?
- What set of jobs does `rselect(I)` output?
- Consider running the linear-time iterative implementation, `iselect` (Section 5.3, second definition) on input I . What values does the variable ℓ take on during the execution? Give them in increasing order.
- Consider the following “theorem” and “proof”. Is the proof correct? If not, what is the first step in the proof that makes an assertion that does not necessarily hold?

Theorem 1. Consider any instance $I' = \{[s_1, e_1], [s_2, e_2], \dots, [s_n, e_n]\}$ of Activity Selection. Let $X = \{x_1, x_2, \dots, x_k\}$ be a set of times such that, for every job $[s_i, e_i]$ in I' , the interval $[s_i, e_i]$ contains at least one of the times in X . Then no pairwise-disjoint subset of I' contains more than k jobs.

Proof (long form).

- Consider any such $I' = \{[s_1, e_1], [s_2, e_2], \dots, [s_n, e_n]\}$ and $X = \{x_1, x_2, \dots, x_k\}$.
 - Assume for contradiction that some pairwise-disjoint subset S of I' has more than k jobs.
 - By the assumption on X in the theorem, each job in S contains at least one point in X .
 - There are more than k jobs in S , but only k points in X .
 - So (by the previous two steps) some point x_i in X is in at least two jobs in S .
 - So there are two jobs in S that are not disjoint (as they each contain x_i).
 - This contradicts that S is pairwise-disjoint.
- By Block 2, no pairwise-disjoint subset of I' contains more than k jobs. □

- Is the following conjecture true or false? Don't justify your answer.

Conjecture 1. Consider any instance $I' = \{[s_1, e_1], [s_2, e_2], \dots, [s_n, e_n]\}$ of Activity Selection. Let $e'_1, e'_2, \dots, e'_k$ be the end times of the jobs that are chosen by the algorithm. Then, for every job $[s_i, e_i]$ in I' , the interval $[s_i, e_i]$ contains at least one of the end times $e'_1, e'_2, \dots, e'_k$.

(Note that if the theorem and conjecture are both true, together they imply that no pairwise-disjoint subset of I' is larger than the one returned by the greedy algorithm.)

Problem 1 answer

- (a) The set D contains the following jobs:

★...★

- (b) The algorithm returns the following set of jobs:

{★...★}

- (c) The variable ℓ takes on the following values (in increasing order):

★...★

- (d) Is the proof correct? If not, what's the first step to make an assertion that doesn't necessarily hold?

★...★

- (e) Is the conjecture true or false? (Do not justify your answer.)

★...★

Problem 1 collaboration and citations

I collaborated on this problem with the following people:

• ★ ⋯ ★

My answers use ideas from the following sources (other than the lecture notes and textbooks):

• ★ ⋯ ★

Problem 2. (Section 5.2) For each $d \in \{6, 7, 8, 9, 11\}$, consider Making Change with denominations (coin values) $\{1, 5, d\}$. For one of these five values, it is true that, for all integers $N \geq 0$, the greedy algorithm (using available denominations $\{1, 5, d\}$) makes change for N using a minimum possible number of coins. For the other four values, it is not true.

(a) Fill out the table in the template showing, for each value of d , the smallest N for which the greedy algorithm uses too many coins. (For the d for which the greedy algorithm always uses a minimum number of coins, put ∞ .)

(b) For the value of d for which the greedy algorithm can't use too many coins, give a proof of the following lemma for that particular value of d :

Lemma 1. For any $N \geq d$, some optimal way of making change for N (using coin values in $\{1, 5, d\}$) contains at least one d .

Note the emphasis on “some”.

Problem 2 answer

(a) Complete the following table to show, for each d , the minimum input N for which greedy with denominations $\{1, 5, d\}$ uses too many coins. For the d for which greedy works, put ∞ .

d	N
6	★...★
7	★...★
8	★...★
9	★...★
11	★...★

(continued)

Problem 2(b) answer (b) The d for which the greedy algorithm works is $d = \star \cdots \star$.

Here is the requested lemma and proof (for that d):

Lemma 2. *For any $N \geq \star \cdots \star$, some optimal way of making change for N (using coin values in $\{1, 5, \star \cdots \star\}$) contains at least one $\star \cdots \star$*

Proof (long form).

1. Consider any such N .
2. Let X^* be an optimal way of making change for N using coins of those values.
- 3.1. Case 1. First consider the case that X^* contains a $\star \cdots \star$.
- 3.2. Then some optimal way of making change for N (namely X^*) contains a $\star \cdots \star$.
- 4.1. Case 2. Next consider the case that $\star \cdots \star$
- 4.2. $\star \cdots \star$
- 4.3. So some optimal way of making change for N contains a $\star \cdots \star$.
- 5.1. Case 3. Otherwise $\star \cdots \star$.
- 5.2. $\star \cdots \star$
- 5.3. $\star \cdots \star$
- 5.4. $\vdots$
6. By Cases 1–3, some optimal way of making change for N contains a $\star \cdots \star$. □

Problem 2 collaboration and citations

I collaborated on this problem with the following people:

• ★ ⋯ ★

My answers use ideas from the following sources (other than the lecture notes and textbooks):

• ★ ⋯ ★

Problem 3. (see Exercise 5.14)(Greedy step for Puncturing Intervals)

The Puncturing Intervals problem is defined as follows:

input: Set $I = \{J_1, J_2, \dots, J_n\}$ of intervals, where $J_i = [s_i, e_i]$. We assume $e_1 \leq e_2 \leq \dots \leq e_n$.

output: A minimum-size finite set $P = \{p_1, p_2, \dots, p_k\}$ of points such that $J_i \cap P \neq \emptyset$ for each $J_i \in I$.

That is, you want to choose a minimum-size set of points such that every given interval J_i contains at least one of those points. For example, for the set I of intervals in Problem 1, the set $P = \{1, 5, 9, 14\}$ of points does have the property that it contains at least one point in each interval in I . However, that set P is not a correct solution, because it doesn't have minimum possible size.

- (a) Give a correct solution P for the instance defined by that set I of intervals.
- (b) Describe a rule that, given an instance I with $e_1 \leq e_2 \leq \dots \leq e_n$, chooses a single point p_1 such that there is guaranteed to exist a correct solution P (for the instance I) that contains p_1 . Describe your rule precisely in a single sentence. Your rule should be implementable as an algorithm that takes I as input and produces the desired point p_1 in *constant* time. Do not explain your reasoning for why you think your rule works.

Problem 3 answer

- (a) Here is a correct solution for the set I of intervals defined in Problem 1:

$$P = \{\star \cdots \star\}$$

- (b) Here is a rule that, given any instance I , produces a point p_1 that is guaranteed to be in some correct solution:

$$\star \cdots \star$$

(continued)

Problem 3 collaboration and citations

I collaborated on this problem with the following people:

• ★ ⋯ ★

My answers use ideas from the following sources (other than the lecture notes and textbooks):

• ★ ⋯ ★