

OS Assignment - 9

NAME : SOURABH PATEL

ADMISSION NO : U19CS082

1. To implement Shortest Seek Time First (SSTF) Disk Scheduling Algorithm.

CODE:

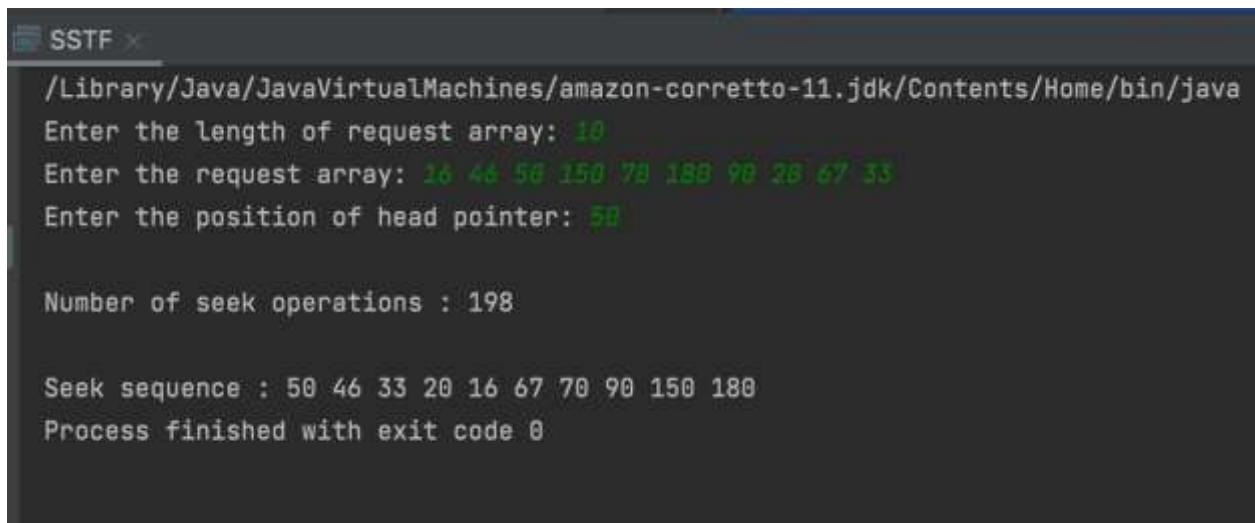
```
import java.util.*;
public class SSTF {
    static class Node{
        int value;
        boolean accessed;
        public Node(int val){
            this.value = val;
            accessed = false;
        }
    }
    public static int findInd(ArrayList<Node> request, int head){
        int min = Integer.MAX_VALUE, index=-1, i;
        for (i = 0; i < request.size(); i++) {
            int diff = Math.abs(request.get(i).value - head);
            if (!request.get(i).accessed && min>diff){
                min = diff;
                index = i;
            }
        }
        return index;
    }
    public static void shortestSeekTimeFirst(ArrayList<Node> request, int head){
        if (request.size()==0){
            System.out.println("Request array is empty");
            return;
        }
        int seek_count = 0;
        int[] seek_sequence = new int[request.size() + 1];
        for (int i = 0; i < request.size(); i++) {
            seek_sequence[i] = head;
            int index = findInd(request, head);
            request.get(index).accessed = true;
        }
    }
}
```

```

        seek_count += Math.abs(request.get(index).value - head);
        head = request.get(index).value;
    }
    seek_sequence[request.size()] = head;
    System.out.println("\nNumber of seek operations : "+seek_count);
    System.out.print("\nSeek sequence : ");
    for (int i = 1; i < seek_sequence.length; i++) {
        System.out.print(seek_sequence[i] + " ");
    }
}
static Scanner in = new Scanner(System.in);
public static void main(String[] args) {
    ArrayList<Node> arr = new ArrayList<>();
    System.out.print("Enter the length of request array: ");
    int n = in.nextInt();
    System.out.print("Enter the request array: ");
    for (int i = 0; i < n; i++) {
        Node temp = new Node(in.nextInt());
        arr.add(temp);
    }
    System.out.print("Enter the position of head pointer: ");
    int head = in.nextInt();
    shortestSeekTimeFirst(arr, head);
}
}

```

OUTPUT:



```

SSTF x
/Library/Java/JavaVirtualMachines/amazon-corretto-11.jdk/Contents/Home/bin/java
Enter the length of request array: 10
Enter the request array: 16 46 50 150 78 188 90 20 67 33
Enter the position of head pointer: 50

Number of seek operations : 198

Seek sequence : 50 46 33 20 16 67 70 90 150 180
Process finished with exit code 0

```

2. To implement the SCAN algorithm for Disk Scheduling.

CODE:

```
import java.util.*;
public class SCAN {
    static int disk_size = 200;
    static String right = "right";
    static String left = "left";
    public static void scanF(ArrayList<Integer> request, int head, String dir){
        int seek_count = 0;
        ArrayList<Integer> l = new ArrayList<>(), r = new ArrayList<>(),
        seek_sequence = new ArrayList<>();
        if (dir.equals(left)){
            l.add(0);
        }else{
            r.add(disk_size-1);
        }
        for (int i = 0; i < request.size(); i++) {
            if (request.get(i) <= head) l.add(request.get(i));
            else if (request.get(i) > head) r.add(request.get(i));
        }
        Collections.sort(l);
        Collections.sort(r);
        int run = 2;
        while (run-->0){
            if (dir.equals(left)){
                for (int i = l.size()-1; i >= 0 ; i--) {
                    int curr = l.get(i);
                    seek_sequence.add(curr);
                    int diff = Math.abs(curr-head);
                    seek_count+=diff;
                    head=curr;
                }
                dir = "right";
            }
            else{
                for (int i = 0; i < r.size(); i++) {
                    int curr = r.get(i);
                    seek_sequence.add(curr);
                    int diff = Math.abs(curr-head);
                    seek_count+=diff;
                    head=curr;
                }
                dir = "left";
            }
        }
    }
}
```

```

        System.out.println("\nNumber of seek operations : "+seek_count);
        System.out.print("\nSeek sequence : ");
        for (int i = 0; i < seek_sequence.size(); i++) {
            System.out.print(seek_sequence.get(i)+" ");
        }
    }
}

public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    ArrayList<Integer> arr = new ArrayList<>();
    System.out.print("Enter the length of request array: ");
    int n = s.nextInt();
    System.out.print("Enter the request array: ");
    for (int i = 0; i < n; i++) {
        arr.add(s.nextInt());
    }
    System.out.print("Enter the position of head pointer: ");
    int head = s.nextInt();
    System.out.print("Enter the direction: ");
    String str = s.next();
    if (!str.equals(right) && !str.equals(left)){
        System.out.println("Incorrect direction!");
        System.exit(1);
    }
    scanF(arr,head,str);
}
}

```

OUTPUT:

```

SCAN x
/Library/Java/JavaVirtualMachines/amazon-corretto-11.jdk/Contents/Home/bin/java
Enter the length of request array: 10
Enter the request array: 16 20 50 70 150 80 46 180 33 67
Enter the position of head pointer: 50
Enter the direction: Left

Number of seek operations : 230

Seek sequence : 50 46 33 20 16 0 67 70 80 150 180
Process finished with exit code 0

```