

# COMPITATIVE ASSIGNMENT

Name:B.Harshini

-HT NO:2303A52242

BT NO:36

## Assignment 1: Segment Tree for Range Sum Query

Problem Statement:

You are given an array of N integers. Build a Segment Tree to efficiently answer multiple range sum queries.

Input Format

- The first line contains an integer T, the number of test cases.
- For each test case:
  - The first line contains an integer N.
  - The second line contains N integers.
  - The next line contains an integer Q, the number of queries.
  - Each of the next Q lines contains two integers L and R.

Output Format

For each query, print the sum of elements in the range [L, R].

Constraints

- $1 \leq T \leq 20$
- $1 \leq N \leq 200000$
- $0 \leq A[i] \leq 10^9$

Sample Input

1

5

1 3 5 7 9

3

0 2

The image shows two screenshots of an online judge platform. The top screenshot displays the source code in a Python file named `main.py`. The code implements a segment tree for range sum queries and updates. The bottom screenshot shows the execution results, including the input data and the output of the program.

**Code (main.py):**

```
1 import sys
2 sys.setrecursionlimit(10**7)
3 def build(node, start, end):
4     if start == end:
5         seg[node] = arr[start]
6     else:
7         mid = (start + end) // 2
8         build(2*node, start, mid)
9         build(2*node + 1, mid + 1, end)
10        seg[node] = seg[2*node] + seg[2*node + 1]
11
12 def query(node, start, end, l, r):
13     if r < start or end < l:
14         return 0
15     if l <= start and end <= r:
16         return seg[node]
17     mid = (start + end) // 2
18     return query(2*node, start, mid, l, r) + \
19            query(2*node + 1, mid + 1, end, l, r)
20 T = int(input())
21 for _ in range(T):
22     N = int(input())
23     arr = list(map(int, input().split()))
24
25 seg = [0] * (4 * N)
26 build(1, 0, N - 1)
27
28 Q = int(input())
29 for _ in range(Q):
30     l, R = map(int, input().split())
31     print(query(1, 0, N - 1, l, R))
32
```

**Execution Results:**

```
1 3 5 7 9
3
0 2
9
1 3
15
2 4
21

...Program finished with exit code 0
Press ENTER to exit console.[]
```

JAVA CODE:

Week-3\_Batch-23,24,45&50 GDB online Debugger | Compiling

onlinegdb.com

Main.java

```
1 import java.util.*;
2
3 public class Main {
4
5     static long[] segTree;
6     static long[] arr;
7
8     static void build(int node, int start, int end) {
9         if (start == end) {
10             segTree[node] = arr[start];
11         } else {
12             int mid = (start + end) / 2;
13             build(2 * node, start, mid);
14             build(2 * node + 1, mid + 1, end);
15             segTree[node] = segTree[2 * node] + segTree[2 * node + 1];
16         }
17     }
18
19     static long query(int node, int start, int end, int l, int r) {
20         if (r < start || end < l)
21             return 0;
22         if (l <= start && end <= r)
23             return segTree[node];
24
25         int mid = (start + end) / 2;
26         return query(2 * node, start, mid, l, r)
27             + query(2 * node + 1, mid + 1, end, l, r);
28     }
29
30     public static void main(String[] args) {
31         Scanner sc = new Scanner(System.in);
32
33         int T = sc.nextInt();
34         while (T-- > 0) {
35             int N = sc.nextInt();
36             for (int i = 0; i < N; i++)
37                 arr[i] = sc.nextInt();
38
39             build(1, 0, N - 1);
40
41             int Q = sc.nextInt();
42             for (int j = 0; j < Q; j++) {
43                 int L = sc.nextInt();
44                 int R = sc.nextInt();
45                 System.out.println(query(1, 0, N - 1, L, R));
46             }
47         }
48     }
49 }
```

close ad [x]

Save 55% on Creative Cloud Pro. Now ₹1,199/mo\* for the first year only. Ends 11 February.

Learn more

Adobe Creative Cloud Pro

Shape it. Fill it. Love it. The new Generative Shape Fill.

Try now

Adobe Illustrator

19°C Sunny ENG IN 06-02-2026 09:21

Week-3\_Batch-23,24,45&50 GDB online Debugger | Compiling

onlinegdb.com

Main.java

```
1 import java.util.*;
2
3 public class Main {
4
5     static long[] segTree;
6     static long[] arr;
7
8     static void build(int node, int start, int end) {
9         if (start == end) {
10             segTree[node] = arr[start];
11         } else {
12             int mid = (start + end) / 2;
13             build(2 * node, start, mid);
14             build(2 * node + 1, mid + 1, end);
15             segTree[node] = segTree[2 * node] + segTree[2 * node + 1];
16         }
17     }
18
19     static long query(int node, int start, int end, int l, int r) {
20         if (r < start || end < l)
21             return 0;
22         if (l <= start && end <= r)
23             return segTree[node];
24
25         int mid = (start + end) / 2;
26         return query(2 * node, start, mid, l, r)
27             + query(2 * node + 1, mid + 1, end, l, r);
28     }
29
30     public static void main(String[] args) {
31         Scanner sc = new Scanner(System.in);
32
33         int T = sc.nextInt();
34         while (T-- > 0) {
35             int N = sc.nextInt();
36             for (int i = 0; i < N; i++)
37                 arr[i] = sc.nextInt();
38
39             build(1, 0, N - 1);
40
41             int Q = sc.nextInt();
42             for (int j = 0; j < Q; j++) {
43                 int L = sc.nextInt();
44                 int R = sc.nextInt();
45                 System.out.println(query(1, 0, N - 1, L, R));
46             }
47         }
48     }
49 }
```

input

```
1 3 5 7 9
3 0 2
9
1 3
15
2 4
21

...Program finished with exit code 0
Press ENTER to exit console.
```

19°C Sunny ENG IN 06-02-2026 09:22