Competitive Programming Lab - 8

Academic year: 2020-2021 **Semester:** Long Sem

Faculty Name: Dr. Ajith Jublison sir **Date:** 9/ 7/ 2022

Student name: Taran Mamidala **Reg. no.:** 19BCE7346

Hotel :

Guess why this picture is given ? This is associated with either dynamic or greedy programming?
A few questions are given which could be solved under either of the cases.

Q1.) Assume the series as (1,1,3,4,8,11,21,29,55,.....)

Which programming principle will you apply to solve the summation of the series ?

```
n1=n2=1
X= Yn-1+Xn-2  n > 2
Y1=Y2 =2
Yn=Xn-1+Yn-2 n >2
```

Explanation

To solve the summation of the series we need Dynamic Programming Because the computation of already solved sub-question
will be stored in a matrix and can be retrieved for further calculation.

Dynamic Programming is mainly an optimization over plain recursion. Wherever we see a recursive solution that has repeated calls for same inputs, we can optimise it using Dynamic Programming. The idea is to simply store the results of subproblems, so that we do not have to re-compute them when needed later. This simple optimization reduces time complexities from exponential to polynomial.

Minimum number of rooms to construct problem :

Q2.) Sampath is considering going into business for himself. He's chosen to launch his career in the hotel industry. A list with each guest's arrival time and length of stay at his hotel has been supplied to him. He may only accept one visitor each room. He wants to determine the bare minimum of rooms he needs to construct in order to accommodate all of the guests because he does not want to let them down. Help him discover the solution.

Sample Input

3 (No of guests)
1 2 3 (Arrival time of the guests those who come to the hotel)
3 3 3 (Duration of the guests)

Output 1

3 rooms are required

Input 2

5 (No of guests)
1 2 3 4 5 (Arrival time of the guests those who come to the hotel)
2 3 4 5 6 (Duration of the guests)

Output 2

3 rooms are required.

TIME TO THINK:

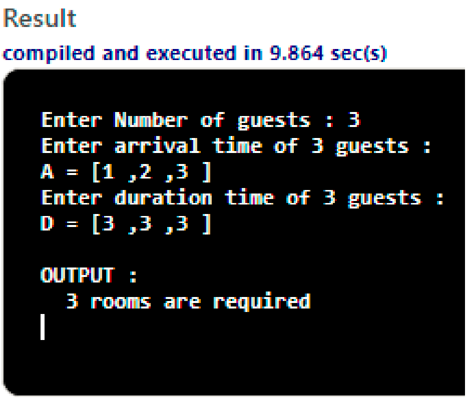
1. What concept have you applied and how have you solved the problem?
2. Find the Time complexity

CODE:

```
import java.util.Scanner;

public class minHotelRooms {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n;
    int results;

    System.out.print("Enter Number of guests : ");
    n = sc.nextInt();

    long[] arrtime = new long[n];
    long[] durtime = new long[n];
    System.out.print("\nEnter arrival time of " + n + " guests : \nA = [");
    arrtime[0] = sc.nextLong();
    for (int i = 1; i < n; i++) {
        System.out.print(",");
        arrtime[i] = sc.nextLong();
    }
    System.out.print("]\n");
    System.out.print("Enter duration time of " + n + " guests : \nD = [");
    durtime[0] = sc.nextLong();
    for (int i = 1; i < n; i++) {
        System.out.print(",");
        durtime[i] = sc.nextLong();
    }
    System.out.print("]\n");

    int[] guestindex = new int[n];
    int guestcount = 0, roomcount = 0;

    for (int i = 0; i < n; i++) {

        if (!isallottedroom(i, guestindex, guestcount)) {

            long guestdurtime = arrtime[i] + durtime[i];
            guestindex[guestcount] = i;

            guestcount++;
            roomcount++;

            for (int j = i + 1; j < n; j++) {
```

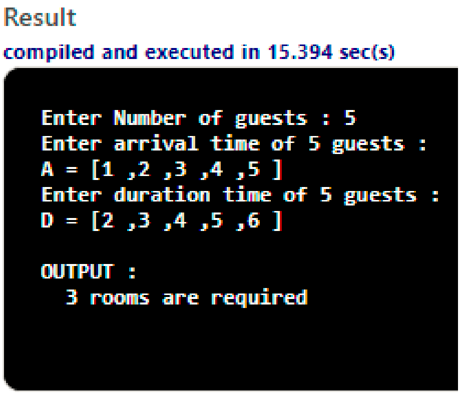
```
            if (guestdurtime == arrtime[j]) {

                guestdurtime = arrtime[j] + durtime[j];
                guestindex[guestcount] = j;

                guestcount++;
            }
            }
        }
    }

    results = roomcount;
    System.out.println("\nOUTPUT :\n  " + results + " rooms are required");
    }

public static boolean isallottedroom(int index, int[] guestindex, int guestcount)
{
    for (int i = 0; i < guestcount; i++) {
        if (guestindex[i] == index) {
            return true;
        }
    }
    return false;
    }
}
```

Output:



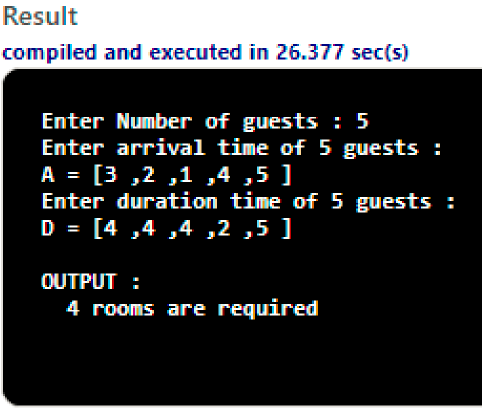
Result
compiled and executed in 9.864 sec(s)

```
Enter Number of guests : 3
Enter arrival time of 3 guests :
A = [1 ,2 ,3 ]
Enter duration time of 3 guests :
D = [3 ,3 ,3 ]

OUTPUT :
  3 rooms are required
```



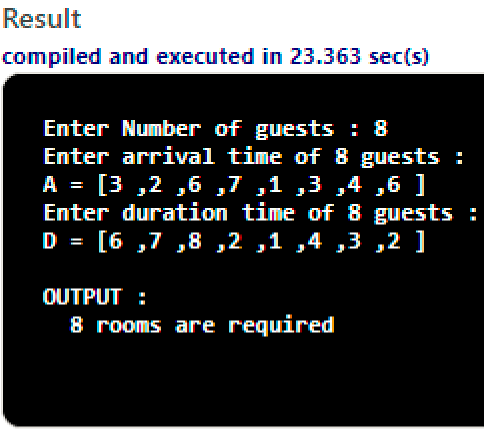
Result
compiled and executed in 15.394 sec(s)

```
Enter Number of guests : 5
Enter arrival time of 5 guests :
A = [1 ,2 ,3 ,4 ,5 ]
Enter duration time of 5 guests :
D = [2 ,3 ,4 ,5 ,6 ]

OUTPUT :
  3 rooms are required
```



Result
compiled and executed in 26.377 sec(s)

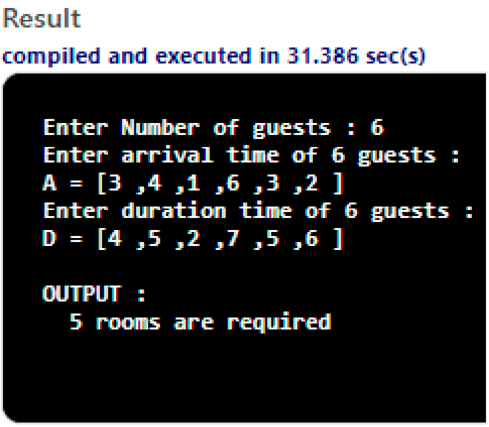
```
Enter Number of guests : 5
Enter arrival time of 5 guests :
A = [3 ,2 ,1 ,4 ,5 ]
Enter duration time of 5 guests :
D = [4 ,4 ,4 ,2 ,5 ]

OUTPUT :
  4 rooms are required
```



Result

compiled and executed in 23.363 sec(s)

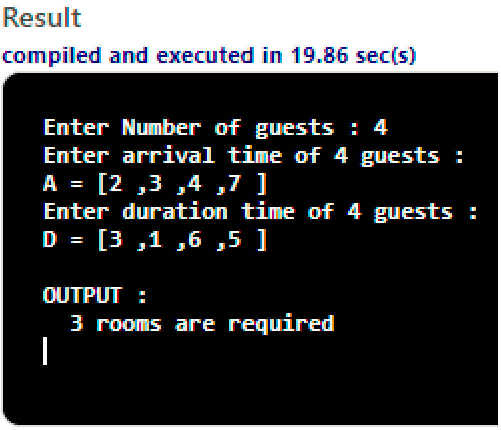
```
Enter Number of guests : 8
Enter arrival time of 8 guests :
A = [3 ,2 ,6 ,7 ,1 ,3 ,4 ,6 ]
Enter duration time of 8 guests :
D = [6 ,7 ,8 ,2 ,1 ,4 ,3 ,2 ]

OUTPUT :
  8 rooms are required
```



Result

compiled and executed in 31.386 sec(s)

```
Enter Number of guests : 6
Enter arrival time of 6 guests :
A = [3 ,4 ,1 ,6 ,3 ,2 ]
Enter duration time of 6 guests :
D = [4 ,5 ,2 ,7 ,5 ,6 ]

OUTPUT :
  5 rooms are required
```



Result

compiled and executed in 19.86 sec(s)

```
Enter Number of guests : 4
Enter arrival time of 4 guests :
A = [2 ,3 ,4 ,7 ]
Enter duration time of 4 guests :
D = [3 ,1 ,6 ,5 ]

OUTPUT :
  3 rooms are required
```

Q3.) Assume there are 2 arrays. A= {10,2,50,60,100} and B ={10,30,20,400,1}

Compare both A and B and find the total points.

If A[i] > B[i] then A[i] gets 1

If A[i] < B[i], then B[i] gets 1

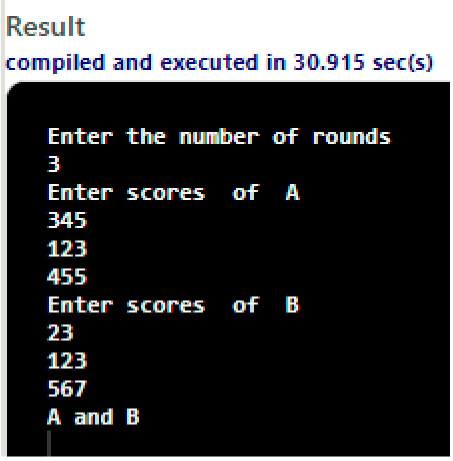
If A[i] == B[i] then both get 1 .

Identify who wins the game by computing the total points earned.

CODE:

```
import java.util.*;
public class Main {

  public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);

     System.out.println("Enter the number of games");
     int n = sc.nextInt();

     int A[] = new int[n];
     int B[] = new int[n];
     System.out.println("Enter total points  of  A ");

     for (int i = 0; i < n; i++) {
     A[i] = sc.nextInt();
     }
System.out.println("Enter total points  of  B ");

for (int i = 0; i < n; i++) {
B[i] = sc.nextInt();
}

winner(A, B, 0, 0, 0);
}

static void winner(int a[], int b[], int i, int countA, int countB) {

      if (i == a.length){

      if (countA > countB) {
      System.out.println("A");
           }
      else if (countA < countB) {
      System.out.println("B");
      }
           else {
           System.out.println("A and B");
           }
      return;
      }
      else if (i < a.length) {

           if (a[i] > b[i]) {
           winner(a, b, i + 1, countA + 1, countB);
           }
           else if (a[i] < b[i]) {
           winner(a, b, i + 1, countA, countB + 1);
           }
           else {
                winner(a, b, i + 1, countA + 1, countB + 1);

           }
         }
      }
}
```

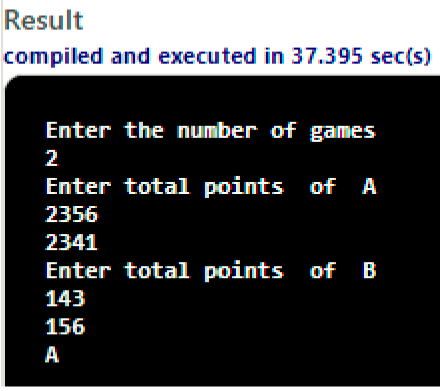
Output:



Result

compiled and executed in 30.915 sec(s)

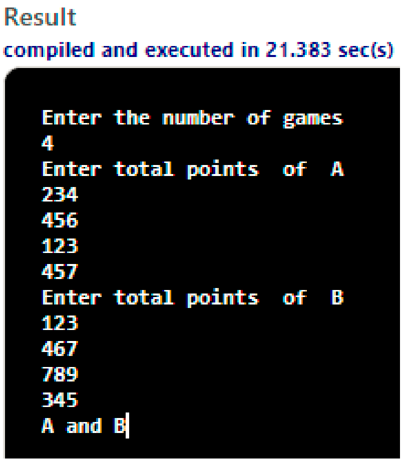
```
Enter the number of rounds
3
Enter scores  of  A
345
123
455
Enter scores  of  B
23
123
567
A and B
```



Result

compiled and executed in 37.395 sec(s)

```
Enter the number of games
2
Enter total points  of  A
2356
2341
Enter total points  of  B
143
156
A
```



Result

compiled and executed in 21.383 sec(s)

```
Enter the number of games
4
Enter total points  of  A
234
456
123
457
Enter total points  of  B
123
467
789
345
A and B
```