

Lab 10

Greedy algorithms

COLLABORATION LEVEL 0 (NO RESTRICTIONS). OPEN NOTES.

(wrap up) From class: **Greedy counterexample:** A pharmacist has W pills and n empty bottles. Bottle i can hold p_i pills and has an associated cost c_i . Given W, p_1, p_2, \dots, p_n and c_1, c_2, \dots, c_n , you want to store all pills using a set of bottles in such a way that the total cost of the bottles is minimized. Note: If you use a bottle you have to pay for its cost no matter if you fill it to capacity or not. Someone proposes the following greedy strategy to solve this problem: Pick the bottle with the smallest cost-per-pill, and recurse on the remaining pills with the remaining bottles. Show that this greedy strategy is not correct by giving a (small) counterexample.

(wrap up) From class: **A different pharmacist problem:** A pharmacist has W pills and n empty bottles, where all bottles cost the same and bottle i can hold p_i pills. The goal is to find the minimum cost for storing the W pills using the bottles.

Sketch the algorithm, analyze its running time and justify that it is correct, that is, it is guaranteed to always reach an optimal solution.

To prove that a greedy algorithm is correct it is sufficient to prove that there exists an optimal solution that includes the first greedy choice. Once you prove this claim about the first greedy choice, you can apply it recursively to all subsequent choices, which will imply that there exists an optimal solution consisting entirely of all greedy choices.

1. **Art gallery guarding:** In the *art gallery guarding* problem we are given a line L that represents a long hallway in an art gallery. We are also given a set $X = \{x_0, x_1, x_2, \dots, x_{n-1}\}$ of real numbers that specify the positions of paintings in this hallway; assume that each painting is a point. Suppose that a single guard can protect all the paintings within a distance at most 1 of his or her position, on both sides. You can assume that the positions of the paintings are sorted.

Design an algorithm for finding a placement of guards that uses the minimum number of guards to guard all the paintings in X . Argue why your algorithm is correct and analyze its running time.

We expect: The algorithm as pseudocode, a brief explanation, analysis and justification of correctness i.e. why does it always assure the minimum nb of guards? Remember that to show that a greedy algorithm is correct it is sufficient to show that there exists an optimal solution that includes the first greedy choice.)

2. **Thanksgiving Turkeys:**¹ On Thanksgiving day, you arrive on an island with n turkeys. You've already had Thanksgiving dinner so you don't want to eat the turkeys (and maybe you

¹Credit: Problem reproduced from Stanford, cs161

prefer tofurkey anyway), but you do want to wish them all a Happy Thanksgiving. However, the turkeys each have very different sleep schedules. Turkey i is awake only in a single closed interval $[a_i, b_i]$. Your plan is to stand in the center of the island and say loudly "Happy Thanksgiving!" at certain times t_1, \dots, t_m . Any turkey who is awake at one of the times t_j will hear the message. It's okay if a turkey hears the message more than once, but you want to be sure that every turkey hears the message at least once.

Design a greedy algorithm which takes as input the list of intervals $[a_i, b_i]$ and outputs a list of times t_1, \dots, t_m so that m is as small as possible and so that every turkey hears the message at least once. Your algorithm should run in time $O(n \log(n))$.

[We are expecting: Pseudocode and an English description of the main idea of your algorithm, as well as a short justification of the running time.]

Hint: This is very similar to a problem discussed in class. Can you see how?

Additional/optional problems

1. **Matching points on a line:**² You are given two arrays of n points in one dimension: red points r_1, r_2, \dots, r_n and blue points b_1, b_2, \dots, b_n . You may assume that all red points are distinct and all blue points are distinct. We want to pair up red and blue points, so that each red point is associated uniquely with a blue point and vice versa. Given a pairing, we assign it a score which is the sum of distances between each pair of matched points.

Find the pairing (matching) of minimum score. Hint: Aim for an $O(n \lg n)$ algorithm. Draw examples for small values of n to get intuition.

Example: Consider the input where the red points are $r_1 = 8, r_2 = 1$ and the blue points are $b_1 = 3$ and $b_2 = 9$. There are two possible matchings:

- match r_1 to b_1 and r_2 to b_2 : the score of this matching is $|r_1 - b_1| + |r_2 - b_2| = |8 - 3| + |1 - 9| = 13$
- match r_1 to b_2 and r_2 to b_1 : the score of this matching is $|r_1 - b_2| + |r_2 - b_1| = |8 - 9| + |1 - 3| = 3$

The algorithm should return the cost of the optimal matching, which is 3.

Hint: Think greedily, and argue with an exchange argument.

²Credit: reproduced from Stanford University, cs161