

- The assignment is due at Gradescope on Thursday, October 6 at 11:59pm. Late assignments will not be accepted. Submit early and often.
- You are permitted to study with friends and discuss the problems; however, *you must write up your own solutions, in your own words*. Do not submit anything you cannot explain. If you do collaborate with any of the other students on any problem, please do list all your collaborators in your submission for each problem.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly prohibited.
- We require that all homework submissions are prepared in Latex. If you need to draw any diagrams, however, you may draw them with your hand. Please use *a new page to begin each answer*.

**PROBLEM 1** *Flip-flop*

Consider the recurrence  $T(n) = 2T(n/2) + f(n)$  in which

$$f(n) = \begin{cases} n^3 & \text{if } \lceil \log(n) \rceil \text{ is even} \\ n^2 & \text{otherwise} \end{cases}$$

Show that  $f(n) = \Omega(n^{\log_b(a)+\epsilon})$ . Explain why the third case of the Master's theorem stated above does not apply. Prove a  $\Theta$ -bound for the recurrence.

PROBLEM 2 *Why 5 in Median?*

Recall the deterministic selection algorithm:

SELECT( $A[1, \dots, n], i$ )

- 1 Base case if  $|A| < 5$ .
- 2  $p \leftarrow \text{MEDIANOFMEDIANS}(A)$
- 3  $A_\ell, A_r, i_p \leftarrow \text{PARTITION}(A, p)$
- 4 **if**  $i_p = i$  **return**  $A[i_p]$
- 5 **elseif**  $i_p < i$  **return** SELECT( $A_r, i - i_p$ )
- 6 **else** **return** SELECT( $A_\ell, i_p - i$ )

MEDIANOFMEDIANS( $A[1 \dots n]$ )

- 1 Divide  $A$  into lists of 5 elements. If only one element, return it.
- 2 Compute the median of each small list, store these medians in a new list  $B$
- 3  $p \leftarrow \text{SELECT}(B, \lceil n/10 \rceil)$
- 4 **return**  $p$

1. Suppose Line 1 of MEDIANOFMEDIANS divides  $A$  into lists of 3 elements each instead of 5 elements and line 3 is modified to pick the  $\lceil n/6 \rceil^{\text{th}}$  element. State an upper and lower bound on the size of  $A_\ell$ . Be as precise as you can.
2. Show a lower-bound for the running time of SELECT under the 3-element version of MEDIANOFMEDIANS.

### PROBLEM 3 Planet Laser

The NASA Near Earth Object Program lists potential future Earth impact events that the JPL Sentry System has detected based on currently available observations. Sentry is a highly automated collision monitoring system that continually scans the most current asteroid catalog for possibilities of future impact with Earth over the next 100 years.

This system allows us to predict that  $i$  years from now, there will be  $x_i$  tons of asteroid material that has near-Earth trajectories. In the mean time, we can build a space laser that can blast asteroids. However, each laser blast will require *exajoules* of energy, and so there will need to be a recharge period on the order of *years* between each use of the laser. The longer the recharge period, the stronger the laser blast; e.g. after  $j$  years of charging, the laser will have enough power to obliterate  $d_j$  tons of asteroid material. This problem explores the best way to use such a laser.

The input to the algorithm consists of the vectors  $(x_1, \dots, x_n)$  and  $(d_1, \dots, d_n)$  representing the incoming asteroid material in years 1 to  $n$ , and the power of the laser  $d_i$  if it charges for  $i$  years. The output consists of the optimal schedule for firing the laser which obliterates the most material.

**Example** Suppose  $(x_1, x_2, x_3, x_4) = (1, 10, 10, 1)$  and  $(d_1, d_2, d_3, d_4) = (1, 2, 4, 8)$ . The best solution is to fire the laser at times 3, 4 in order to blast 5 tons of asteroids.

- (a) Construct an instance of the problem on which the following “greedy” algorithm returns the wrong answer:

$\text{BADLASER}((x_1, \dots, x_n), (d_1, \dots, d_n))$

- 1 Compute the smallest  $j$  such that  $d_j \geq x_n$ . Set  $j = n$  if no such  $j$  exists.
- 2 Shoot the laser at time  $n$ .
- 3 if  $n > j$  then  $\text{BADLASER}((x_1, \dots, x_{n-j}), (d_1, \dots, d_{n-j}))$ .

Intuitively, the algorithm figures out how many years ( $j$ ) are needed to blast all the material in the last time slot. It shoots during that last time slot, and then accounts for the  $j$  years required to recharge for that last slot, and recursively considers the best solution for the smaller problem of size  $n - j$ .

- (b) Given an array holding  $x_i$  and  $d_i$ , devise an algorithm that blasts the most asteroid material. Analyze the running time of your solution.

**PROBLEM 4** *Price Run*

Given a list of closing stock ticker prices  $p_1, p_2, \dots, p_n$ , devise an  $O(n^2)$  algorithm that finds the longest (not necessarily consecutive) streak of prices that increase or stays the same. For example, given the prices 2, 5, 2, 6, 3, 3, 6, 7, 4, 5, there is the streak 2, 5, 6, 6, 7 of prices that increase or stay the same, but an even longer streak is 2, 2, 3, 3, 4, 5.

(Challenge: by using both dynamic programming and binary search, you can solve this problem in  $O(n \log n)$  time.)

PROBLEM 5 *Tug of War*

We want to play *roughly fair* tug of war in cs4800. You are given an array that holds the weights of  $n$  people in the class  $W = (w_1, w_2, \dots, w_n)$ . Your goal is to divide the  $n$  people into two teams such that the total weight of the two teams is equal or as close as possible to equal. Describe such an algorithm and give its running time. The total number of people on each team should differ by at most 1. Assume that  $M$  is the maximum weight of a person, i.e.,  $\forall i, w_i \leq M$ . The running time should be  $O(n^3M)$ . The output should be the list of people on each team and the difference in weight between the teams.

PROBLEM 6 *Programming Tug of War*

In this part, you will implement your algorithm for Tug of War and test it on real data. Register and take on the challenge at <https://www.hackerrank.com/contests/cs4800f16>