Data Structures - Assignment 1

Name - Gaurang Shukla
Rollno - CS2415
Course - Mtech CS

Write a program to find the minimum element between two given indices i and j (where i<j) of a given array. Also, explain your algorithm and complexity.

Example:- Given an array, whose elements are
12 4 5 2 5 -9 -25 22 9 8
and array index is started from 0. Given two indices are 3 and 8. So, the minimum element is -22.

Program

```
#include<stdio.h>
#include<math.h>
#include<limits.h>


int main() {


   int n;


   printf("Enter the size of the array : ");
   scanf("%d", &n);


   int arr[n];

   printf("Enter the elements of the array : ");
   for(int i=0;i<n;i++) {
      scanf("%d", &arr[i]);
   }




   // preprocessing
```

```
int b = sqrt(n);


int min_arr[n/b+1];


for(int i=0;i<n;i+=b) {
    int min = arr[i];
    for(int j=i+1;j<i+b&&j<n;j++) {
        if(arr[j] < min) {
            min = arr[j];
        }
    }
    min_arr[i/b]=min;
}








//queries

printf("Enter the number of queries : ");
int q;
scanf("%d", &q);

int query[q][2];


printf("Enter the starting and ending index (i j) : ");
for(int i=0;i<q;i++) {


    int a,b;
    scanf("%d %d",&a,&b);
    query[i][0]=a;
    query[i][1]=b;
}




for(int i=0;i<q;i++){
        int x = query[i][0];
        int y = query[i][1];


        int s_block = x/b;


        int e_block = y/b;


        int ans = INT_MAX;


        for(int j=x;j<(s_block+1)*b;j++) {
            if(arr[j]< ans) {
                ans = arr[j];
            }
        }




        for(int j=e_block*b;j<=y;j++) {
            if(arr[j]< ans) {
                ans = arr[j];
            }
        }




        for(int j=s_block+1;j<e_block;j++) {
            if(min_arr[j] < ans) {
                ans = min_arr[j];
            }
        }


        printf("%d\n",ans);

    }


    return 0;
}
```

Algorithm

1. Divide the array into partitions of each ‘b’ length.
2. Compute the minimum of each partition and store it in an array.
3. For every query of type (i,j) where i is the starting index and j is the end index, the minimum element will be minimum of
    a. Starting partition elements (maximum b elements),
    b. Minimum element of partitions between the given indices (maximum n/b partitions),
    c. Ending partition elements (maximum b elements).

Time Complexity

Time Complexity of above program will be

1. PreProcessing the input : O(n)
2. Query : For 1 query it will be : O(b + n/b)

$$\frac{d(b+n/b)}{db} = 0$$

$$2 - \frac{n}{b^2} = 0$$

$$b = \sqrt{n}$$

3. Query : For 1 query it will be : O($\sqrt{n}$)