UNIVERSIDAD DE GRANADA

Divide y Vencerás

Algorítmica. Práctica 2

Jose Alberto Hoces Castro
Javier Gómez López
Moya Martín Castaño

Contenidos

Introducción

Problemas planteados

- **Ejercicio 1**: Buscar en un vector ordenado un elemento tal que $v[i] = i$.
- **Ejercicio 2**: Dados k vectores ordenados, de n elementos cada uno, combinarlos en un vector ordenado.

Objetivo de la prática

Apreciar la utilidad de la técnica divide y vencerás (DyV) para resolver problemas de forma más eficiente que otras alternativas más sencillas o directas.

Ejercicio 1

Búsqueda secuencial

Es la manera más obvia de buscar en un vector. Empezamos en el primer elemento y lo vamos recorriendo hasta encontrar el elemento deseado. En caso de no encontrarlo, devolvemos un valor que indique error (en nuestro caso -1).

Búsqueda secuencial. Código

```
1  int buscarSecuencial(int v[], int n){
2      for (size_t i = 0; i < n; i++) //O(n)
3      {
4          if (v[i] == i){ //O(1)
5              return i; //O(1)
6          }
7      }
8
9      return -1;//O(1)
10 }
```

Búsqueda secuencial. Eficiencia teórica

Observamos claramente que

$$T(n) \in O(n)$$

Búsqueda secuencial. Eficiencia híbrida

INSERTAR GRAFICA

Búsqueda binaria

La tećnica Divide y Vencerás usada es la búsqueda binaria. Al estar ante un vector ordenado, podemos recurrir hasta algoritmo cuya eficiencia es logarítmica, mucho más preferible que una lineal.

Búsqueda binaria. Código

```
int buscarBinaria(vector<int> v, int n){

    int inicio = 0; //O(1)
    int fin = n-1; //O(1)
    int medio = (inicio+fin)/2; //O(1)

    while(inicio <= fin){ //O(log(n))

        if(v.at(medio) > medio){ //O(1)
            fin = medio - 1; //O(1)
        }
        else if(v.at(medio) < medio){ //O(1)
            inicio = medio + 1; //O(1)
        }
        else{
            return medio; //O(1)
        }

        medio = (inicio + fin)/2; //O(1)
    }
```

Búsqueda binaria. Eficiencia teórica

Observamos claramente que

$$T(n) \in O(\log(n))$$

Búsqueda binaria. Eficiencia híbrida

INSERTAR GRÁFICA

¿Elementos repetidos?

¿Qué pasaría si tuviésemos elementos repetidos? Por ejemplo:

1 2 3 4 4 5 6 7

¿Elementos repetidos?. Solución

```
1  int buscarBinaria(vector<int> v, int inicio,int fin){
2
3      if(inicio == fin){
4          if(v[inicio] == inicio){
5              return inicio;
6          }
7          else{
8              return -1;
9          }
10     }
11
12     if(v[inicio] != inicio && v[fin-1] != fin-1){
13
14         int medio = (inicio+fin)/2;
15         int resultado = buscarBinaria(v,inicio+1,medio);
16         if(resultado != -1){
17             return resultado;
18         }
19         else{
20                 resultado = buscarBinaria(v,medio+1,fin-1);
```

¿Elementos repetidos?. Eficacia

Tras un estudio, observamos que nuestra solución es $O(n)$ igual que nuestra búsqueda secuencial. Por tanto, no es rentable usar este algoritmo.