

PROGRAMACION II - GUIA DE EJERCICIOS DE PROGRAMACIÓN DINÁMICA

Ejercicio 1:

Escriba un programa que calcule el valor de Fibonacci de manera recursiva para el número entero 30 e imprima la cantidad de llamadas que se ejecutan para obtener el mismo. Luego, optimice dicho algoritmo utilizando algunas de las estrategias de Programación Dinámica Top-Down o Bottom-Up. Vuelva a imprimir el valor de Fibonacci para el mismo entero 30 y la cantidad de llamadas que se utilizó para el mismo.

Por último verifique lo siguiente:

- ¿Fueron los resultados de cálculo de Fibo(30) similares?
- ¿Obtuvo optimización en las llamadas recursivas utilizando alguna de las estrategias de Programación Dinámica?
- ¿Cual es la relación de optimización?

Ejercicio 2:

El combinatorio $\binom{n}{k}$ se puede calcular como $\frac{n!}{k!(n-k)!}$. Sabiendo que el cálculo combinatorio se obtiene a partir de la fórmula:

$$n! = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 2 \cdot 1$$

Realice un algoritmo que permita obtener el combinatorio de 2 números enteros. Intente optimizar dicho algoritmo utilizando alguna de las técnicas de Programación Dinámica para evitar realizar cálculos repetitivos.

Ejercicio 3:

Maximum Increase (Ejercicio de CodeForces)
<http://codeforces.com/problemset/problem/702/A>

Dado un arreglo compuesto de n enteros, se solicita encontrar la longitud máxima de un subarray creciente del arreglo dato.

Sabiendo que: un subarray es la secuencia de elementos consecutivos de la matriz. Subarray se llama creciente si cada elemento de este subarray es estrictamente mayor que el anterior.

Entrada

La primera línea contiene un único entero positivo n ($1 \leq n \leq 10^5$).

La segunda línea contiene n enteros positivos a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Salida

Imprimir la longitud máxima del subarreglo creciente del arreglo dado.

Ejemplos

Entrada	Salida
5 1 7 2 11 15	3
6 100 100 100 100 100 100	1
3 1 2 3	3