2-Dimensional Range Minimum Query

Time limit: 2.0s **Memory limit:** 1G

You are given a 2-dimensional array of integers of size $N \times N$. We will call this array arr. You are then required to answer multiple queries for the minimum integer in a given submatrix.

Formally, given 4 integers a, b, c, d, determine $\min\{arr[i][j] \mid a \leq i \leq b, c \leq j \leq d\}$.

Implementation Details

You should implement the following functions:

```
void init(std::vector<std::vector<int>> arr);
```

- arr: a 2-dimensional array of size $N \times N$.

```
int query(int a, int b, int c, int d);
```

- a: the lower bound on the first dimension of the submatrix.
- b: the upper bound on the first dimension of the submatrix.
- c: the lower bound on the second dimension of the submatrix.
- d: the upper bound on the second dimension of the submatrix.
- This function should return the minimum integer in the submatrix formed by a, b, c, d.

It is guaranteed that `init` will be called exactly once, and that this call will be made before any calls to `query`.

Constraints

For all subtasks:

$1 \leq N \leq 1000$
$1 \leq arr[i][j] \leq 10^9$ for $0 \leq i, j \leq N - 1$
$0 \leq a \leq b \leq N - 1$ for all calls to `query`
$0 \leq c \leq d \leq N - 1$ for all calls to `query`

Disclaimer: $arr[i][j], a, b, c, d$ were generated randomly.
Your answer will be considered correct if the bitwise xor of the answers to all the queries matches the expected result.

Subtask 1 [10%]

`query` will be called at most $1\,000$ times.

Subtask 2 [20%]

`query` will be called at most $100\,000$ times.

Subtask 3 [30%]

`query` will be called at most $1\,000\,000$ times.

Subtask 4 [40%]

`query` will be called at most $10\,000\,000$ times.

Sample Interaction

Please note that the sample will not appear in any of the test cases.

Function Call	Return Value
`init({{1, 2}, {3, 4}})`	
`query(0, 1, 0, 1)`	`1`
`query(1, 1, 0, 1)`	`3`
`query(0, 0, 1, 1)`	`2`