

2.3 Art Gallery

Andre Monteiro

Term 3, 2025

Problem Summary

You are given n chores to do over the next n days. Each chore takes 1 day, and only one chore can be done per day.

You get paid $p_i > 0$ cents for chore i if you finish it on or before its due date $t_i \geq 1$. You get nothing if you do it after the due date.

Example:

| Chore i | X | Y | Z |
|-----------|----|----|----|
| p_i | 20 | 50 | 30 |
| t_i | 2 | 2 | 1 |

The best way to schedule the chores is to do Z on day 1 and Y on day 2, for a total of 80 cents.

Assume a greedy algorithm is used to find the schedule that gives the highest total profit.

Question a)

A greedy heuristic is one that:

- (a) Makes a locally optimized choice
- (b) Does not look ahead or explore alternatives
- (c) Assumes this will lead to a globally optimized solution

As such, the heuristic used in this problem is first sorting the array by highest cost, and then placing each item at the latest possible date for its execution. This creates a locally optimized result in choosing the highest cost at each iteration. Alongside choosing the highest cost, we also schedule it at the latest possible time, ensuring that it's not possible to further optimize this result.

Proof of validity

Suppose a schedule is not optimal, meaning it does not produce the maximum possible profit. Then there must exist at least one step where the algorithm scheduled a less profitable chore, even though a more profitable one was still available and could have been placed within its deadline. This contradicts the greedy heuristic rule, which always selects the highest-paying available chore and schedules it at the latest possible day before its deadline. Therefore, if no such mistake is made, the greedy algorithm must have produced the optimal result.

Example

| Chore i | A | B | C | D |
|------------------|----|----|----|----|
| p_i (profit) | 20 | 10 | 70 | 60 |
| t_i (deadline) | 1 | 3 | 1 | 2 |

Sorted Chores by Profit (Descending)

The greedy algorithm begins by sorting the chores in descending order of profit:

| Chore i | C | D | A | B |
|-----------|----|----|----|----|
| p_i | 70 | 60 | 20 | 10 |
| t_i | 1 | 2 | 1 | 3 |

We maintain a schedule of days 1 to 4. For each chore in the sorted list, we place it on the latest available day $\leq t_i$.

Insert Chore C

$$i = C, \quad p_i = 70, \quad t_i = 1$$

Latest available day ≤ 1 is day 1.

| | | | | |
|-------|--------|---|---|---|
| Day | 1 | 2 | 3 | 4 |
| Chore | C (70) | | | |

Insert Chore D

$$i = D, \quad p_i = 60, \quad t_i = 2$$

Latest available day ≤ 2 is day 2.

| | | | | |
|-------|--------|--------|---|---|
| Day | 1 | 2 | 3 | 4 |
| Chore | C (70) | D (60) | | |

Insert Chore A

$$i = A, \quad p_i = 20, \quad t_i = 1$$

Day 1 is already taken. No available day ≤ 1 .

Chore A is skipped.

Insert Chore B

$$i = B, \quad p_i = 10, \quad t_i = 3$$

Latest available day ≤ 3 is day 3.

| | | | | |
|-------|--------|--------|--------|---|
| Day | 1 | 2 | 3 | 4 |
| Chore | C (70) | D (60) | B (10) | . |

Final Schedule and Total Profit

- Day 1: Chore C (70 cents)
- Day 2: Chore D (60 cents)
- Day 3: Chore B (10 cents)
- Day 4: Unused

$$\text{Total Profit} = 70 + 60 + 10 = \boxed{140 \text{ cents}}$$