Homework 6

Strukture podataka i algoritmi I - I053

Homework instructions

The submission deadline is **January 10, 2024** at 9:00. You can type the tasks in LaTeX or write them by hand and scan them. Programming tasks should be submitted as .cpp files. All files need to be submitted to Teams. You can achieve a maximum of 100 points.

Task 1 (25 pts.)**.** Recall the edit distance[1] problem from the class. To transform the string s into the string t, you're allowed to use the following operations with cost 1:

- **add** any character anywhere into s;
- **remove** any character into s s;
- **replace** any character from s with any other character.

Inside the $(|s|+1) \times (|t|+1)$ DP table, we placed the edit distance of the prefix $s[0..i-1]$ i $t[0..j-1]$ into $DP[i][j]$ as follows:

$$DP[i][j] = \begin{cases} i, & \text{if } j = 0 \\ j, & \text{if } i = 0 \\ dp[i-1][j-1], & \text{if } s[i-1] = s[j-1] \\ 1 + \min\big(\underbrace{dp[i-1][j]}_{removal}, \underbrace{dp[i][j-1]}_{insertion}, \underbrace{dp[i-1][j-1]}_{replacement}\big), & \text{otherwise} \end{cases}$$

Fill the whole DP table for strings'ACAATCC' and 'AGCATGC'. List all the required operations needed to get the second string from the first string in the least amount of steps.

Task 2 (15 pts.)**.** Mirko is leaving Croatia for winter holidays. The holidays last n days, and each day Mirko can choose one of the following activities:

- Sledding, which brings him a_i happiness points.
- Ice skating, which brings him b_i happiness points.
- Studying for an exam, which brings him c_i happiness points.

As Mirko quickly gets bored with an activity, he does not want to choose the same activity two days in a row.

Create a program that will calculate the maximum number of happiness points that Mirko can achieve

The first line of input contains $1 \leq n \leq 10^5$: the number of holiday days.

The i-th of the following n lines contains three numbers $1 \leq a_i, b_i, c_i \leq 10^4$: happiness points for each activity on the i-th day.

In a single line you need to output a single number: the maximum possible number of happiness points.

[1] or Levenshtein distance, see https://en.wikipedia.org/wiki/Levenshtein_distance

INPUT	INPUT

3
10 40 70
20 50 80
30 60 90
	7
	6 7 8
	8 8 3
	2 5 2
	7 8 6
	4 6 8
	2 3 4
	7 5 1

OUTPUT	**OUTPUT**
210	46

Task 3 (15 pts.)**.** You’re given two strings, s and t. Implement an algorithm that computes the longest string that is both a substring of s and t. We say that string a is a substring of string b if a can be obtained by deleting some characters from b. For example ’hlo’ is a substring of ’hello’, but ’heol’ is not a substring of ’hello’.
The first line of input contains the string s built from lowercase english alphabet letters, such that $1 \leq |s| \leq 3000$.
The second line of input contains the string t built from lowercase english alphabet letters, such that $1 \leq |t| \leq 3000$.
In a single line you need to output the requested string.

INPUT	INPUT
xayaz xcyz	abracadabra avadakedavra
OUTPUT	**OUTPUT**
xyz	aaadara

Task 4 (15 pts.)**.** Ivica and Marica invented a new game. They wrote n numbers $a_1, \dots, a_n$ on pieces of paper. At the start of the game, they have a stack of k stones. In each move, the player chooses a number a_i from the slip of paper and removes a_i stones from the pile. A player who cannot make a move loses. Marica is the first to move, and then she alternates with Ivica. Create a program that calculates the winner if both players play optimally.
The first line of input contains numbers $1 \leq n \leq 100$ and $1 \leq k \leq 10^5$: the number of papers and stones.
The second line of input contains numbers $1 \leq a_1 < a_2 < \cdots < a_n \leq k$: numbers on the papers.
In a single line you need to output ’Marica’ if Marica wins the game, and ’Ivica’ otherwise.

INPUT	INPUT
2 4 2 3	2 5 2 3
OUTPUT	**OUTPUT**
Marica	Ivica

In the first test case, if Marica chooses 3 in the first move, Ivica will not be able to make any move because there will be one stone left on the pile. In the second test example, whatever Marica chooses in the first move, Ivica will be able to remove the rest of the stones from the pile in the second move.

Task 5 (15 pts.)**.** You are given an $n \times n$ grid to move on, but some fields are filled with rocks and you cannot move through them. Your task is to create a program that counts the number of different paths from the upper left to the lower right field. Assume that you are only allowed to move down or to the right.
The first line of input contains a number $1 \leq n \leq 1000$: the number of rows and columns of the grid.
Each of the n following lines contains a length n string containing ’.’ or ’*’, representing empty fields and taken fields respectively.
In a single line you need to output the required number of paths. If that number doesn’t fit in a 64-bit integer, output it modulo $10^9 + 7$[2].

[2] See `https://codeforces.com/blog/entry/72527` as an intro to modular arithmetic.

INPUT	**INPUT**
4 .... .*.. ...* *...	3 .** *** ***
OUTPUT	**OUTPUT**
3	0

Task 6 (15 pts.)**.** You're given a number n. In each step, you can subtract one of its digits from the number. For example, you can subtract 2 or 7 from the number 27 and get 25 or 20. Create a program that calculates the minimum number of steps to get to 0 starting with n.

The only input line contains the number $1 \leq n \leq 10^6$ from the problem statement.

In a single line you need to output the required number of steps.

INPUT	**INPUT**
27	35
OUTPUT	**OUTPUT**
5	7