

Homework 2

Out: Saturday, June 2, 2018

Due: 10pm, Tuesday, June 12, 2018

*Please keep your answers clear and concise. For all algorithms **you** suggest, you must prove correctness and give the best upper bound that you can for the running time. You should always describe your algorithm clearly in English **and** give pseudocode.*

You may not use any external resources or collaborate with your classmates for this homework: doing so will have a negative impact on your performance in the midterm exam and in interviews.

I suggest you first solve problems 1, 2, and parts a,b,c from recommended exercise 1. Then work on recommended exercises 2, 3, 1d and problems 3, 4, 5 (in this order).

If you do not type your solutions, make sure that your hand-writing is legible, your scan is high-quality and your name is clearly written on your homework.

1. (20 points) Consider a long country road with houses scattered very sparsely along it. (You may picture the road as a long line segment, with an eastern endpoint and a western endpoint.) You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations.

Give an efficient algorithm that achieves this goal, using as few base stations as possible.

2. (20 points) A server has n customers waiting to be served. The service time for customer i is t_i minutes. So if the customers are served in order of increasing i , the i -th customer spends $\sum_{j=1}^i t_j$ minutes waiting to be served.

Given $n, \{t_1, t_2, \dots, t_n\}$, design an efficient algorithm to compute the optimal order in which to process the customers so that the total waiting time below is minimized:

$$T = \sum_{i=1}^n (\text{time spent waiting by customer } i)$$

3. (20 points) You are going on a long trip. You start on the road at mile post 0. Along the way there are n hotels, at mile posts $a_1 < a_2 < \dots < a_n$, where each a_i is measured from the starting point. The only places you are allowed to stop are at these hotels, but you can choose which of the hotels you stop at. Your destination is the final hotel (at distance a_n) and you must stop there.

You'd like to travel 200 miles a day, but this may not be possible, depending on the spacing of the hotels. If you travel x miles during a day, the *penalty* for that day is $(200 - x)^2$. You want to plan your trip so as to minimize the total penalty—that is, the sum, over all travel days, of the daily penalties. Give an efficient algorithm that determines the total penalty of the optimal sequence of hotels at which to stop.

4. (20 points) Alice and Bob are playing a match to see who is the first to win n games, for some fixed $n > 0$. Suppose Alice and Bob are equally competent and they have already played $i + j$ games, of which Alice won i and Bob won j .

Give an efficient algorithm to compute the probability that Alice will go on to win the match. For example, if $i = n - 1$ and $j = n - 3$, then the probability that Alice will win the match is $7/8$, since she must win any of the next three games.

5. (20 points) You are given a rectangular piece of cloth with dimensions $m \times n$, where m, n are positive integers, and a list of k products that can be made using the cloth. For each product $i \in [1, k]$, you need a rectangle of cloth of dimensions $a_i \times b_i$ and its selling price is p_i , where a_i, b_i, p_i are all positive integers. You also have a machine that can cut any rectangular piece of cloth into two pieces, either horizontally or vertically.

Design an algorithm that computes the best return on the $m \times n$ piece of cloth, that is, a strategy for cutting the cloth so that the products made from the resulting pieces give the maximum sum of selling prices. (For every product, you may make as many copies as you wish, or none.)

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

1. Problem 16.1 from your textbook (pp. 446-447).
2. Exercise 15.4-5 from your textbook (p. 397).
3. Consider an array A with n numbers, some of which (but not all) may be negative. We wish to find indices i and j such that

$$\sum_{k=i}^j A[k]$$

is maximized. Give an efficient algorithm for this problem.

4. Given two strings $x = x_1x_2 \cdots x_m$ and $y = y_1y_2 \cdots y_n$, we wish to find the length of their longest common substring, that is, the largest k for which there are indices i and j such that

$$x_i x_{i+1} \cdots x_{i+k-1} = y_j y_{j+1} \cdots y_{j+k-1}.$$

Give an efficient algorithm for this problem.

5. Exercises 15.4-6 from your textbook (p. 397).
6. *Challenge problem:* Trapping Rain Water
Description at <https://leetcode.com/problems/trapping-rain-water/#/description>.