ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2023

Course Project

Due: 23:59, June 4, 2023

1. This project requires you to solve four problems. For each problem, write a program with the input and output formats as specified in the problem.
2. All programs must terminate within 10 seconds and use at most 4GB of memory. Programs exceeding either limit will not be accepted. You can write your programs in C, C++, Java or Python, though certain languages may lead to more efficient implementations.
3. You may not use third-party libraries in your solution, *e.g.* numpy in Python is not allowed. If you are not sure whether a library is third-party or not, you can try to submit your program, and it will cause a Runtime Error or Compile Error if it is.
4. Your program submission is scored by an Online Judge platform containing a number of test cases. Your score depends on the percentage of test cases your program passes.
5. The Online Judge website is `http://10.19.127.47`. You can also find the problem list at `http://10.19.127.47/contest/1/problems`.
6. Each student's account name and initial password are the **STUDENT ID of your Gradescope account**. After logging into your account for the first time, please change your password. You can view your **STUDENT ID** at `https://www.gradescope.com/account/edit`. If you do not set it, your account name and initial password will be the prefix of your email in Gradescope.
7. To avoid abuse of the OJ, you are allowed at most one submission every five minutes. Violations of this rule may result in penalties to your score.
8. You must **NOT**
   (a) Read or use solutions written by others.

(b) Allow other people to read or use your solutions.

(c) Obtain test data by repeated submissions or other means.

Problem 1:

Given two positive integers N and K, find the lexicographically smallest permutation of the integers $1, \ldots, N$ which satisfies the following condition:

$$|A_i - i| \geq K \text{ for all } 1 \leq i \leq N.$$

Note: A sequence $v_1, ..., v_N$ is lexicographically smaller than another sequence $v'_1, ..., v'_N$ if there exists $1 \leq i \leq N$ such that $v_j = v'_j$ for all $1 \leq j < i$, and $v_i < v'_i$. For example, $(1, 2, 3)$ is lexicographically smaller than $(2, 1, 3)$, which is lexicographically smaller than $(2, 3, 1)$.

Input: Two numbers N, K

Output:

Print the lexicographically smallest permutation of the integers $1, \ldots, N$ which satisfies the above condition, in the format $A_1\, A_2\, \ldots\, A_N$.

If there is no such permutation, print -1.

Constraints:

$2 \leq N \leq 3 \times 10^5$

$1 \leq K \leq N - 1$

Sample Input 1:

3 1

Sample Output 1:

2 3 1

Two permutations satisfy the condition, (2,3,1) and (3,1,2). For instance, for (2, 3, 1) we have:

$|A_1 - 1| = 1 \geq K$

$|A_2 - 2| = 1 \geq K$

$|A_3 - 3| = 2 \geq K$

Sample Input 2:

8 3

Sample Output 2:

4 5 6 7 8 1 2 3

Sample Input 3:

8 6

Sample Output 3:

-1

Problem 2:

A fastidious farmer wants to pick apples from the apple trees in his orchard. His trees are arranged in a circle, so that the first tree is a neighbor of the last one. To keep his orchard beautiful, he cannot pick apples from neighboring trees.
You are given the number of apples on each tree. Design an algorithm for the farmer to pick the maximum number of apples, and output this quantity.

Input:
An array A of N numbers, where N is the number of apple trees, and A indicates the number of apples on each tree.

Output:
The maximum number of apples the farmer can pick which satisfies the above condition.

Constraints:

- $1 \leq N \leq 100$
- $0 \leq A_i \leq 1000$ for $A_i \in A$

Sample Input 1:
$[1, 2, 3]$

Sample Output 1:
3

Sample Input 2:
$[1, 2, 3, 1]$

Sample Output 2:
4

Problem 3:

Given an $n \times n$ matrix, an *increasing path* in the matrix is a path which starts at an arbitrary cell, and where the values of cells along the path are strictly increasing. Design an algorithm to find a longest increasing path in a given input matrix, and output the path's length.

Input:
An $n \times n$ matrix M, which each cell contains a value $M_{i,j}$.

Output:
The length of a longest increasing path in the matrix.

Constraints:

- $1 \leq n \leq 200$
- $0 \leq M_{i,j} \leq 2^{31} - 1$ for $A_{i,j} \in A$

Sample Input 1:
$[[3, 4, 5], [3, 2, 6], [2, 2, 7]]$

Sample Output 1:
5

Explanation 1:
The longest increasing path is $3, 4, 5, 6, 7$

Sample Input 2:
$[[9, 9, 3], [6, 6, 8], [2, 1, 1]]$

Sample Output 2:
4

Explanation 2:
The longest increasing path is $1, 2, 6, 9$

Sample Input 3:
$[[1]]$

Sample Output 3:
1

Problem 4:

Consider an infinite sequence of numbers consisting of all natural numbers arranged in ascending order. That is, consider the sequence $S = 12345678910111213141516171819202122\ldots$. Let S_i denote the i'th number in the sequence; *e.g.* $S_1 = 1, S_2 = 2, \ldots, S_{10} = 1, S_{11} = 0$, etc.
Let B be a given input value. Give an algorithm to compute the smallest index in S where B occurs. For example, if $B = 101$, then the index is 10, since $S_{10}S_{11}S_{12} = 101$, and no earlier substring of S equals 101.

Input:
A string representing the value B, with no more than 200 digits.

Output:
The smallest positive integer k for which $B_1 = S_k, B_2 = S_{k+1}, \ldots, B_r = S_{k+r-1}$ is true, where r is the number of digits in B.

Sample Input 1:
101

Sample Output 1:
10

Sample Input 2:
81

Sample Output 2:
27