Institute Of Technology, Nirma university

BRANCH :- **Computer Science Engineering**

PRACTICAL SUBMISSION

|*|*STUDENT INFO*|*|

Name :- **Pratik Kansara**

Roll No. :- **20BCE510**

Division :- **E4**

|*|*SUBJECT INFO*|*|

Subject :- **Advanced Data Structures**

Practical No. :- **6**

Practical – 6

AIM :- Segment trees are useful to find range sum of a given interval. Write a program to demonstrate usage of segment tree structure to find range sum of numbers in a given range. Example: Given

Index 0 1 2 3 4 5 6 7 8 9

Data 2 3 4 5 6 7 8 9 10 11

Sum(0,4) = 20

Sum (2,6) = 30

Code:

STNode.java

```java
public class STNode {
    int sum;
    int start, end;
    STNode left;
    STNode right;

    public STNode(int l, int r, int s) {
        start = l;
        end = r;
        sum = s;
    }
}
```

SegmentTree.java

```
public class SegmentTree {
    public STNode buildSegmentTree(int Arr[], int l, int r) {
        if (l == r) {
            STNode node = new STNode(l, r, Arr[l]);
            return node;
        }
        int mid = (l + r) / 2;
        STNode leftNode = buildSegmentTree(Arr, l, mid);
        STNode rightNode = buildSegmentTree(Arr, mid + 1, r);
        STNode root = new STNode(leftNode.start, rightNode.end, leftNode.sum +
rightNode.sum);
        root.left = leftNode;
        root.right = rightNode;

        return root;
    }

    public int getQuerySum(STNode root, int l, int r) {
        if (root.end < l || root.start > r) {
            return 0;
        }
        if (root.start >= l && root.end <= r) {
            return root.sum;
        }
        return getQuerySum(root.left, l, r) + getQuerySum(root.right, l, r);
    }

    public void preOrd(STNode root) {
        if (root != null) {
            System.out.print(root.sum + " ");
            preOrd(root.left);
            preOrd(root.right);
        }
    }

}
```

STMain.java

```
import java.util.Scanner;

public class STMain {
    public static void main(String[] args) {
        // int arr[] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11}; // static array
according to
        // defination

        SegmentTree sgt = new SegmentTree();

        int a[];
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Size of the Array : ");
        int n = sc.nextInt();

        a = new int[n];
        System.out.println("Enter array elements : ");
        for (int i = 0; i < a.length; i++) {
            a[i] = sc.nextInt();
        }

        STNode root = sgt.buildSegmentTree(a, 0, n - 1);

        while (true) {
            System.out.println("1. For sum range queries");
            System.out.println("2. exit");

            int ch = sc.nextInt();

            switch (ch) {
                case 1:
                    int left = sc.nextInt();
                    int right = sc.nextInt();
                    System.out.println("Sum : (" + left + ", " + right + ") : "
+ sgt.getQuerySum(root, left, right));
                    break;
                case 2:
                    System.exit(0);
            }
        }

    }
}
```

OUTPUT

```
Enter the Size of the Array : 8
Enter array elements :
1 2 3 4 5 6 7 8
1. For sum range queries
2. exit
1
2
5
Sum : (2, 5) : 18
```

```
1. For sum range queries
2. exit
1
1
7
Sum : (1, 7) : 35
1. For sum range queries
2. exit
1
0
7
Sum : (0, 7) : 36
```