Hong Kong Olympiad in Informatics 2014 Team Formation Test

Time allowed: 5 hours

Task Overview

Task	CPU time limit	Score
Bytefest	1 second	$10 + 15 + 16 + 12 + 47 = 100$
Word Game	1 second	$19 + 30 + 23 + 28 = 100$
Territory	2 seconds	$16 + 29 + 18 + 37 = 100$
Lost Sequence	3 seconds	$10 + 90 = 100$

Notice:

Unless otherwise specified, inputs and outputs shall follow the format below:
- One space between a number and another number or character in the same line.
- No space between characters in the same line.
- Each string shall be placed in its own separate line.
- No trailing space(s) in each line.
- No empty lines, except that the input and output should end with the endline character.

C++ programmers should be aware that using C++ streams (cin / cout) may lead to I/O bottlenecks and substantially slower performance.

C/C++ programmers should use `"%lld"` for 64-bit integers I/O.

For some problems 64-bit integers may be required. In Pascal it is `int64`. In C/C++ it is `long long int`.

Bytefest

Time Limit: 1 second

Problem

There are n cities in Byteland (labelled from 1 to n) and there are $n-1$ two-way roads connecting them. The roads are build such that it is possible to get from any city to another. Once a year in Byteland, there is a huge celebration called Bytefest. The festival can last as many days as the organizers want. One exciting event of the festival is the parade.

On each day, the parade team start from one city, parade along some roads and end in some city. When the parade team parade through road i, it will gives h_i happiness to the people. It is not allowed to pass through any road more than once. At the same time, the parade team can only parade at most m units length in a single day where the length of road i is l_i.

To reduce the inconvenience bring to the citizens, the parade team is not allowed to parade in any city and any road more than once during the whole Bytefest. In other words, the parade route should be disjoint for any two days, even sharing a common city is **not** allowed.

The goal is to maximize the sum of happiness during the festival by picking the routes optimally. Your task is to find this value.

Input

The first line contains 2 integers n and m. The next $n-1$ lines each contains 4 integers x_i, y_i, l_i, h_i, respecting a road connecting city x_i and y_i with length l_i and happiness h_i.

Output

Output a single integer, the maximum total happiness that can be reach.

Sample test

Input	Output
6 10 1 2 6 10 2 3 1 2 3 4 4 4 3 5 5 3 1 6 6 9	17

Explanation

In the first day, pick the route $4 \to 3 \to 5$. In the second day, pick the route $2 \to 1$.

Scoring

Subtask 1 (10 points)
$1 \leq n \leq 10$, $1 \leq m \leq 2000$, $1 \leq l_i \leq 2000$

Subtask 2 (15 points)
$1 \leq n \leq 100$, $1 \leq m \leq 2000$, $1 \leq l_i \leq 2000$

Subtask 3 (16 points)
$1 \leq n \leq 200$, $1 \leq m \leq 10^5$, $1 \leq l_i \leq 10^5$

Subtask 4 (12 points)
$1 \leq n \leq 200$, $1 \leq m \leq 10^{18}$, $1 \leq l_i \leq 10^9$

Subtask 5 (47 points)
$1 \leq n \leq 2000$, $1 \leq m \leq 10^{18}$, $1 \leq l_i \leq 10^9$

In addition $0 \leq h_i \leq 10^6$ for all test cases.

Word Game

Time Limit: 1 second

Problem

When Dr. Jones get bored after a day of work in the hospital, he plays with her daughter Laura. However, today Laura needs to attend some sports training after school, so Dr. Jones need some entertainment.

In a normal word find game, one needs to find a *substring* in a matrix that forms a English word. A substring is defined as a continuous sequence of character in the matrix from left to right horizontally, or up to down vertically.

Today, Dr. Jones would like to play a more mathematical version: the matrix is composed of integers, and he would like to find *good* substring $s_1 s_2 \dots s_m$ which satisfy:

- $m \geq 2$
- $s_1 \leq s_i$ $(i = 2, 3, \dots, m-1)$
- $s_m \leq s_i$ $(i = 2, 3, \dots, m-1)$

For example, `[5 3]` and `[1 2 4 2]` are good while `[1 2 5 4]` is not good.

Your task is, given a $n \times n$ integer matrix, find the number of good substrings that can be found in the matrix. Two substring are counted different if they are found in different sets of positions, even the sequence itself is identical.

Input

The first line contains of integer n. The next n lines each contains n integers describing the matrix. All integers in the matrix lies between 0 and 10^9.

Output

The output consist of a single integer, the number of good substring.

Sample test

Input	Output
4 3 1 4 1 2 5 8 5 7 4 1 2 5 5 6 3	33

Scoring

Subtask 1 (19 points)
$1 \leq n \leq 50$.

Subtask 2 (30 points)
$1 \leq n \leq 400$.

Subtask 3 (23 points)
$1 \leq n \leq 2500$. In addition the integers in the matrix are all distinct.

Subtask 4 (28 points)
$1 \leq n \leq 2500$.

Territory

Time Limit: 2 seconds

Problem

Once upon a time, Byteland was divided into n villages, labelled from 1 to n. There are also m two-way roads, each connects two different villages. There are k kingdoms ruling Byteland together and they are labelled from 1 to k. The i-th kingdom has its capital at village c_i. No two kingdom has capital located at the same village.

To divide the land to the kingdom, they decide to use the following agreement: for the i-th village, it belongs to the j-th kingdom if and only if the capital of the j-th kingdom is the **unique** closest kingdom to it. Here the distance is defined as the length of the shortest route from the i-the village to the capital of the j-th kingdom. If there does not exists a path between two vertex, their distance are considered to be infinity. Notice that under this definition, there may be village that does not belong to any kingdom, which we will call it a *neutral* village.

Your task is, find the owner of each village.

Input

The first line contains 3 integers n, m, k.
The second line contains k distinct integers $c_1, c_2, \ldots, c_k$.
The next m lines each contains 3 integers x_i, y_i, l_i $(x_i \neq y_i)$, respecting a two-way road with length l_i connecting village x_i and y_i .

Output

The output consist of n integers separated by a single space. The i-th integer should be the owner of the i-th village. If the city is neutral, print 0 instead.

Sample test

Input	Output
6 6 2 1 5 1 2 1 2 3 1 4 2 2 5 3 3 4 5 3 5 6 1	1 1 1 0 2 2

Scoring

Subtask 1 (16 points)
$1 \leq n \leq 1000$, $0 \leq m \leq 2000$, $l_i = 1$

Subtask 2 (29 points)
$1 \leq n \leq 1000$, $0 \leq m \leq 2000$, $1 \leq l_i \leq 10000$

Subtask 3 (18 points)
$1 \leq n \leq 100000$, $0 \leq m \leq 200000$, $l_i = 1$

Subtask 4 (37 points)
$1 \leq n \leq 100000$, $0 \leq m \leq 200000$, $1 \leq l_i \leq 10000$

In addition, $1 \leq k \leq n$ for all test cases.

Lost Sequence

Time Limit: 3 seconds

Problem

Legend says that once upon a time there was a sacred integer sequence a with length $2n$. If one can find the length of the *longest increasing subsequence* of a, he will receive great power and wisdom. For example, the longest increasing subsequence of `[2 6 3 5 1 5]` is `[2 3 5]`, which has a length 3.

After years of searching, Alex and Bob finally found pieces of a. However, Alex only get the first half of a and Bob only get the second half the a. Since they are good friends, Alex decided to send Bob a 01-message s and let Bob compute the length of the longest increasing subsequence of a. As paper is expensive, the message Alex writes should not be too long.

Fortunately, some error is allowed for Bob's answer. In particular, Bob can make at most 0.1% error for his output. That is, if the real answer is Ans, any integer in the range $[0.999Ans, 1.001Ans]$ is treated as an acceptable answer.

Your task is to write a program with two modes, `A` and `B`, that represents the behaviour of Alex and Bob:

Mode `A`
Input: The sequence $a_1, a_2, \ldots a_n$
Output: A `01`-sequence s, the message passing to Bob.
Mode `B`
Input: The sequence $a_{n+1}, a_{n+2}, \ldots a_{2n}$ and s, the message from Alex.
Output: The length of the longest increasing subsequence of a.

Input

Input for mode `A`:
The first line contains a single character `A`.
The second line contains a single integer n.
The third line contains n integers, the sequence $a_1, a_2, \ldots a_n$.
Input for mode `B`:
The first line contains a single character `B`.
The second line contains a single integer n.
The third line contains n integers, the sequence $a_{n+1}, a_{n+2}, \ldots a_{2n}$.
The forth line contains a single integer, the length of s.
The fifth line contains a 01-string s, the output by mode `A`.

Output

Output for mode `A`:
A 01-string s, which will be part of input for mode `B`. Print a new line if s is empty.
Output for mode `B`:
A single integer, the length of the longest increasing subsequence of a.

Judging

For each test case, we will first run your program in Mode `A`. If the program runs successfully and returns a valid output, we will run your program again in Mode `B`. This time the output, if also valid, will be used for scoring.

Scoring

If the error is not within 0.1% of the real answer in some test case within a subtask, you get 0 marks in that subtask. Let L be the maximum length of s in a subtask.

Subtask 1 (10 points)
$n = 1000, 0 \leq a_i \leq 1048575$.
You get 10 points if $L \leq 20000$.

Subtask 2 (90 points)
$n = 100000, 0 \leq a_i \leq 1048575$.
If $L > 2000000$, you get 0 point. Otherwise, the points you get equals

$$\min(90, \left\lfloor 35 - 42 \log_{10} \log_{10} \frac{L}{12500} \right\rfloor)$$

For example, you get 20 points in this subtask if $L = 2000000$. You get all 90 points if $L \leq 13993$.

Sample test

Input of Mode A	Output of Mode A
A 5 6 5 3 2 7	01100011010001

Input of Mode B	Output of Mode B
B 5 8 10 1 10 11 14 01100011010001	5

Note:
- The longest increasing subsequence is `[6 7 8 10 11]`.
- In the sample $n = 5$. However in real tests n will be either 1000 or 100000.
- `01100011010001` is only a sample to illustrate the format and has no special meaning.