Graph

Bob lives in a city of mountain $G(P)$. In this city, there are M mountains with **distinct** heights, where the height of mountain i is $P[i]$ ($0 \le i < M$). Bob can only travel through the city using bridges that connect the mountains.

For mountain i and mountain j, if the heights of all mountains with indices between i and j (inclusive) are within the range of the heights of mountain i and mountain j, i.e., for every integer $k \in [i, j]$, it holds that $\min(P[i], P[j]) \le P[k] \le \max(P[i], P[j])$, there exists a bridge between them. Otherwise, there does not exist a bridge between them.

Bob lives on mountain 0 and works on mountain $M-1$. He wants to know the minimum number of bridges he needs to cross to get to work, denoted by $F(P)$.

You are given $A[0], A[1], \cdots, A[N-1]$, a permutation of $0, 1, \cdots, N-1$. There will be Q queries. The i-th $(1 \le i \le Q)$ query gives you two integers $l[i], r[i]$ $(0 \le l[i] \le r[i] < N)$, and you should find the value of $F(\left[A[l[i]], A[l[i]+1], \cdots, A[r[i]]\right])$ for Bob.

Implementation details

You need to implement the following functions:

```
void init_permutation(int N, std::vector<int> A);
```

- N: the length of the given permutation.
- A: an array of length N, denoting the permutation A in the statements.
- This function will be called exactly once at the beginning.

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

- After calling `init_permutation`, there will be Q calls of `calc_f`.
- This function should return $F(\left[A[l], A[l+1], \cdots, A[r]\right])$, denoting the shortest path from vertex 0 to vertex $M-1$ in $G(\left[A[l], A[l+1], \cdots, A[r]\right])$.

Example

Consider the following call:

```
init_permutation(5, {0, 3, 1, 2, 4});
```

Followed by the following calls:

1. `calc_f(0, 4);`. You should return 1, because this query asks the value of $F(\big[0,3,1,2,4\big])$:

   $G(\big[0,3,1,2,4\big])$ consists of 5 vertices $0,1,2,3,4$, and we are asked to find the shortest path between 0 and 4.

   From the definition of G, 0 and 4 are directly connected, because for every integer $0 \le k \le 4$, $0 = \min(A[0], A[4]) \le A[k] \le \max(A[0], A[4]) = 4$. Thus, the length of the shortest path from 0 to 4 is 1, so you should return 1.

2. `calc_f(0, 3);`. You should return $F(\big[0,3,1,2\big])$, which is equal to 3.

3. `calc_f(1, 4);`. You should return $F(\big[3,1,2,4\big])$, which is equal to 2.

4. `calc_f(1, 1);`. You should return $F(\big[3\big])$, which is equal to 0.

Constraints

- $1 \le N, Q \le 3 \times 10^5$
- $0 \le A[i] < N$ $(0 \le i < N)$
- $0 \le l[i] \le r[i] < N$ $(1 \le i \le Q)$

Subtasks

1. (1 point): $N \le 10, Q \le 10$.
2. (19 points): $N \le 5000, Q \le 5000$.
3. (10 points): $N \le 10^4$,
4. (10 points): $N \le 10^5, Q \le 10^5$, the permutation A is sampled uniformly from all permutations.
5. (60 points): No additional constraints.

Sample Grader

The sample grader reads the input in the following format:

- Line 1: N Q
- Line 2: $A[0]$ $A[1]$ $\cdots$ $A[N-1]$
- Line $2+i$ $(1 \le i \le Q)$: $l[i]$ $r[i]$

The sample grader prints your answer in the following format:

- Line i $(1 \le i \le Q)$: your answer of the i-th query