

Design analysis and algorithm

Lab 8

Divide And Conquer online\_bet

Name: Naomi George

Slot: L29

Reg No: 19bce7572

Course code: CSE3004

Naïve\_Online\_Bet:

```
import java.util.Scanner;
public class main {
static int[] naivecount(int[] starts, int[] ends, int[] points) {
int[] cnt = new int[points.length];
for (int i = 0; i < points.length; i++) {
for (int j = 0; j < starts.length; j++) {
if (starts[j] <= points[i] && points[i] <= ends[j]) {
cnt[i]++;
}
}
}
return cnt;
}
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
int n, m;
n = scanner.nextInt();
m = scanner.nextInt();
int[] starts = new int[n];
int[] ends = new int[n];
int[] points = new int[m];
for (int i = 0; i < n; i++) {
starts[i] = scanner.nextInt();
ends[i] = scanner.nextInt();
}
for (int i = 0; i < m; i++) {
points[i] = scanner.nextInt();
}
int[] cnt = naivecount(starts, ends, points);
for (int x : cnt) {
System.out.print(x + " ");
}
}
}}
```

Output:

**Result**  
**compiled and executed in 11.962 sec(s)**

```
2 3
0 5
7 10
16 11
1 0 0 |
```

Analysis:

Scanning for each point(p) if present in each segment(s).

For(i=0;i<p;i++) runs p times

```
{  
    For(j=0;j<s;j++) runs n times  
    {  
        If(P[i]>=S[j][0] && P[i]<=S[j][1])  
        C++;  
    }  
    System.out.print(c + " ");  
    C=0;  
}
```

Therefore, Time complexity is  $O(s*p)$

i.e.  $O(n^2)$