

# Divide and Conquer

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            2 seconds  
Memory limit:         1024 megabytes

Fall has an array of length  $n$  and  $q$  queries on this array. Each query is an interval  $[l_i, r_i]$ .

Fall decides to handle these queries using a divide-and-conquer approach. Initially, all queries to be processed lie within the interval  $[1, n]$ , which is at layer 1. Suppose the current set of queries to be processed lies within  $[L, R]$  at layer  $d$ . He may choose a position  $x$  from  $[L, R]$  and process all queries whose intervals contain  $x$  at layer  $d$ . The remaining queries lie in  $[L, x - 1]$  and  $[x + 1, R]$ , which can be handled recursively as two subproblems at layer  $d + 1$  (if there are still queries in those intervals).

You are asked to minimize the maximum layer number at which any query is processed.

## Input

Each test file contains multiple test cases. The first line contains the number of test cases  $T$  ( $1 \leq T \leq 10^4$ ). The description of the test cases follows.

The first line of each test case contains two integers  $n$  and  $q$  ( $1 \leq n, q \leq 10^6$ ), denoting the length of the array and the number of queries.

Each of the next  $q$  lines contains two integers  $l_i$  and  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ ), representing a query.

For each test file, it is guaranteed that the sum of  $n$  over all test cases does not exceed  $10^6$ , and the sum of  $q$  over all test cases does not exceed  $10^6$ .

## Output

For each test case, output one line containing a single integer: the minimum possible value of the maximum layer number at which the queries are processed.

## Example

standard input	standard output
3	3
5 5	1
1 5	2
1 1	
2 2	
4 4	
5 5	
5 2	
1 3	
3 5	
3 3	
1 1	
2 2	
3 3	

## Note

For the first sample test case, one optimal divide-and-conquer strategy is: choose  $x = 3$ , process the first query at layer 1, recurse the second and third queries into the interval  $[1, 2]$ , and recurse the fourth and fifth queries into the interval  $[4, 5]$ . Both child intervals require one more recursion, so the maximum layer at which queries are processed is 3.