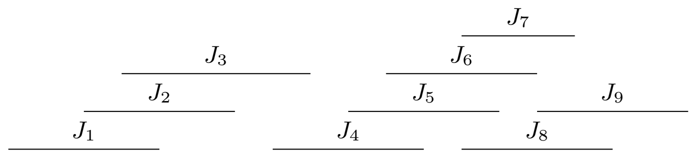
Problem 1. Consider executing the Activity Selection algorithm on the following instance of the Activity Selection problem:

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



(a) When rselect(I) executes, what jobs are in the set D that it computes in Line 3, before recursing?

(b) What set of jobs does rselect(I) output?

(c) Consider running the linear-time iterative implementation, iselect on input I. What values does the variable ℓ take on during the execution? Give them in increasing order.

(d) 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 $I' = \{[s_1,e_1],[s_2,e_2],\ldots,[s_n,e_n]\}$ of Activity Selection. Let $X = \{x_1,x_2,\ldots,x_k\}$ 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).

1. Consider any such $I' = \{[s_1,e_1],[s_2,e_2],\ldots,[s_n,e_n]\}$ and $X = \{x_1,x_2,\ldots,x_k\}$.

2.1. Assume for contradiction that some pairwise-disjoint subset S of I' has more than k jobs.

2.2. By the assumption on X in the theorem, each job in S contains at least one point in X.

2.3. There are more than k jobs in S, but only k points in X.

2.4. So (by the previous two steps) some point x_i in X is in at least two jobs in S.

2.5. So there are two jobs in S that are not disjoint (as they each contain x_i).

2.6. This contradicts that S is pairwise-disjoint.

3. By Block 2, no pairwise-disjoint subset of I' contains more than k jobs. $\square$

(e) Is the following conjecture true or false? Don't justify your answer.

Conjecture 1. *Consider any instance $I' = \{[s_1,e_1],[s_2,e_2],\ldots,[s_n,e_n]\}$ of Activity Selection. Let $e'_1,e'_2,\ldots,e'_k$ 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 $e'_1,e'_2,\ldots,e'_k$.*

(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:

$$\{J_4[7,11], J_5[9,13], J_6[10,14], J_7[12,15], J_8[12,16], J_9[14,18]\}$$

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

$$\{J_1[0,4], J_4[7,11], J_7[12,15]\}$$

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

$$\ell = -\infty, \ell = 4, \ell = 11, \ell = 15$$

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

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

Problem 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 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	10
7	10
8	10
9	∞
11	15

(continued)

Problem 2(b) answer (b) The d for which the greedy algorithm works is $d = 9$.

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

Lemma 2. *For any $N \geq 9$, some optimal way of making change for N (using coin values in $\{1, 5, 9\}$) contains at least one 9*

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 9 .

3.2. Then some optimal way of making change for N (namely X^*) contains a 9 .

4. If X* is an optimal solution for $N \geq 9$ and it doesn't contain a 9, it must only contain nickels and no more than 3 pennies, or else you could combine 4 pennies and a nickel to form a 9, and the X* wouldn't be optimal.

5.1. Case 2. Next consider the case that X* contains 2m nickels (an even number of nickels).

5.2. 2m nickels = 2m * 5 = 10m = 9m + m = m(9 + 1).

5.3. For even nickels, you can replace every 2 nickels with a 9 and a penny (1). Because in either case you use 2 coins for every 10 value, both are an optimal solution.

5.4. So some optimal way of making change for N contains a 9 .

6.1. Case 3. Otherwise X* contains 2m + 1 (an odd number of nickels).

6.2. 2m + 1 nickels = (2m + 1) * 5 = 10m + 5 = 9m + m + 5 = m(9 + 1) + 5.

6.3. Because N $\geq$ 9 with X* which doesn't contain a 9, to have odd nickels, you would have to have 3 or more nickels in X*.

6.4. For odd nickels, you can use a 9 and a penny m times to replace the 2m nickels, and have 1 nickel left over. So you would be replacing 3 nickels with a 9, a penny, and leaving the last nickel (both have value 15), and replacing the rest of the now even pair of nickels each with a 9 and a penny.

6.5. Because in either case you would use 2m + 1 coins, both the nickels, and the 9 and penny to replace every 10 and leaving a nickel left over are an optimal solution.

7. By Cases 1–3, some optimal way of making change for N contains a 9 . $\square$

Problem 3. The Puncturing Intervals problem is defined as follows:

input: Set $I = \{J_1, J_2, \ldots, J_n\}$ of intervals, where $J_i = [s_i, e_i]$. We assume $e_1 \le e_2 \le \cdots \le e_n$.
output: A minimum-size finite set $P = \{p_1, p_2, \ldots, p_k\}$ 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 $e_1 \le e_2 \le \cdots \le e_n$, 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:

$$P = \{4, 11, 14\}$$

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

Choose p_1 to be the end time of the earliest ending interval, and since they are sorted by end time, you would choose the endpoint of J_1.