

Homework 4

Due Date: October 14, 2019 at 11:59PM on Gradescope.

Instructor: Jeremiah Blocki

Homework Guideline Reminders

- Assignments must be typed. Submit one pdf file to Gradescope by 11:59PM, or else late penalties will apply. The pdf file can include hand-drawn images of figures.
- Each question needs to start with the resources and collaborator (RC) statement. You will not be penalized for using resources or having collaborators if your answers are expressed in your own words. If you consulted no resources outside of course material or had no collaborators, you must state so. A question without a complete RC statement will not be graded.

Question 1: Greedy Permutations (25 points)

Suppose you are given two coordinates in n -dimensional space, $\mathbf{p} = (p_1, p_2, \dots, p_n)$, and $\mathbf{q} = (q_1, q_2, \dots, q_n)$. The *golden* distance d between \mathbf{p} , and \mathbf{q} is defined as $d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n (p_i - q_i)^4$. Let \mathbf{p}_σ be formed by permuting the elements of \mathbf{p} , and let \mathbf{q}_π be formed by permuting the elements of \mathbf{q} .

For example, if $\mathbf{p} = (1, 2, 3)$ and $\mathbf{q} = (1, 3, 5)$, then one possible way to form the permutations could be $\mathbf{p}_\sigma = (2, 1, 3)$ and $\mathbf{q}_\pi = (5, 3, 1)$. The golden distance, $d(\mathbf{p}_\sigma, \mathbf{q}_\pi)$, of these two points would be $(2 - 5)^4 + (1 - 3)^4 + (3 - 1)^4 = 113$.

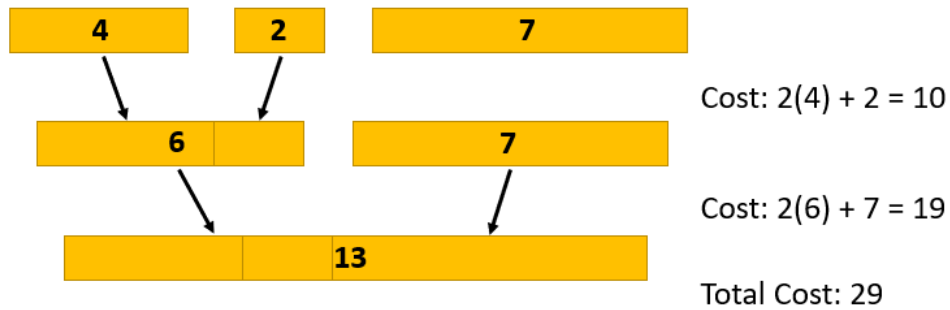
- (a) Design a greedy algorithm that chooses the permutations for \mathbf{p}_σ and \mathbf{q}_π that maximizes their golden distance, $d(\mathbf{p}_\sigma, \mathbf{q}_\pi)$. Analyze the time complexity of your algorithm.
- (b) Prove the correctness of your algorithm. (**Hint:** You may use the following fact without proof $(d - a)^4 + (c - b)^4 \geq (b - a)^4 + (d - c)^4$ for any numbers a, b, c, d such that $a \leq c$ and $d \geq b$.)

Question 2: Rod Welding (25 points)

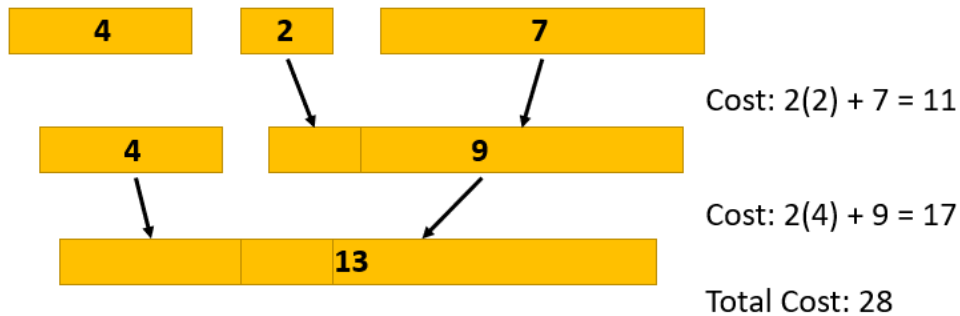
George needs a long metal rod for his latest construction project, but all of the rods have been cut into pieces by CS 381 students. He decides to reconstruct a rod out of all of the n pieces. He is given an array, R , of length n , where $R[i]$ denotes the length of the i th rod piece. George needs to join together two pieces at a time until he finishes with the final connected rod. The rod pieces must be in the same order in the final rod as in the given array (i.e. you can't swap rods around), but you can join two adjacent pieces in any order. The cost of joining two adjacent rods, where the left rod has length l and the right rod has

length r , is $2l + r$, as longer rods are heavier to move, and George is right-handed. The resulting rod has length $l + r$. George wants to expend the least amount of effort putting the rod together, so he wants to find the order in which he should join rods with the least cost.

For instance, suppose we have rod pieces of length 4, 2, and 7 which we want to join into one rod of length 13. We could start by joining the first two rod pieces and then join the result with the third rod piece as shown below.



This results in a total cost of 29. We could instead choose to initially join the second and third rods and then join that with the first rod to get a reduced cost of 28 as shown below.



Design an algorithm to determine the least cost that it would take to join all n rod pieces into the final rod. Prove the correctness of your algorithm and analyze its time and space complexity.

Question 3: Study Schedule (25 points)

Suppose that you are preparing for n incoming midterm exams numbered as $1, 2, \dots, n$. Each exam will be graded on a scale from 0 to 100. You have decided to spend a total of H hours to study for the exams and you want to divide up the time for studying the exams. For simplicity, assume that H is a positive integer, and you will spend a non-negative integer number of hours studying for each exam. To find out how to best divide up your time, you roughly estimate that if you spend h hours on studying for the exam i you will get a grade

of $f_i(h)$. You may assume that each function f_i is non-decreasing (i.e. $f_i(h) \geq f_i(h')$ for every $h \geq h'$). In other words, the harder you study for the exam, the better grade you will get.

Given the set of estimated score functions $\{f_1, f_2, \dots, f_n\}$, devise an efficient (polynomial in n and pseudopolynomial in H) algorithm to figure out how many hours (in integer values only) to spend on studying for each exam so that your total sum of all the grades is maximized. Prove the correctness of your algorithm and analyze its time and space complexity. Note that to get full credits your algorithm needs to output both the optimal total grade and the corresponding distribution of time.

Question 4: Book Rush (25 points)

Prof. Blocki plans to help Theory students by donating some of his books. He decides to put those books in the LWSN commons, and arranges them in a row. You notice the books and immediately jump towards them with your backpack. You are very excited to see all the books, however, unfortunately you also see Jane standing close to you with her backpack, equally excited to grab those books. There is no other person around the books area, so you both come up with few rules to pick the books. Before we discuss the rules, here are few things to keep in mind:

- There are n number of books and you can assume that n is even.
- Each book i has a value v_i and a space s_i (integer) associated with it.
- The total space of your backpack is S (positive integer). Unfortunately for you, Jane brought a “magical” Austen backpack, which doesn’t have any space constraint.
- Jane is also currently taking 381, so assume that she will also play optimally.

Here are the rules for picking the books:

1. The objective for both of you is to maximize the value of books you pick, given the ones you choose fit in your backpack.
2. Both of you will be alternating turns to choose to pick a book.
3. By some luck, you get to start first.
4. A book can only be picked from either ends of the row.
5. In the situation where you lack sufficient space to fit the book at either end into your backpack, the remaining books are given to Jane.

For this problem,

- (a) Describe an efficient DP algorithm which determines the maximum total value of the books that you can achieve with optimal play. Clearly state and justify your recurrence (a full proof of correctness is not required). Analyze the time and space complexity of

your algorithm (You may assume that $S \leq n^2$).

Hint 1: Notice that this is a zero-sum game since Jane gets to keep every book that you don't take.

Hint 2: It may be helpful to review the "coins in a line" problem from PSO. Can you find a way to extend the core ideas by adding more sub-problems? How will you incorporate information about remaining space?

- (b) **Bonus (5 points):** Consider a modification of the problem such that the players are allowed to skip their turn as long as we never have two skips in a row (i.e., if you skip a turn then Jane must select a book and vice versa). If you must select a book but lack space then all remaining books are given to Jane. How would you modify your algorithm to accommodate this? Analyze the time and space complexity of your new modified algorithm.