X

Week 8 Programming Assignment

Due on 2020-11-12, 23:59 IST

For this assignment, you have to write a complete Python program.

- You may define additional auxiliary functions as needed.
- In all cases you may assume that the input to your program has the expected format, so your program does not have to check for malformed inputs.
- There are 3 public test cases and 12 (hidden) private test cases.
- "Compile and run" will evaluate your submission against the public test cases.
- "Submit" will evaluate your submission against the hidden private test cases and report a score on 100.
- Ignore warnings about "Presentation errors".

Here there be Dragons

(IOI Training Camp, 2012)

The kingdom is falling into ruin. People live in fear. Dragons pillage, kill, and just generally cause as much havoc as they possibly can. The king has just sent out a royal decree:

> *To any man out there who is able to bring me the heads of K dragons, I shall bequeath a lordship–to him, his sons and his grandsons, till the end of time.*

Having seen this royal decree, and knowing that you are capable of killing dragons thanks to your extensive medieval combat training, you set out on a quest to hunt down the evil creatures. Being a busy kind of guy, you would like to complete your quest

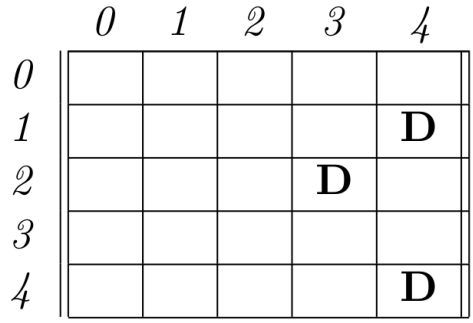
quickly and kill K dragons through the shortest route.

The kingdom is arranged in a grid with R rows, numbered 0 to R-1, and C columns, numbered 0 to C-1 You start your quest at the top left corner of the grid, (0,0).

The total number of dragons in the kingdom is D, of which you have to kill K. Dragons are very territorial in nature, so each row of the grid contains at most one dragon. Also, since the kingdom is situated on a hill, you travel only downwards on the grid, though you may move left or right as you please.

You are told that no two dragons are on the same row of the grid. Also, no dragon is at position (0,0).

For example, suppose the grid has 5 rows and 5 columns with 3 dragons, of which you have to kill any 2. The three dragons are located at (1,4), (2,3) and (4,4), as shown below. In this case, your shortest route is to take 7 steps and kill the dragons in row 1 and row 2. Killing any other combination of 2 dragons takes 8 steps, so this is the minimum possible. Note that once you've killed K dragons, you don't incur any cost to return home. You just want to find how long it takes to do all the killing.



Solution hint

Number the dragons 1,2,…,D in ascending order of rows. Let mindist(i,j) denote the minimum distance travelled when the j^{th} dragon killed is dragon i. Recall the constraint that there is no dragon at (0,0). Use dynamic programming to compute mindist(i,j) for all values of i and j, then find the minimum among mindist(i,K) for all i ≥ K.

Input format

- Line 1 : Four space-separated integers, R, C, K and D.
- Lines 2 to D+1 : Each line has two-space separated integers r and c, the row and column of the corresponding dragon.

Output format

A single integer, the minimum total distance travelled to kill K dragons.

Test Data:

- In all testcases, K ≤ D ≤ R, and, for each dragon position (r,c), 0 ≤ r < R, and 0 ≤ c < C.
- In all testcases, 1 ≤ K,D ≤ 300

Text Transcripts

Books

Download Videos

- In all testcases, $1 \le R,D \le 300$.
- In 60% of the testcases, $1 \le R,C \le 300$. In the remaining testcases, $1 \le R,C \le 100000$.
- No two dragons will be on the same row.
- No dragon will be at position (0,0).

Sample Input:

```
5 5 2 3
1 4
4 4
2 3
```

Sample Output:

```
7
```

Select the Language for this assignment. --

You may submit any number of times before the due date. The final submission will be considered for grading.

This assignment has Public Test cases. Please click on "Compile & Run" button to see the status of Public test cases. Assignment will be evaluated only after submitting using Submit button below. If you only save as or compile and run the Program , your assignment will not be graded and you will not see your score after the deadline.

Save as Draft | **Compile & Run** | **Submit** | **Reset**

Sample Test Cases

	Input	Output
Test Case 1	5 5 2 3 1 4 4 4 2 3	7
Test Case 2	5 5 3 4 1 1 3 2 2 4 0 4	9

Test Case 3	10 10 5 7 1 7 3 6 5 1 6 5 7 4 4 3 2 2	17