CSE3032 - Competitive Programming
WIN SEM (2022-2023) AMR
Class Number: AP2022236001007
Slot: L11+L12+L19+L20
ASSIGNMENT - 9
Last Date for Submission: Thursday (13-04-2023) @ 12.40PM

Name: Chilamakuru Sai Venkata Yaswanth

Reg No: 20BCI7076

Write the program using (C / C++ / Java / Python) to solve the following problems.

Concept: Dynamic Programming

S.No	Problem Name	Statement
1	Lengthiest Growing Subsequence	Given student roll numbers s1. . . sn, determine a subsequence (not necessarily contiguous) of maximum length in which the values in the subsequence form a strictly increasing sequence. If the sequence is (55,66,22,33,44,11.88,99,65), then (55,66), (33,44), (11,88,99) are all increasing sub-sequences. The longest one of them is (22,33,44,88,99).
2	DP memoization	Convert the following recurrence to code. T(0) = T(1) = 2 T(n) = $\sum \quad 2 * T(i) * T(i-1)$, for n > 1 Write the Recursive Code and DP Memoization code and expected the time complexity O(n), Space Complexity: O(n).
3	Extreme member Contiguous Subsequence	Given some random n values in an array, find the contiguous subsequence for which the sum of elements is maximum. Example: {-12, 21, -14, 23, -15, 12} → 30 and {-22, -33, 44, -11, -22, 11,15, -33} → 37 Expected Time Complexity: O(n), Space Complexity: O(1)

Note:

- If Code similarity is found, assignment will not be considered and Zero (0) Marks will be awarded.
- You have to upload a single document consisting of all the above programs and corresponding Output.

1) Lengthiest Growing Subsequences:

Code:

```
import java.util.*;
public class Assignments{
   public static void main(String [] args){
      Scanner sc = new Scanner(System.in);
               System.out.print("");
      int n=sc.nextInt();
      int[] arr=new int[n];
      System.out.print("");
      for(int i=0;i<n;i++){
         arr[i]=sc.nextInt();
      }
      int[] a=new int[n];
      Arrays.fill(a,1);
      for(int i=1;i<n;i++){
         for(int j=0;j<i;j++){
            if(arr[j]<arr[i]){
               a[i]=Math.max(a[i],a[j]+1);
            }
         }
      }
      int max=0;
      for(int i=0;i<n;i++){
         max=Math.max(max,a[i]);
      }
      List<Integer> subsequence=new ArrayList<Integer>();
      int currentLength=max;
      for(int i=n-1;i>=0;i--){
         if (a[i]==currentLength) {
            subsequence.add(arr[i]);
            currentLength--;
         }
      }
      Collections.reverse(subsequence);
      System.out.println("" + subsequence);
   }
}
```

Output:

```
C:\Users\Windows\OneDrive\Documents\CP>javac Assignments.java

C:\Users\Windows\OneDrive\Documents\CP>java Assignments.java
9
55 66 22 33 44 11 88 99 65
[22, 33, 44, 88, 99]

C:\Users\Windows\OneDrive\Documents\CP>
```

2) DP Memoization:

Code:

```
import java.util.*;
public class Assignments{
	public static void main(String [] args){
		Scanner sc=new Scanner(System.in);
		System.out.print("");
		int n=sc.nextInt();
		sc.close();
		System.out.println("T(" + n + ")=" + T(n));
		}
		public static int T(int n){
			int[] m=new int[n+1];
			Arrays.fill(m,-1);
			return TH(n, m);
		}
			public static int TH(int n,int[] m){
				if(n==0 || n==1){
					return 2;
				}
					if(m[n]!=-1){
						return m[n];
					}
						int res=0;
						for(int i=1;i<=n-1;i++){
							res+=2*TH(i-1,m)*TH(n-i,m);
						}
```

```
                                        m[n]=res;
                                        return res;
                    }
}
```

Output:

```
C:\Users\Windows\OneDrive\Documents\CP>javac Assignments.java

C:\Users\Windows\OneDrive\Documents\CP>java Assignments.java
5
T(5)=1344

C:\Users\Windows\OneDrive\Documents\CP>
```

3) Extreme member Contiguous Subsequence

Code:

```
import java.util.Scanner;
public class Assignments{
   public static void main(String [] args){
      Scanner sc=new Scanner(System.in);
      System.out.print("");
      int n=sc.nextInt();
      int[] arr=new int[n];
      System.out.print("");
      for(int i=0;i<n;i++){
         arr[i]=sc.nextInt();
      }
      int max=arr[0];
               int cur=arr[0];
      for(int i=1;i<n;i++){
         cur=Math.max(cur+arr[i],arr[i]);
         max=Math.max(max,cur);
      }
      System.out.println("" + max);
      sc.close();
   }
}
```

Output:

```
C:\Users\Windows\OneDrive\Documents\CP>javac Assignments.java

C:\Users\Windows\OneDrive\Documents\CP>java Assignments.java
6
-12 21 -14 23 -15 12
30

C:\Users\Windows\OneDrive\Documents\CP>
```