

Queen's Attack II

332.41 more points to get your next star!

Rank: 24809 | Points: 1867.59/2200



Problem

Submissions

Leaderboard

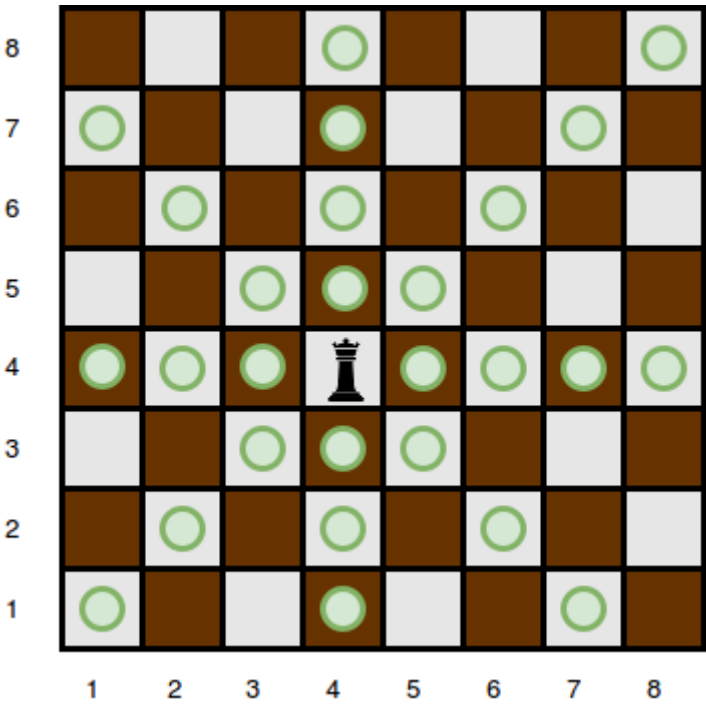
Discussions

Editorial

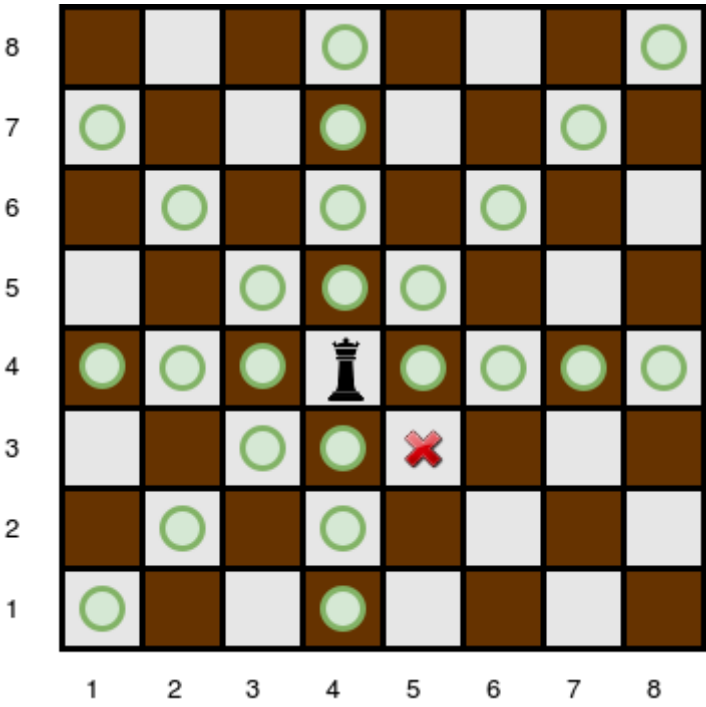
You will be given a square chess board with one queen and a number of obstacles placed on it. Determine how many squares the queen can attack.

A **queen** is standing on an  $n \times n$  chessboard. The chess board's rows are numbered from **1** to  $n$ , going from bottom to top. Its columns are numbered from **1** to  $n$ , going from left to right. Each square is referenced by a tuple,  $(r, c)$ , describing the row,  $r$ , and column,  $c$ , where the square is located.

The queen is standing at position  $(r_q, c_q)$ . In a single move, she can attack any square in any of the eight directions (left, right, up, down, and the four diagonals). In the diagram below, the green circles denote all the cells the queen can attack from  $(4, 4)$ :



There are obstacles on the chessboard, each preventing the queen from attacking any square beyond it on that path. For example, an obstacle at location  $(3, 5)$  in the diagram above prevents the queen from attacking cells  $(3, 5)$ ,  $(2, 6)$ , and  $(1, 7)$ :



Given the queen's position and the locations of all the obstacles, find and print the number of squares the queen can attack from her position at  $(r_q, c_q)$ . In the board above, there are **24** such squares.

Function Description

Author

bishop15

Difficulty

Medium

Max Score

30

Submitted By

25616

NEED HELP?

- View discussions
- View editorial
- View top submissions

RATE THIS CHALLENGE



MORE DETAILS

- Download problem statement
- Download sample test cases
- Suggest Edits



Complete the queensAttack function in the editor below. It should return an integer that describes the number of squares the queen can attack.

queensAttack has the following parameters:

- n: an integer, the number of rows and columns in the board
- k: an integer, the number of obstacles on the board
- r\_q: integer, the row number of the queen's position
- c\_q: integer, the column number of the queen's position
- obstacles: a two dimensional array of integers where each element is an array of **2** integers, the row and column of an obstacle

**Input Format**

The first line contains two space-separated integers ***n*** and ***k***, the length of the board's sides and the number of obstacles.

The next line contains two space-separated integers ***r<sub>q</sub>*** and ***c<sub>q</sub>***, the queen's row and column position.

Each of the next ***k*** lines contains two space-separated integers ***r[i]*** and ***c[i]***, the row and column position of ***obstacle[i]***.

**Constraints**

- $0 < n \leq 10^5$
- $0 \leq k \leq 10^5$
- A single cell may contain more than one obstacle.
- There will never be an obstacle at the position where the queen is located.

**Subtasks**

For **30%** of the maximum score:

- $0 < n \leq 100$
- $0 \leq k \leq 100$

For **55%** of the maximum score:

- $0 < n \leq 1000$
- $0 \leq k \leq 10^5$

**Output Format**

Print the number of squares that the queen can attack from position ***(r<sub>q</sub>, c<sub>q</sub>)***.

**Sample Input 0**

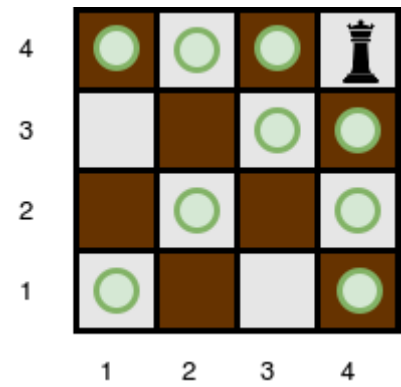
```
4 0
4 4
```

**Sample Output 0**

```
9
```

**Explanation 0**

The queen is standing at position ***(4, 4)*** on a **4 × 4** chessboard with no obstacles:



Sample Input 1

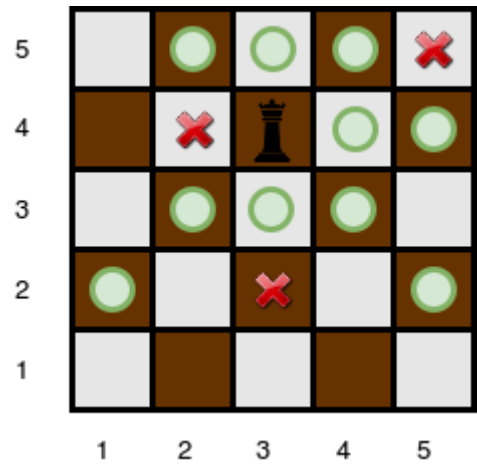
```
5 3
4 3
5 5
4 2
2 3
```

Sample Output 1

```
10
```

Explanation 1

The queen is standing at position **(4,3)** on a **5 × 5** chessboard with **k = 3** obstacles:



The number of squares she can attack from that position is **10**.

Sample Input 2

```
1 0
1 1
```

Sample Output 2

```
0
```

Explanation 2

Since there is only one square, and the queen is on it, the queen can move 0 squares.

Rust

```
1 // Enter your code here
2
3
```

Line: 1 Col: 1



[Upload Code as File](#)



Test against custom input

Run Code

Submit Code