

2011 Canadian Computing Competition

Day 2, Problem 1

Reorganization

Input: from standard input
Output: to standard output
Source file: `reorg.{c,cc,pas}`

Problem Description

Alice and Bob own a huge company. This company was losing money consistently over the last 30 years, since its owners spent too much time playing games with mathematicians. Alice and Bob decide to make a change.

Alice and Bob start by giving unique employee IDs to each of the n employees ($1 \leq n \leq 100,000$), where each ID I is in the range ($1 \leq I \leq 100,000$).

Then, Alice and Bob give unique ranks to each employee. Each rank R is an integer such that $1 \leq R \leq 10,000,000$. After this, they plan to reorganize the company, by making sure that the employees satisfy the following conditions:

1. There is exactly one director, who has no supervisor;
2. Except for the director, each employee has a supervisor, and this supervisor has a smaller employee ID and a higher rank (the value of rank is smaller); and
3. Each employee can supervise at most 2 people.

Alice and Bob are eager to know whether their company can be reorganized successfully.

Input Specification

The input is a total of $n + 1$ lines. The first line contains n ($1 \leq n \leq 100,000$), indicating the number of employees. On the next n lines are n distinct integers R ($1 \leq R \leq 10,000,000$), one integer per line, where the i th integer indicates the rank of the employee with ID i .

Output Specification

Output YES if the company can be reorganized successfully; output NO otherwise.

Sample Input 1

```
6
1
6
5
```

2
3
4

Output for Sample Input 1

NO

Explanation of Output for Sample Input 1

Employee with rank 1 has employee ID 1, and thus, must be the supervisor. Employees 2 and 3 (with ranks 6 and 5) can only be supervised by employee 1 (with rank 1). However, no other employee (4, 5 or 6) can be supervised by employee 2 or employee 3, since ranks of supervisors must be smaller than the employees they supervise.

Sample Input 2

6
1
6
2
3
4
5

Output for Sample Input 2

YES

Explanation for Output for Sample Input 2

Employee 1 (rank 1) supervises both employee 2 (rank 6) and employee 3 (rank 2).

Employee 3 (rank 2) supervises employee 4 (rank 3) and employee 5 (rank 4).

Employee 4 (rank 3) supervises employee 6 (rank 5).

2011 Canadian Computing Competition
Day 2, Problem 2
Fixing Disks

Input: from standard input

Output: to standard output

Source file: `disk.{c,cc,pas}`

Problem Description

You are playing a game with a stack of N disks ($1 \leq N \leq 100$). The goal of the game is remove all of the disks from your stack. However, there is a cost associated with removing disks, and you wish to minimize the cost of removing all the disks from your stack.

Each disk contains a label, with the label L being in the range $1 \leq L \leq 20$.

You are also given a “Master stack” of N disks which you can use to help you remove disks from your stack.

You can remove disks from the top of your stack of disks in two ways:

1. remove a disk at the top of your stack: if label of the disk on top of your stack is c , that disk can be removed from your stack with cost c ;
2. remove a disk from both the top of the Master stack and the top of your stack if the label is the same between both disks: in this case, there is no cost with removing both disks.

You are also allowed to modify the order of the top K ($1 \leq K \leq 4$) elements of your stack, so long as immediately after each modification, you remove the top element of your stack. There are three allowed modifications:

1. Reverse: you may reverse the order of the top r ($2 \leq r \leq K$) disks on your stack. In other words, if the top r disks are d_1, d_2, \dots, d_r (reading from the top down), then after this operation, the top r disks will be d_r, \dots, d_2, d_1 (reading from the top down). The cost of one reverse operation is R ($1 \leq R \leq 1,000,000$).
2. (Cyclic Shift) Up: you can shift up one disk in the range of the top u disks ($2 \leq u \leq K$). For example, if the top four disks are d_1, d_2, d_3, d_4 (read from the top down), you can perform an up shift in the range of the top three elements to get d_2, d_3, d_1, d_4 or an up shift of the top four elements to get d_2, d_3, d_4, d_1 . The cost of one up shift operation is U ($1 \leq U \leq 1,000,000$).
3. (Cyclic Shift) Down: you can shift down one disk in the range of the top d disks ($2 \leq d \leq K$). For example, if the top four disks are d_1, d_2, d_3, d_4 (read from the top down), you can

perform a down shift in the range of the top three elements to get d_3, d_1, d_2, d_4 or a down shift of the top four elements to get d_4, d_1, d_2, d_3 . The cost of one down shift operation is D ($1 \leq D \leq 1,000,000$).

If the operations yield a match between the top of the master stack and the top of your stack, you can pop for free. If not, you must pay the cost of the pop.

There is one additional constraint. At any point in the game, if a disk at level j is being popped (levels start at 0 at the bottom of the stack), then all elements that were *originally* at level $j + M$ or higher must have already been popped, where $1 \leq M \leq 5$.

Minimize the cost required to pop all the elements off of your stack.

Input Specification

The first line will contain 6 space-separated integers: N, K, M, D, U, R where:

- N is the number of disks in each of the stacks ($1 \leq N \leq 100$)
- K , the maximum depth at which operations can be done ($1 \leq K \leq 4$)
- M , the threshold before which elements can be removed from our stack, ($1 \leq M \leq 5$)
- D , the cost for a shift in which the bottom element of the selected range goes to the top ($1 \leq D \leq 10^6$)
- U , the cost for an up shift operation (in which the top element of the selected range goes to the bottom), ($1 \leq U \leq 10^6$)
- R , the cost for reversing the top elements of the selected range ($1 \leq R \leq 10^6$)

$2N$ lines will follow, each containing a number L ($1 \leq L \leq 20$). The first N lines will contain the labels of the disks in the Master stack, from top to bottom. The next N lines will contain the labels of the disks in your stack, from top to bottom

Note: for 20% of the marks for this question, you may assume that $K \leq 2$.

Output Specification

Output the integer (on one line) which is the minimal cost required to remove all disks from your stack.

Sample Input

```
7 3 3 4 4 3
5
6
3
```

5
4
1
2
3
5
6
5
1
4
1

Output for Sample Input

5

Explanation for Output for Sample Input

We take 3 to the bottom and shift the two blocks below it up, with cost 4. Then we remove four blocks from each stack, remove a 1 from the playing stack, with cost 1, and remove two blocks from each stack.

2011 Canadian Computing Competition
Day 2, Problem 3
Biggest (Zero Carbon) Footprint

Input: from standard input
Output: to standard output
Source file: `bigfoot.{c,cc,pas}`

Problem Description

Having just recently won the lottery, you decide to build a summer resort nestled deep in a forest. However, being a very eco-friendly person, you decide not to cut down any of the trees that grow in the forest. Given a map of the forest and the positions of its trees, determine the area of the largest rectangular plot you can buy that does not contain any of the trees. (Note that your plot must have edges which are parallel to the x and y axes.)

Input Specification

The first line contains n , m , and t ($0 < n, m < 10,000$, $0 < t < 10,000$) representing the dimensions of the given map of the forest and the number of trees indicated on the map respectively. The next t lines each contain two integers x and y ($0 \leq x \leq n$, $0 \leq y \leq m$) describing the location of each tree (where $(0, 0)$ is the bottom leftmost point on the map and (n, m) is the top rightmost point on the map).

Note: for 20% of the marks for this question, you may assume that $t \leq 100$, and for 45% of the marks for this question, $t \leq 400$.

Output Specification

Output the area of the largest rectangle that does not contain any of the given trees.

Sample Input

```
5 5 2
1 1
3 3
```

Output for Sample Input

```
12
```