**Mo,s Algorithm (query square root decompostion)**

\* It is offline algorithm . We need full input to right this algorithm.

It is algorithm use for query optimization .like we have n queries for L to R range that worst case it take order of O(n2) time.

We can do it O(sqrt(n) \*N).

Link- <https://blog.anudeep2011.com/mos-algorithm/>

Given an array of size N. All elements of array <= N. You need to answer M queries. Each query is of the form L, R. You need to answer the count of values in range [L, R] which are repeated at least 3 times.  
Example: Let the array be {1, 2, 3, 1, 1, 2, 1, 2, 3, 1} (zero indexed)  
Query: L = 0, R = 4. Answer = 1. Values in the range [L, R] = {1, 2, 3, 1, 1} only 1 is repeated at least 3 times.  
Query: L = 1, R = 8. Answer = 2. Values in the range [L, R] = {2, 3, 1, 1, 2, 1, 2, 3} 1 is repeated 3 times and 2 is repeated 3 times. Number of elements repeated at least 3 times = Answer = 2.

Sort the queries first like this .

Here is the example-

How does the final order look like?  
All the queries are first ordered in ascending order of their block number (block number is the block in which its opening falls). Ties are ordered in ascending order of their R value.  
For example consider following queries and assume we have 3 blocks each of size 3.  
{0, 3} {1, 7} {2, 8} {7, 8} {4, 8} {4, 4} {1, 2}  
Let us re-order them based on their block number.  
{0, 3} {1, 7} {2, 8} {1, 2} {4, 8} {4, 4} {7, 8}  
Now let us re-order ties based on their R value.  
{1, 2} {0, 3} {1, 7} {2, 8} {4, 4} {4, 8} {7, 8}

Code

Mo algo-

**int Block= sqrt(N)** //numbers of element

Bool compare( query a, query b) // query is of struct type have L, R and I (postion of

if (a.L/Block!=b.L/block)

return a.L/Block<=b.L/Block;

else

Return a.R<=b.R;