UNIVERSIDAD DE GRANADA

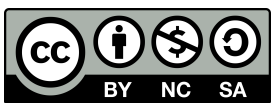
Escuela Técnica Superior de Ingeniería Informática y Telecomunicaciones

Práctica 2: Divide y Vencerás

Doble Grado Ingeniería Informática y Matemáticas

Autores:
Jose Alberto Hoces Castro
Javier Gómez López
Moya Martín Castaño

Abril 2022



Índice

1. Introducción

El objetivo de esta práctica es utilizar la técnica "divide y vencerás" para resolver problemas de forma más eficiente que otras alternativas más sencillas o directas. Para ello, se plantean los siguientes dos problemas:

- **Ejercicio 1:** Este problema consiste en realizar la búsqueda de un elemento en un vector ordenado con n elementos.
- **Ejercicio 2:** Este problema consiste en dados k vectores de n elementos, todos ellos ordenados de menor a mayor, combinar todos los vectores en uno único ordenado.

2. Desarrollo

Para los anásis de algoritmos que nos pedirán más adelante, hemos realizado los siguientes pasos:

1. Un **análisis teórico** de los algoritmos usando las técnicas vistas en clase.
2. Un **análisis empírico** donde hemos ejecutado los algoritmos en nuestros ordenadores bajo las misma normas y condiciones. Hemos compilado usando la compilación `-Og`. Además hemos usado como *datasets* de pruebas generadores de datos aleatorios proporcionados por la profesora. Por otro lado, para automatizar el proceso, hemos generado unos *scripts* de generación de datos de prueba y de ejecución de nuestros programas. Hemos ejecutado cada algoritmo 15 veces en cada uno de los tamaños que han sido probados, y hemos hecho la media de ellos para reducir perturbaciones que puedan alterar el resultado.
3. Un **análisis híbrido** donde hemos tomado los datos de cada uno de los alumnos del grupo y hemos hallado la K (constante oculta). Para ello hemos usado gnuplot.

2.1. Ejercicio 1

El enunciado del problema es el siguiente: *Dado un vector ordenado (de forma no decreciente) de números enteros v, todos distintos, el objetivo es determinar si existe un índice i tal que $v[i] = i$ y encontrarlo en ese caso. Diseñar e implementar un algoritmo "divide y vencerás" que permita resolver el problema. ¿Cuál es la complejidad de ese algoritmo y la del algoritmo "obvio" para realizar esta tarea? Realizar también un estudio empírico e híbrido de la eficiencia de ambos algoritmos.*

Supóngase ahora que los enteros no tienen por qué ser todos distintos (pueden repetirse). Determinar si el algoritmo anterior sigue siendo válido, y en caso negativo proponer uno que sí lo sea. ¿Sigue siendo preferible al algoritmo obvio?

2.1.1. Algoritmo "obvio" o de fuerza bruta

La manera obvia de resolver este ejercicio sería mediante un algoritmo secuencial, que vaya recorriendo el vector hasta encontrar el elemento buscado. Pasemos a realizar un análisis de este algoritmo:

1. **Análisis Teórico**

```
int buscarSecuencial(int v[], int n){
    for (size_t i = 0; i < n; i++) //O(n)
    {
        if (v[i] == i){ //O(1)
            return i; //O(1)
        }
        else if(v[i] > n){ //O(1)
            return -1; //O(1)
        }
    }

    return -1;//O(1)
}
```

Tal y como se ha indicado en los comentarios del código, todas las operaciones de asignación y comprobación de los `if` son $\mathcal{O}(1)$. Estas, a sus vez, se incluyen dentro de un bucle `for`. Dicho bucle es $\mathcal{O}(n)$, obteniendo así que la función `int buscarSecuencial` es $\mathcal{O}(n)$, es decir

$$T(n) \in \mathcal{O}(n)$$