Practice Compete Jobs Rank Leaderboard

Dashboard > Algorithms > Greedy > Cutting Boards

Badge Progress (Details)

Points: 126 Rank: 270114

Cutting Boards

by dweep123

Problem Submissions Leaderboard Discussions Editorial

Chinese Version
Russian Version

Alice gives Bob a board composed of $m \times n$ wooden squares and asks him to find the minimum cost of breaking the board back down into individual 1×1 pieces. To break the board down, Bob must make cuts along its horizontal and vertical lines.

To reduce the board to squares, x_{n-1} vertical cuts must be made at locations $x_1, x_2, \ldots, x_{n-2}, x_{n-1}$ and y_{m-1} horizontal cuts must be made at locations $y_1, y_2, \ldots, y_{m-2}, y_{m-1}$. Each cut along some x_i (or y_j) has a cost, c_{x_i} (or c_{y_j}). The total cost of a cut is $n \times c$, where n is the number of already-cut segments that the cut passes through.

The cost of cutting the whole board down into 1×1 squares is the sum of the cost of each successive cut. Recall that the cost of a cut is multiplied by the number of already-cut segments it crosses through.

Can you help Bob find the minimum cost?

Input Format

The first line contains a single integer, q, denoting the number of queries. The $3 \cdot q$ subsequent lines describe each query over 3 lines according to the following format:

1. The first line has two positive space-separated integers, m and n, detailing the respective height (y) and width (x) of the board.
2. The second line has $m - 1$ space-separated integers listing the cost, c_{y_j}, of cutting a segment of the board at each respective location from $y_1, y_2, \ldots, y_{m-2}, y_{m-1}$.
3. The third line has $n - 1$ space-separated integers listing the cost, c_{x_i}, of cutting a segment of the board at each respective location from $x_1, x_2, \ldots, x_{n-2}, x_{n-1}$.

Note: If we were to superimpose the $m \times n$ board on a 2D graph, x_0, x_n, y_0, and y_n would all be edges of the board and thus not valid cut lines.

Constraints

- $1 \le q \le 20$
- $2 \le m, n \le 1000000$
- $0 \le c_{x_i}, c_{y_j} \le 10^9$

Output Format

For each of the q queries, find the minimum cost ($minCost$) of cutting the board into 1×1 squares and print the value of $minCost \ \% \ (10^9 + 7)$.

Sample Input 0

```
1
2 2
2
1
```

Sample Output 0

```
4
```

Explanation 0

We have a 2×2 board, with cut costs $c_{y_1} = 2$ and $c_{x_1} = 1$. Our first cut is horizontal at y_1, because that is the line with the highest cost (2). Our second cut is vertical, at x_1. Our first cut has a $totalCost$ of 2, because we are making a cut with cost $c_{y_1} = 2$ across 1 segment (the uncut board). The second cut also has a $totalCost$ of 2, because we are making a cut of cost $c_{x_1} = 1$ across 2 segments. Thus, our answer is $minCost = ((2 \times 1) + (1 \times 2))\ \%\ (10^9 + 7) = 4$.

Sample Input 1

```
1
6 4
2 1 3 1 4
4 1 2
```

Sample Output 1

```
42
```

Explanation 1

Our sequence of cuts is: y_5, x_1, y_3, y_1, x_3, y_2, y_4 and x_2.
Cut 1: Horizontal with cost $c_{y_5} = 4$ across 1 segment. $totalCost = 4 \times 1 = 4$.
Cut 2: Vertical with cost $c_{x_1} = 4$ across 2 segments. $totalCost = 4 \times 2 = 8$.
Cut 3: Horizontal with cost $c_{y_3} = 3$ across 2 segments. $totalCost = 3 \times 2 = 6$.
Cut 4: Horizontal with cost $c_{y_1} = 2$ across 2 segments. $totalCost = 2 \times 2 = 4$.
Cut 5: Vertical with cost $c_{x_3} = 2$ across 4 segments. $totalCost = 2 \times 4 = 8$.
Cut 6: Horizontal with cost $c_{y_2} = 1$ across 3 segments. $totalCost = 1 \times 3 = 3$.
Cut 7: Horizontal with cost $c_{y_4} = 1$ across 3 segments. $totalCost = 1 \times 3 = 3$.
Cut 8: Vertical with cost $c_{x_2} = 1$ across 6 segments. $totalCost = 1 \times 6 = 6$.

When we sum the $totalCost$ for all minimum cuts, we get $4 + 8 + 6 + 4 + 8 + 3 + 3 + 6 = 42$. We then print the value of $42\ \%\ (10^9 + 7)$.

Submissions:6416

Max Score:60
Difficulty: Hard

Rate This Challenge:
☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)

Java 7

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
    }
}
```

Line: 1 Col: 1

Upload Code as File

□ Test against custom input

Run Code

Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature