

### Homework 3

Out: Monday, February 19, 2018

Due: 8pm, Monday, March 5, 2018

*For all algorithms you design, you should always describe them clearly in English, give pseudocode, prove correctness and give the best upper bound that you can for the running time. If you give a dynamic programming algorithm, you should clearly define the subproblems, give the recurrence, analyze time and space requirements and explain how to fill in the dynamic programming table.*

**You may not use any external resources for this homework: doing so will have a negative impact on your performance in the midterm exam and in interviews. Further, you should solve problems 1 and 2 entirely on your own. I suggest you work on recommended exercises 1 and 2 first.**

*Collaboration is limited to discussion of ideas only. You should write up the solutions entirely on your own. You should list your collaborators on your write-up. If you do not type your solutions, be sure that your hand-writing is legible, your scan is high-quality and your name is clearly written on your homework.*

1. (12 points) Consider the problem of finding the longest monotonically increasing subsequence of a sequence of  $n$  numbers  $a_1, \dots, a_n$ .

A subsequence is any subset of these numbers taken in order, of the form  $a_{i_1}, a_{i_2}, \dots, a_{i_k}$  where  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ . An increasing subsequence is one in which the numbers are getting strictly larger. For example, the longest increasing subsequence of 5, 2, 8, 6, 3, 6, 7 is 2, 3, 6, 7 and its length is 4.

Give a dynamic programming algorithm that solves this problem in time  $O(n^2)$ .

2. (22 points) There are  $n$  libraries  $L_1, L_2, \dots, L_n$ . We want to store copies of a book in some of them. Storing a copy at  $L_i$  incurs a purchase cost  $c_i$  (assume integer  $c_i > 0$ ). A copy of the book is stored at  $L_n$ . If a user requests the book from  $L_i$  and  $L_i$  does not have it, then  $L_{i+1}, L_{i+2}, \dots$  are searched sequentially until a copy of the book is found at  $L_j$  for some  $j > i$ . This results in a user delay of  $j - i$ . (Note that, in this case, no library  $L_k$  with an index  $k$  smaller than  $i$  is searched; also, if the user finds the book at  $L_i$ , then the user delay is 0.)

We define the **total cost** as the sum of the purchase costs and the *user delays associated with all  $n$  servers*. For example, if there are 4 libraries, and copies of the book are stored at  $L_1$  and  $L_4$ , the total cost is  $c_1 + c_4 + 2 + 1$ .

Give an efficient algorithm that determines

- (a) at which libraries to place copies of the books so that the **total cost** is minimized;
- (b) the minimum total cost.

3. (26 points) There are  $n$  tasks to complete. Each task first needs to be *preprocessed* on a supercomputer and then finished on one of  $n$  processors. Task  $i$  requires  $p_i$  seconds of preprocessing on the supercomputer followed by  $f_i$  seconds on some processor to complete.

Note that tasks are fed to the supercomputer sequentially but can be executed in parallel on different processors once they have been preprocessed. The process *terminates* when all tasks have been completed. The *duration* of the process is the total time until termination.

Give an efficient algorithm that determines an ordering of the tasks for the supercomputer that minimizes the duration of the process, as well as the minimum duration.

4. (30 points) Given integers  $a_1, \dots, a_n$ , we want to determine whether it is possible to partition  $\{1, \dots, n\}$  into three disjoint subsets  $I, J, K$  such that

$$\sum_{i \in I} a_i = \sum_{j \in J} a_j = \sum_{k \in K} a_k = \frac{1}{3} \sum_{i=1}^n a_i$$

For example, for input  $\{1, 2, 3, 4, 4, 5, 8\}$  the answer is **yes**, because there is the partition  $(1, 8), (4, 5), (2, 3, 4)$ . On the other hand, for input  $\{2, 2, 3, 5\}$  the answer is **no**.

Design and analyze a dynamic programming algorithm for this problem that runs in time polynomial in  $n$  and  $\sum_{i=1}^n a_i$ .

5. (40 points) You are given some data to analyze. You can spend  $D$  dollars to perform the analysis. You have organized the process of analyzing the data so that it consists of  $n$  tasks that have to be performed sequentially by using dedicated hardware: you will use a processor  $P_i$  to perform task  $i$ , for every  $i$ . Each processor is relatively cheap but may fail to complete its task with some probability, independently of the other processors. Specifically,  $P_i$  costs  $c_i$  dollars and succeeds to complete its task with probability  $s_i$ , while it fails with probability  $1 - s_i$ .

- (a) (3 points) What is the probability that the process of analyzing the data will be completed successfully?
- (b) Note that you can improve this success probability by using  $p_i$  identical processors  $P_i$  for task  $i$  instead of just one.
- (8 points) What is the probability that task  $i$  will be completed successfully now?
  - (4 points) What is the probability that the process of analyzing the data will be completed successfully?
  - (25 points) Given  $s_1, \dots, s_n, c_1, \dots, c_n$  and  $D$ , compute  $p_1, \dots, p_n$  such that the success probability of the entire process is maximized while you do not spend more than  $D$  dollars.

**Recommended exercises: do NOT return, they will not be graded**

1. Problem 16-1 in your textbook (pp. 446-447).
2. Consider an array  $A$  with  $n$  numbers, some of which may be negative. We wish to find indices  $i$  and  $j$  such that  $\sum_{k=i}^j A[k]$  is maximized. Design and analyze an algorithm that runs in  $O(n)$  time.

3. Show how to solve problem 1 in time  $O(n \log n)$ .

*Hint: The last element of a candidate subsequence of length  $i$  is at least as large as the last element of a candidate subsequence of length  $i-1$ . Maintain candidate subsequences by linking them through the input sequence.*

4. Trapping Rain Water — *This one is hard!*

Description at <https://leetcode.com/problems/trapping-rain-water/#/description>