



Task Tests

 0.8 sec.  512 MB

In this task we will try to recreate one of the challenges for the authors of tasks - generating tests. Your challenge is to generate trees* and to answer queries on them. The detailed description is below.

*Tree - a graph without a cycle.



Implementation details

You have to implement the following function:

```
std::vector<std::vector<std::pair<int,int>>> generate(int N, int T);
```

It will be called once by the jury. You have to return T distinct trees (as a list of edges) with exactly N vertices, in which the root is 1. The edges in the trees must not have a weight. If these conditions are not satisfied, you get a wrong answer on the respective test. After that you have to implement the following three functions, with which you will answer queries:

```
int query(int l, int r);
```

This function can be called multiple times by the jury. You have to return the answer to the following question about the trees you generated in the previous function:

"In an interval of indices l to r of the trees, return the height* of the tree which has the greatest height in this interval."

*Height of a tree - the greatest distance from the root of the tree to some vertex. As a distance we take the number of edges needed to reach the vertex from the root.

```
void remove(int ind);
```

This function can be called multiple times by the jury. It must make the height of the tree with index `ind` equal to 0 when answering the queries, i.e. the respective tree does not count.

```
void add(int ind);
```

This function can be called multiple times by the jury. It must restore the original height of the tree with index `ind`.



Constraints

Let the sum of the number of all calls to query, remove and add be equal to Q .

- $1 \leq N, T \leq 1067$
- $0 \leq Q \leq 5 * 10^5$
- $0 \leq l, r, ind < T$



Subtasks

Subtask	Points	Constraint
1	6	$1 \leq N, T \leq \text{about } 6 - 7, Q = 0$
2	9	$1 \leq N, T \leq 34, Q = 0$
3	9	$1 \leq N, T \leq 167, Q = 0$
4	11	$1 \leq N, T \leq 1067, Q = 0$
5	4	$1 \leq N, T \leq \text{about } 6 - 7, Q = 1$
6 – 7	4	$1 \leq N, T \leq 34, Q = 1$
6 – 7	4	$1 \leq N, T \leq 167, Q = 1$
8	4	$1 \leq N, T \leq 1067, Q = 1$
9	9	$1 \leq N, T \leq 1067, 1 \leq Q \leq 5 * 10^5$, there are no add, remove queries
10	6	$1 \leq N, T \leq \text{about } 6 - 7, 1 \leq Q \leq 5 * 10^5$
11	7	$1 \leq N, T \leq 34, 1 \leq Q \leq 5 * 10^5$
12	3	$1 \leq N, T \leq 167, 1 \leq Q \leq 5 * 10^5$
13	24	$1 \leq N, T \leq 1067, 1 \leq Q \leq 5 * 10^5$



Example

Jury	Contestant
<code>generate(4, 1)</code>	<code>{{{1, 2}, {2, 3}, {3, 4}}}</code>
<code>query(0, 0)</code>	3
<code>remove(0)</code>	
<code>query(0, 0)</code>	0
<code>add(0)</code>	
<code>query(0, 0)</code>	3



Sample grader

The format of the input is the following:

- line 1: N, T, Q .
- lines 2 – $(Q + 1)$: type of query (0 - query, 1 - add, 2 - remove) and its arguments.

The grader first prints what was returned by `generate`, and after that answers all queries of type 0.