

Competitive Programming

Week5.10_Assignment-3(Wednesday)

Week 5-10 Exploring Applications of Advanced Tree Data

Code:

```
Real-Time Minimum and Maximum Query Using Segment
import java.util.*;

public class Main
{
    static int[] minTree, maxTree, arr;

    static int n;

    static void build(int idx, int l, int r)
    {
        if (l == r)
        {
            minTree[idx] = arr[l];
            maxTree[idx] = arr[l];
            return;
        }

        int mid = (l + r) / 2;

        build(2 * idx + 1, l, mid);

        build(2 * idx + 2, mid + 1, r);

        minTree[idx] = Math.min(minTree[2 * idx + 1], minTree[2 * idx + 2]);
        maxTree[idx] = Math.max(maxTree[2 * idx + 1], maxTree[2 * idx + 2]);
    }

    static int rangeMin(int idx, int l, int r, int ql, int qr)
    {
        if (qr < l || ql > r)
            return Integer.MAX_VALUE;

        if (ql <= l && r <= qr)
```

```

        return minTree[idx];
    int mid = (l + r) / 2;
    return Math.min(
        rangeMin(2 * idx + 1, l, mid, ql, qr),
        rangeMin(2 * idx + 2, mid + 1, r, ql, qr)
    );
}

static int rangeMax(int idx, int l, int r, int ql, int qr)
{
    if (qr < l || ql > r)
        return Integer.MIN_VALUE;
    if (ql <= l && r <= qr)
        return maxTree[idx];
    int mid = (l + r) / 2;
    return Math.max(
        rangeMax(2 * idx + 1, l, mid, ql, qr),
        rangeMax(2 * idx + 2, mid + 1, r, ql, qr)
    );
}

static void update(int idx, int l, int r, int pos, int val)
{
    if (l == r)
    {
        minTree[idx] = val;
        maxTree[idx] = val;
        return;
    }
    int mid = (l + r) / 2;

```

```

    if (pos <= mid)
        update(2 * idx + 1, l, mid, pos, val);
    else
        update(2 * idx + 2, mid + 1, r, pos, val);

    minTree[idx] = Math.min(minTree[2 * idx + 1], minTree[2 * idx + 2]);
    maxTree[idx] = Math.max(maxTree[2 * idx + 1], maxTree[2 * idx + 2]);
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter number of sensors: ");

    n = sc.nextInt();

    arr = new int[n];

    System.out.println("Enter temperature readings:");

    for (int i = 0; i < n; i++)
        arr[i] = sc.nextInt();

    minTree = new int[4 * n];
    maxTree = new int[4 * n];

    build(0, 0, n - 1);

    System.out.print("Enter left and right for RangeMin query: ");

    int l1 = sc.nextInt();

    int r1 = sc.nextInt();

    System.out.println(rangeMin(0, 0, n - 1, l1, r1));

    System.out.print("Enter left and right for RangeMax query: ");

    int l2 = sc.nextInt();

    int r2 = sc.nextInt();

    System.out.println(rangeMax(0, 0, n - 1, l2, r2));
}

```

```
System.out.print("Enter index and new value for Update: ");
```

```
int pos = sc.nextInt();
```

```
int val = sc.nextInt();
```

```
update(0, 0, n - 1, pos, val);
```

```
System.out.print("Enter left and right for RangeMin query after update: ");
```

```
int l3 = sc.nextInt();
```

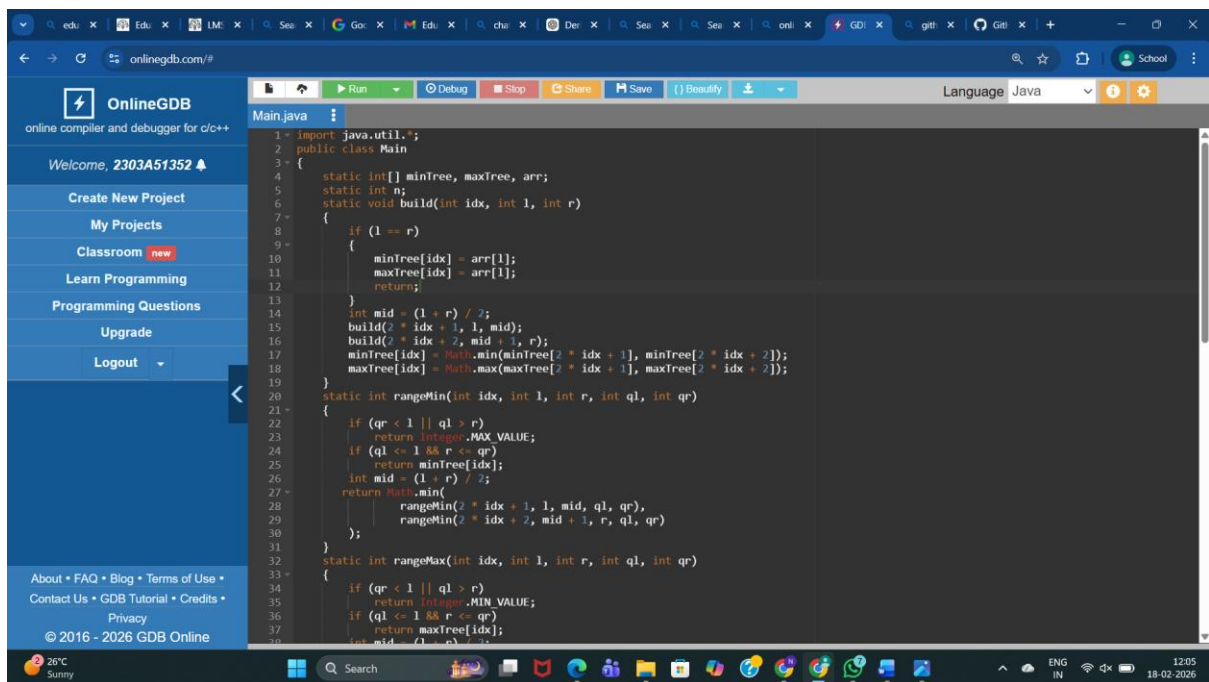
```
int r3 = sc.nextInt();
```

```
System.out.println(rangeMin(0, 0, n - 1, l3, r3));
```

```
sc.close();
```

```
}
```

```
}
```



The screenshot shows a web browser window with the OnlineGDB website. The main content area displays a Java program for a segment tree. The code includes a `build` method to construct the tree and `rangeMin` and `rangeMax` methods for querying. The `build` method recursively divides the array into segments, storing the minimum and maximum values for each segment in `minTree` and `maxTree` arrays. The `rangeMin` method uses a recursive approach to find the minimum value in a given range, while `rangeMax` finds the maximum. The `main` class contains the `main` method where the tree is built and queries are performed.

```
1 import java.util.*;
2 public class Main
3 {
4     static int[] minTree, maxTree, arr;
5     static int n;
6     static void build(int idx, int l, int r)
7     {
8         if (l == r)
9         {
10             minTree[idx] = arr[l];
11             maxTree[idx] = arr[l];
12             return;
13         }
14         int mid = (l + r) / 2;
15         build(2 * idx + 1, l, mid);
16         build(2 * idx + 2, mid + 1, r);
17         minTree[idx] = Math.min(minTree[2 * idx + 1], minTree[2 * idx + 2]);
18         maxTree[idx] = Math.max(maxTree[2 * idx + 1], maxTree[2 * idx + 2]);
19     }
20     static int rangeMin(int idx, int l, int r, int ql, int qr)
21     {
22         if (qr < l || ql > r)
23             return Integer.MAX_VALUE;
24         if (ql <= l && r <= qr)
25             return minTree[idx];
26         int mid = (l + r) / 2;
27         return Math.min(
28             rangeMin(2 * idx + 1, l, mid, ql, qr),
29             rangeMin(2 * idx + 2, mid + 1, r, ql, qr)
30         );
31     }
32     static int rangeMax(int idx, int l, int r, int ql, int qr)
33     {
34         if (qr < l || ql > r)
35             return Integer.MIN_VALUE;
36         if (ql <= l && r <= qr)
37             return maxTree[idx];
38         int mid = (l + r) / 2;
39         return Math.max(
40             rangeMax(2 * idx + 1, l, mid, ql, qr),
41             rangeMax(2 * idx + 2, mid + 1, r, ql, qr)
42         );
43     }
44 }
```

OnlineGDB
online compiler and debugger for c/c++

Welcome, 2303A51352

Create New Project
My Projects
Classroom **new**
Learn Programming
Programming Questions
Upgrade
Logout

About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy
© 2016 - 2026 GDB Online

```
36 if (ql <= 1 && r <= qr)
37     return maxTree[idx];
38 int mid = (l + r) / 2;
39 return Math.max(
40     rangeMax(2 * idx + 1, l, mid, ql, qr),
41     rangeMax(2 * idx + 2, mid + 1, r, ql, qr)
42 );
43 }
44 static void update(int idx, int l, int r, int pos, int val)
45 {
46     if (l == r)
47     {
48         minTree[idx] = val;
49         maxTree[idx] = val;
50         return;
51     }
52     int mid = (l + r) / 2;
53     if (pos <= mid)
54         update(2 * idx + 1, l, mid, pos, val);
55     else
56         update(2 * idx + 2, mid + 1, r, pos, val);
57     minTree[idx] = Math.min(minTree[2 * idx + 1], minTree[2 * idx + 2]);
58     maxTree[idx] = Math.max(maxTree[2 * idx + 1], maxTree[2 * idx + 2]);
59 }
60 public static void main(String[] args)
61 {
62     Scanner sc = new Scanner(System.in);
63     System.out.print("Enter number of sensors: ");
64     n = sc.nextInt();
65     arr = new int[n];
66     System.out.println("Enter temperature readings:");
67     for (int i = 0; i < n; i++)
68         arr[i] = sc.nextInt();
69     minTree = new int[4 * n];
70     maxTree = new int[4 * n];
71     build(0, 0, n - 1);
72     System.out.print("Enter left and right for RangeMin query: ");
```

```
54         update(2 * idx + 1, l, mid, pos, val);
55     else
56         update(2 * idx + 2, mid + 1, r, pos, val);
57     minTree[idx] = Math.min(minTree[2 * idx + 1], minTree[2 * idx + 2]);
58     maxTree[idx] = Math.max(maxTree[2 * idx + 1], maxTree[2 * idx + 2]);
59 }
60 public static void main(String[] args)
61 {
62     Scanner sc = new Scanner(System.in);
63     System.out.print("Enter number of sensors: ");
64     n = sc.nextInt();
65     arr = new int[n];
66     System.out.println("Enter temperature readings:");
67     for (int i = 0; i < n; i++)
68         arr[i] = sc.nextInt();
69     minTree = new int[4 * n];
70     maxTree = new int[4 * n];
71     build(0, 0, n - 1);
72     System.out.print("Enter left and right for RangeMin query: ");
73     int l1 = sc.nextInt();
74     int r1 = sc.nextInt();
75     System.out.println(rangeMin(0, 0, n - 1, l1, r1));
76     System.out.print("Enter left and right for RangeMax query: ");
77     int l2 = sc.nextInt();
78     int r2 = sc.nextInt();
79     System.out.println(rangeMax(0, 0, n - 1, l2, r2));
80     System.out.print("Enter index and new value for Update: ");
81     int pos = sc.nextInt();
82     int val = sc.nextInt();
83     update(0, 0, n - 1, pos, val);
84     System.out.print("Enter left and right for RangeMin query after update: ");
85     int l3 = sc.nextInt();
86     int r3 = sc.nextInt();
87     System.out.println(rangeMin(0, 0, n - 1, l3, r3));
88     sc.close();
89 }
90 }
91 }
```

OnlineGDB
online compiler and debugger for c/c++

Welcome, 2303A51352

Create New Project

My Projects

Classroom new

Learn Programming

Programming Questions

Upgrade

Logout

About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy
© 2016 - 2026 GDB Online

Run Debug Stop Share Save Beautify

Language Java

Main.java
54 update(2 * idx + 1, mid, pos, val);
55 else
56 update(2 * idx + 2, mid + 1, r, pos, val);
57
58 minTree[idx] = Math.min(minTree[2 * idx + 1], minTree[2 * idx + 2]);
59 maxTree[idx] = Math.max(maxTree[2 * idx + 1], maxTree[2 * idx + 2]);
60
61 }
62
63 public static void main(String[] args)
64 {
65 Scanner sc = new Scanner(System.in);
66 System.out.print("Enter number of sensors: ");
67 n = sc.nextInt();
68 arr = new int[n];
69 System.out.println("Enter temperature readings:");
70 for (int i = 0; i < n; i++)
71 arr[i] = sc.nextInt();
72 minTree = new int[4 * n];
73 maxTree = new int[4 * n];
74 build(0, 0, n - 1);
75
76 }
77
78 }

input
Enter number of sensors: 8
Enter temperature readings:
32 28 30 35 29 31 34 33
Enter left and right for RangeMin query: 2 6
29
Enter left and right for RangeMax query: 1 5
35
Enter index and new value for Update: 3 27
Enter left and right for RangeMin query after update: 2 6
27
...Program finished with exit code 0

26°C Sunny

Search

ENG IN

12:05 18-02-2026