Bit Shift Matrix

Java May'19 Advanced Practice 4 - 1 day 01:58:02

Submit solution

My submissions
All submissions
Best submissions

Points: 100 (partial)
Time limit: 1.0s
Memory limit: 32M

Tags
Arrays
Difficulty
Intermediate

Allowed languages
C#, java, JavaScript

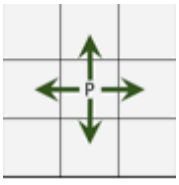
Problem 3 – Bit Shift Matrix

	0	1	2	3	4	5
0	16	32	64	128	256	512
1	8	16	32	64	128	256
2	4	8	16	32	64	128
3	2	4	8	16	32	64
4	1	2	4	8	16	32

You are given rectangular matrix. The matrix is filled with numbers that are power of 2, as follows:

- The bottom left corner holds the value 1
- The next cell above holds value of 2, the next cell above holds of 4, etc…
- The second cell the bottom row holds a value of 2, the cell next to it holds a value of 4

You have a pawn on the field. The pawn can only move to the cells that directly above, below it or right/left of it. We have four directions UP, DOWN, LEFT, RIGHT.



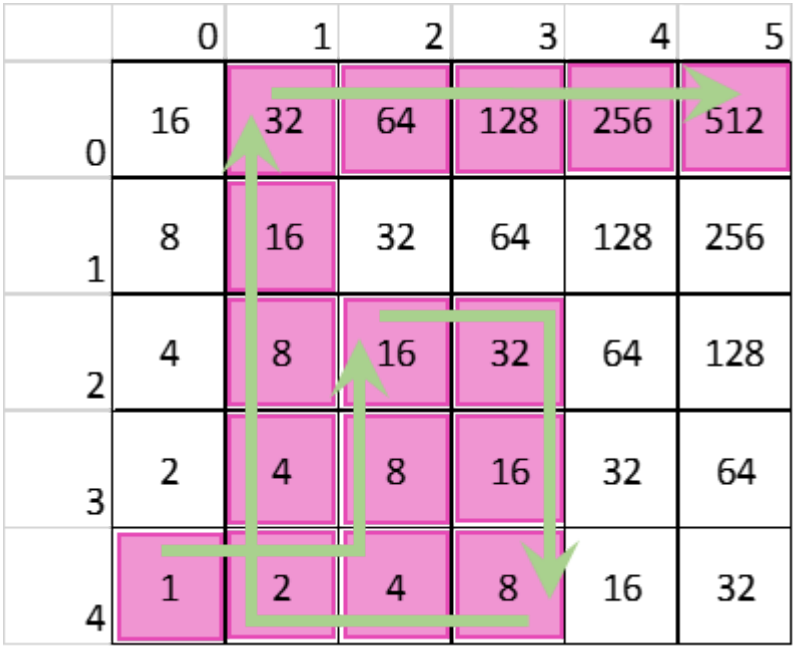
Given that initially the pawn starts at the bottom left corner and a list of cells that the pawn must reach calculate the sum of the cells that the pawn has to go through.

The top row is at position 0, the bottommost ro

The pawn goes from one cell to the other, following the rules:

- First go to the target column
- Second go to the target row

Example:



The pawn collects values: 1, 2, 4, 8, 16, 32, 16, 8, 4, 8, 16, 32, 64, 128, 256 and 512. Their sum is 1107.

Input

The input data is given at the standard input, i.e. the console

On the first and the second lines you will find **the dimensions of the field R and C**

On the third line you will find the number **N**, **the number of moves**

On the fourth line you will find the CODEs, **decimal numbers** that represents the positions from the path of the pawn. They will be separated by a single space. The position is encoded as follows:

- A coefficient is calculated, COEFF = MAX(R, C)
- ROW = CODE / COEF
- COL = CODE % COEF

The input will be valid, in the specified format, within the constraints given below. There is no need to check the input data explicitly.

Output

Print the sum of cells contained in the path of pawn

- **R** will always be between **1 and 100**
- **C** will always be between **1 and 75**
- **N** will always be between **1 and 1000**

Sample Input

```
5
6
4
14 27 1 5
```

Copy

Sample Output

```
1107
```

Copy

Clarifications

No clarifications have been made at this time.