Indian Institute of Technology Roorkee
Department of Computer Science and Engineering

CSN-261: Data Structures Laboratory
(Autumn 2019-2020)
Lab Assignment-7 (L7)

Name :- Harshit Verma **Enroll no:- 18114029**
Branch :- CSE **Sub Batch :- O2**

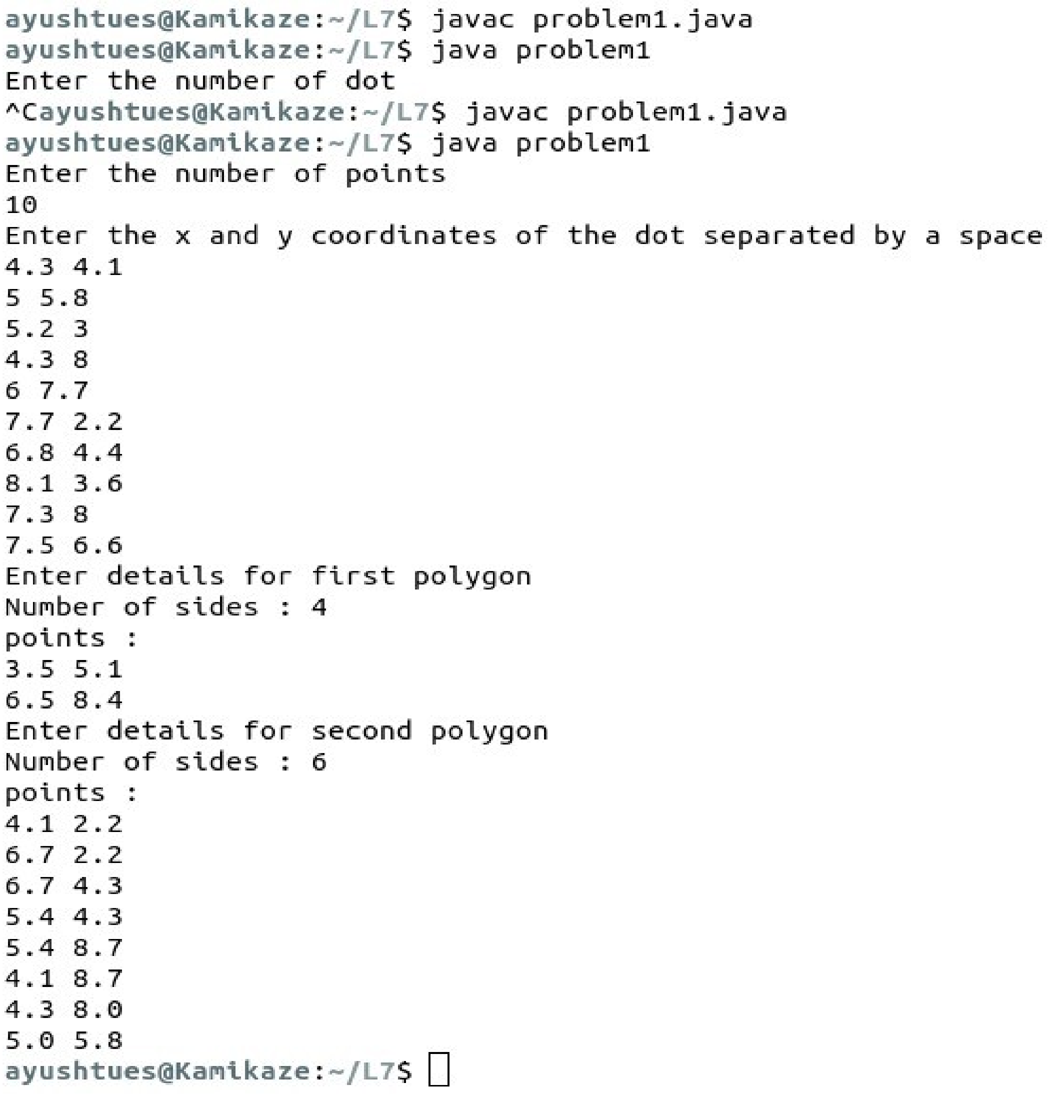
Problem Statement 1:

Given: n 2D points and two orthogonal polygons.
Problem: Find the set of points lie inside the overlapping region (rectangular) of the two given orthogonal
polygons.
Write a program in Java to solve the above problem applying k-d tree data structure.

Data-Structure: K-D Tree.

Algorithm: Divide and conqour, Line sweeping.

Screenshots:



```
ayushtues@Kamikaze:~/L7$ javac problem1.java
ayushtues@Kamikaze:~/L7$ java problem1
Enter the number of dot
^Cayushtues@Kamikaze:~/L7$ javac problem1.java
ayushtues@Kamikaze:~/L7$ java problem1
Enter the number of points
10
Enter the x and y coordinates of the dot separated by a space
4.3 4.1
5 5.8
5.2 3
4.3 8
6 7.7
7.7 2.2
6.8 4.4
8.1 3.6
7.3 8
7.5 6.6
Enter details for first polygon
Number of sides : 4
points :
3.5 5.1
6.5 8.4
Enter details for second polygon
Number of sides : 6
points :
4.1 2.2
6.7 2.2
6.7 4.3
5.4 4.3
5.4 8.7
4.1 8.7
4.3 8.0
5.0 5.8
ayushtues@Kamikaze:~/L7$ 
```

Problem Statement 2:

-

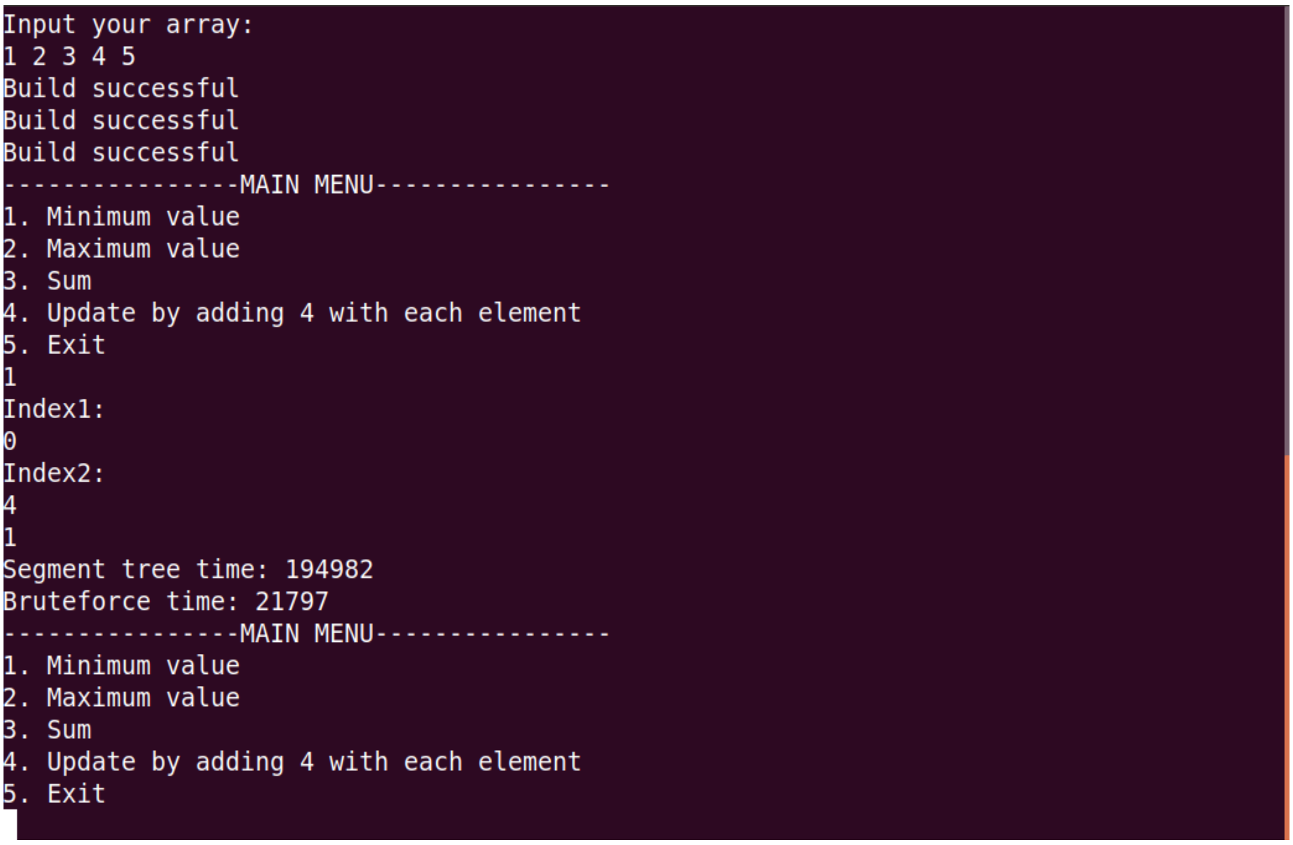
Given n values in an array and two index values, find the result of the following queries 1. minimum value 2. maximum value 3. sum 4. update by adding 4 with each element, within the given index range using Segment tree.
Also implement the brute-force method and compare the execution time of both the methods.

A segment tree is a data structure used for storing information about intervals, range or segments. It facilitates efficient range querying in O(log n), where n is the size of the given problem.

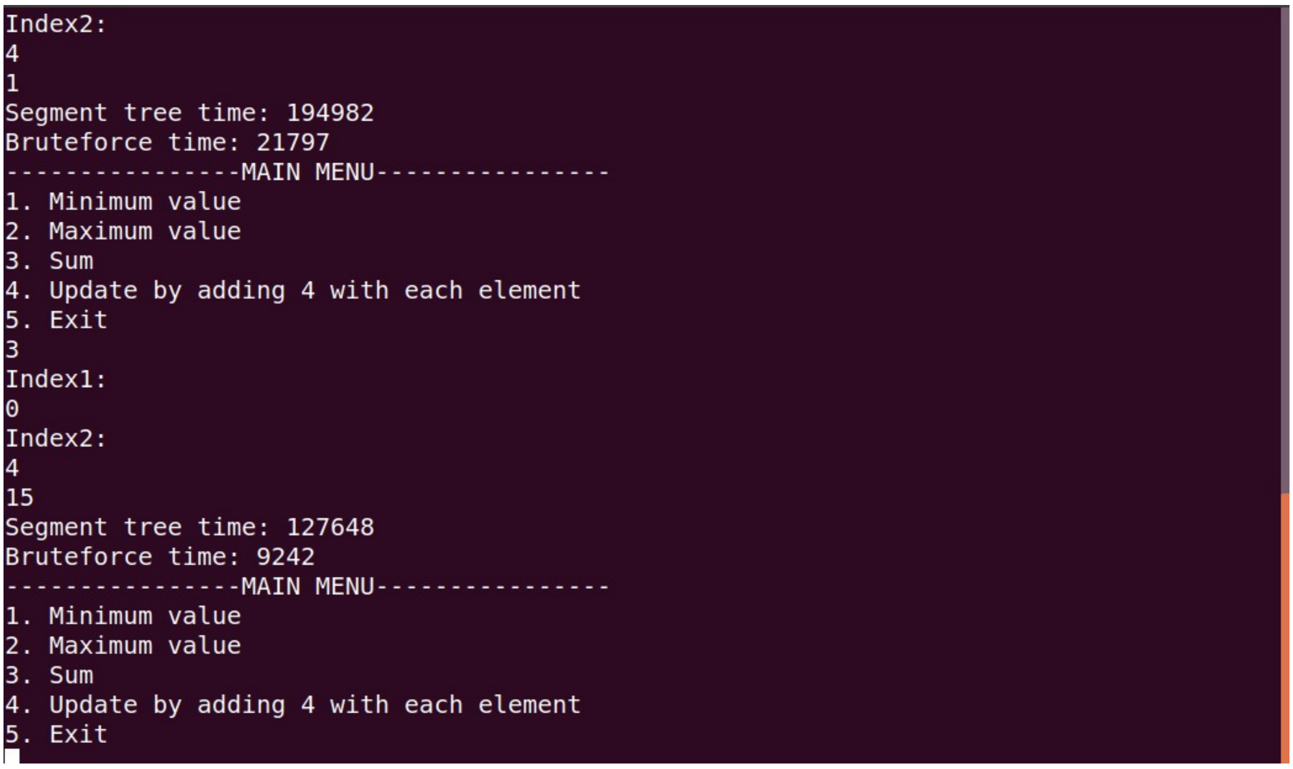
Data-Structure: Segment Trees.

Algorithm: Divide and conqour.

Screenshots:



```
Input your array:
1 2 3 4 5
Build successful
Build successful
Build successful
---------------MAIN MENU---------------
1. Minimum value
2. Maximum value
3. Sum
4. Update by adding 4 with each element
5. Exit
1
Index1:
0
Index2:
4
1
Segment tree time: 194982
Bruteforce time: 21797
---------------MAIN MENU---------------
1. Minimum value
2. Maximum value
3. Sum
4. Update by adding 4 with each element
5. Exit
```



```
Index2:
4
1
Segment tree time: 194982
Bruteforce time: 21797
----------------MAIN MENU----------------
1. Minimum value
2. Maximum value
3. Sum
4. Update by adding 4 with each element
5. Exit
3
Index1:
0
Index2:
4
15
Segment tree time: 127648
Bruteforce time: 9242
----------------MAIN MENU----------------
1. Minimum value
2. Maximum value
3. Sum
4. Update by adding 4 with each element
5. Exit
```