

Prácticas de Algorítmica.
3º de Grado en Ingeniería Informática.
Curso 2016-2017.

Práctica 2. Recursividad.

Objetivos.

Con esta que el alumno se pretende que el alumno implemente algunos algoritmos recursivos típicos, evaluando de forma híbrida su complejidad computacional y comparando las complejidades computacionales entre versiones recursivas e iterativas. En uno de los casos se verá como se puede evitar la repetición de llamadas recursivas usando una tabla que almacene los valores devueltos por las llamadas ya realizadas.

Enunciado.

Implementad en C++ un programa que contenga un menú con la siguientes opciones

Opción 1.

Cálculo del número combinatorio $C_{n,k}$ de tres formas distintas:

1. Usando recursividad, sabiendo que $C_{n,k} = C_{n-1,k-1} + C_{n-1,k}$ y que si $k = 0$ ó $k = n$ entonces $C_{n,k} = 1$.
2. Usando recursividad pero almacenando en una tabla los valores ya conocidos de C_{ij} para no repetir llamadas recursivas.
3. Haciendo uso de un algoritmo no recursivo.

En los tres casos habrá que evaluar la complejidad computacional, en función de n solamente, usando un enfoque híbrido, mostrando la curva de ajuste y dando la posibilidad de hacer una predicción para cualquier valor de n .

Nota máxima de este apartado: 5 puntos.

Opción 2.

Cálculo del número de movimientos del algoritmo de las torres de Hanoi para un valor dado del número de discos (n). Evaluar la complejidad computacional en función del número de discos usando un enfoque híbrido, mostrando la curva de ajuste y dando la posibilidad de hacer una predicción para cualquier valor de n .

Nota máxima de este apartado: 3 puntos.

Parte opcional: Dar la posibilidad de que el usuario pueda ver en pantalla una representación paso a paso en modo texto de los diferentes movimientos que resuelven el problema.

Nota máxima de este apartado: 2 puntos.

Fecha de Comienzo: 13 de Octubre de 2016.

Fecha de Entrega: 27 de Octubre de 2016.