

CSE 321 Homework 4

Due date: 30 / 12 / 2018

1-) 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. You must stop at the final hotel (at distance a_n), which is your destination. You would ideally 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. Propose a dynamic programming algorithm that determines the optimal sequence of hotels at which to stop. Implement the algorithm with Python.

(Use $A = [190, 220, 410, 580, 640, 770, 950, 1100, 1350]$ as distances of hotels.)

2-) You are given a string of n characters $s[1 \dots n]$, which you believe to be a corrupted text document in which all punctuation has vanished (so that it looks something like “itwasthebestoftimes...”). You wish to reconstruct the document using a dictionary, which is available in the form of a Boolean function $\text{dict}(\cdot)$: for any string w ,

$$\text{dict}(w) = \begin{cases} \text{True} & \text{if } w \text{ is a valid word} \\ \text{False} & \text{otherwise} \end{cases}$$

Propose a dynamic programming algorithm that determines whether the string s can be reconstituted as a sequence of valid words. The running time should be at most $O(n^2)$, assuming calls to dict take unit time. Implement the algorithm with Python.

3-) Suppose you have k sorted arrays, each with n elements, and you want to combine them into a single sorted array that has $k \times n$ elements. Propose a divide-and-conquer algorithm for the problem and implement it with Python. K arrayi k*n boyutunda tek arrayde birleştirip merge sort yap.

4-) Alice wants to throw a party and is deciding whom to call. She has n people to choose from, and she has made up a list of which pairs of these people know each other. She wants to pick as many people as possible, subject to two constraints: at the party, each person should have at least five other people whom they know and five other people whom they don't know.

Propose a greedy algorithm that takes as input the list of n people and the list of pairs who know each other, and outputs the best choice of party invitees. Implement the algorithm with Python.

5-) For a set of variables x_1, \dots, x_n , you are given some *equality* constraints of the form " $x_i = x_j$ " and some *inequality* constraints, of the form " $x_i \neq x_j$ ". For instance, the constraints

$$x_1 = x_2; x_2 = x_3; x_3 = x_4; x_1 \neq x_4$$

cannot be satisfied. Propose a greedy algorithm that takes as input m constraints over n variables and decides whether the constraints can be satisfied. Implement the algorithm with Python.

Explain the algorithms clearly, and analyze time complexity of them in a single PDF file. Add your .py file /s and the PDF file in a folder which is named as your number. Don't forget to zip your homework before upload it to the Moodle.

Note:

- Submit your homework on Moodle.
- You can send an email to b.koca@gtu.edu.tr to ask any question about the homework
- Do your homework personally, group studies will be considered as cheating.