

# Algorítmica

Divide y vencerás

Álvaro Maximino Linares Herrera

### Mediana de un vector

La segunda práctica de la asignatura consiste en calcular la mediana de un vector mediante la operación pivote vista en las clases teóricas.

El código es el siguiente:

```
#include <iostream>
using namespace std;

void Intercambiar(int *&v, int i, int l){
    int aux;
    aux=v[i];
    v[i]=v[l];
    v[l]=aux;
} //Fin Intercambiar

void Pivote(int *v, int i, int j, int &l){

    int p;
    int k;

    p=v[i];
    k=i;
    l=j+1;
    while((v[k]<=p) && (k<j)){
        k=k+1;
    }
    while(v[l]>p){
        l=l-1;
    }

    while(k<l){
        Intercambiar(v,k,l);
        while(v[k]<=p){
            k=k+1;
        }
        while(v[l]>p){
            l=l-1;
        }
    }
}
```

```

    }
    Intercambiar(v,i,l);

} //Fin pivote

int Seleccion(int *v,int tam,int s){
    int i,j,l;
    i=0;
    j=tam;
    do{
        Pivote(v,i,j,l);
        if(s<l){
            j=l-1;
        }else if(s>l) {
            i=l+1;
        }
    }while(l!=s);

    return v[l];

} //Fin seleccion

int main(){

    int tam;
    cout << "introduce tamaño: ";
    cin >> tam;
    int v[tam];
    cout << "Introduce elementos del vector: \n";
    for(int i=0;i<tam;i++){
        cin >> v[i];
    }
    int mediana=tam/2;
    cout << "La mediana del vector es: " << Seleccion(v,tam,mediana) << endl;

} //Fin main

```

En este código uso varias funciones de las que explico su funcionamiento a continuación.

La función Intercambiar, como su propio nombre indica lo que hace es intercambiar dos valores

enteros.

La función Pivote lo que hace es recorrer el vector que le pasamos como parámetro, recorrerlo de izquierda a derecha y de derecha a izquierda e ir intercambiando los valores que no están ordenados.

La función Selección lo que nos devuelve es el elemento de la posición que le pasamos, como lo que le pasamos en el cálculo de la mediana es la mitad del tamaño nos devuelve el elemento medio, con lo que nos resolvería el problema.

### Orden de eficiencia

Nos encontramos ante el problema de Selección, es un algoritmo de reducción, por lo que vemos el problema es descompuesto en un solo subproblema de tamaño menor y vemos que el procedimiento es no recursivo.

La función más interesante es la función Pivote pues nos permite ordenar el vector que le pasemos con un coste de  $O(n)$ , que es lo mejor que podemos obtener ordenando vectores.

Para resolver el problema de la mediana con la función Selección necesitaríamos, ordenar el vector y devolver la posición que le pasamos. Esto requeriría un coste de  $O(n \log n)$ . Pero usando la función Pivote lo resolveríamos con un coste de  $O(n)$ . Por lo tanto como la usamos el orden de eficiencia es de  $O(n)$ .

En el mejor caso el subproblema que obtenemos es de tamaño  $n/2$ :

$$t(n)=t(n/2)+a*n; \quad t(n) \in O(n)$$

En el peor de los casos, es decir, que el vector este ordenado, el subproblema es de tamaño  $n-1$  por lo que el orden sería:

$$t(n)=t(n-1)+a*n; \quad t(n) \in O(n^2)$$

En el caso promedio el tiempo del algoritmo es un  $O(n)$ .

### Ejecuciones

Vamos a ver unas cuantas ejecuciones.

Para un vector de tamaño 5 y de elementos 5,3,1,4,2 la mediana debería de ser 3., veamos si el programa es capaz de hacerlo bien...

```
>g++ -o mediana pivote.cpp
>./mediana
introduce tamaño: 5
Introduce elementos del vector:
5
3
1
4
2
35 int tam;
16 cout << "introduce tamaño: ";
47 cin >> tam;
28 int v[tam];
La mediana del vector es: 3
```

Como podemos observar el programa lo hace correctamente.

Veamos ahora un ejemplo algo más grande y que sea el peor caso, es decir, que el vector este ordenado.

Le pongo de tamaño 12 y como elementos del vector: 1,2,3...10,11,12.

```
> ./mediana
introduce tamaño: 12
Introduce elementos del vector:
1
2
3
4
5
6
7
8
9
10
11
12
La mediana del vector es: 6
```

Como se puede observar el programa se comporta como es esperado y nos devuelve el elemento medio, es decir, 6.

Vamos a pasarle ahora un archivo de texto que contenga el tamaño y los datos del vector.

El tamaño del vector es 38 y esta formado por los siguientes elementos escogidos al hazar:

1 , 4, 634, 5 , 45 , 45 ,634, 54, 2 , 787, 6 , 46 , 43 , 43 , 78 , 655, 63, 4 , 3652, 345 , 632, 23 , 148, 65 , 341 , 384, 33 , 213, 43, 2 , 486, 43 , 123, 8 , 64, 3 , 13 , 8.

Veamos el ejemplo de ejecución:

```
> ./mediana < pruebamediana
introduce tamaño: Introduce elementos del vector:
La mediana del vector es: 45
```

Nos devuelve de mediana 45.