09/06/2020

Práctica FINAL. Análisis de algoritmos de búsqueda y ordenación

Ismael Da Palma Fernández

Grupo de Teoría: T1 **Grupo de Prácticas:** L4

----- ÍNDICE -----

1. Algoritmo del examen. Cálculo del tiempo experimental

1.1 Pseudocódigo (Junio2020)

El algoritmo que ha sido propuesto en el Grupo de prácticas L4 se trata de un algoritmo de ordenación llamado “Junio2020” que consiste en tomar elemento a elemento del array e ir insertando cada elemento en su posición correcta de manera que se mantiene el orden de los elementos ya ordenados. El algoritmo dispone de 3 métodos para su implementación en la práctica.

```
void AlgoritmosOrdenacion::ordenaJunio2020(int v[], int size)
{
    Junio2020(v, 1, size);
}
```

El método **ordenaJunio2020** se encarga de llamar al método **Junio2020** inicializado la primera posición del vector en la 1, ya que suponemos que la posición 0 está ordenada, por lo que realizará una ordenación desde el segundo hasta el último.

```
void AlgoritmosOrdenacion::Junio2020(int a[], int primero, int ultimo)
{
    int x, k;
    for (int i = primero+1; i < ultimo; i++)
    {
        x = a[i];
        /*Busqueda binaria de la posicion insercion*/
        k = buscarPosicion(a, primero, i-1, x);
        /*desplazamos a la derecha los elementos ordenados para insertar el nuevo*/
        for (int j = i-1; j >= k; j--)
            a[j + 1] = a[j];

        /*Insertar el nuevo*/
        a[k] = x;
    }
}
```

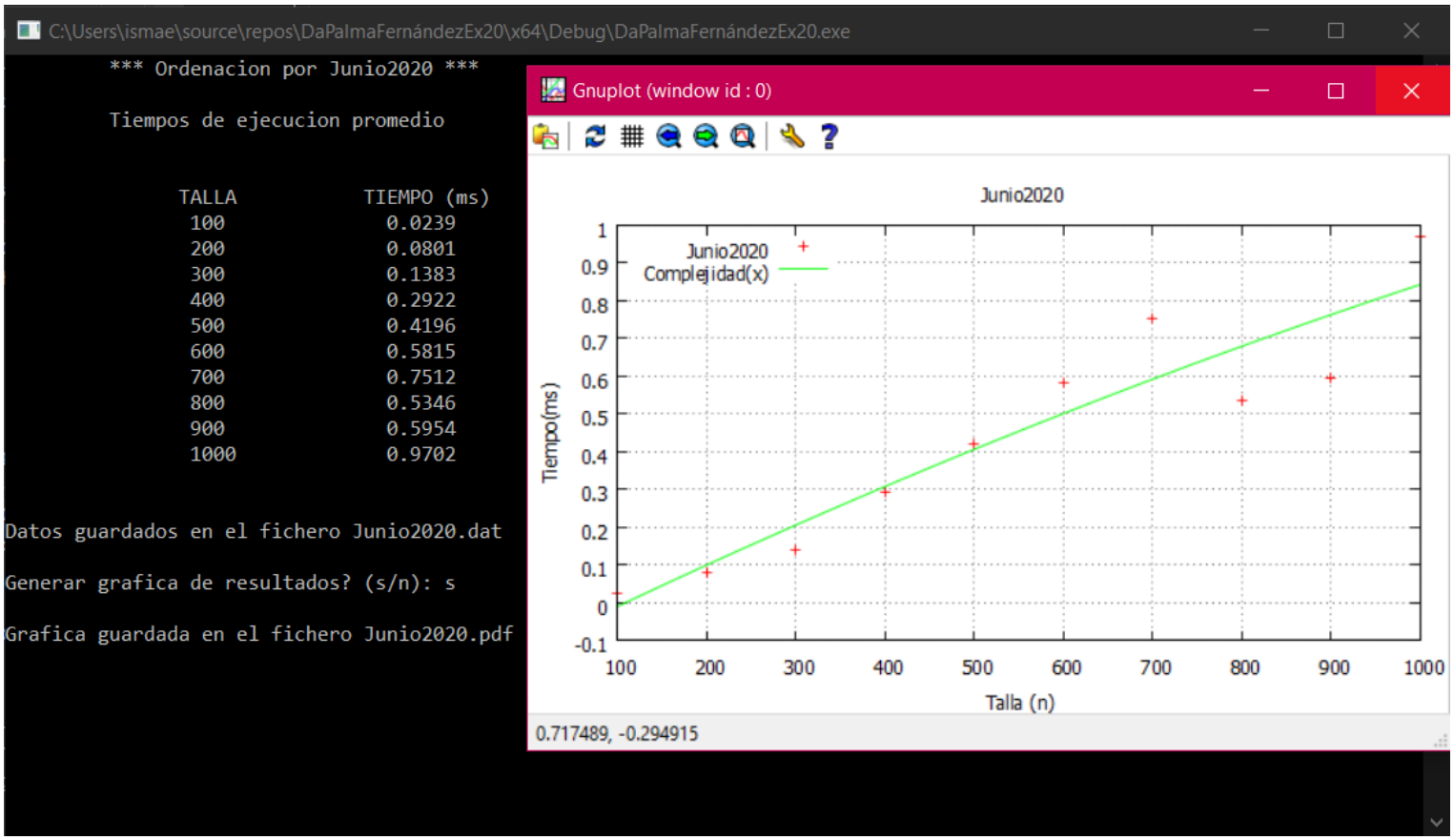
El método **Junio2020** se encarga de ordenar a partir del segundo elemento, como ya hemos dicho, suponiendo ordenados los (i-1) primeros elementos.

Realiza una búsqueda binaria que es la que se encuentra en el método **buscarPosición**, desplaza los elementos a la derecha y los inserta, obteniendo i elementos ordenados.

```
int AlgoritmosOrdenacion::buscarPosicion(int v[], int primero, int ultimo, int key)
{
    while (primero <= ultimo)
    {
        int mitad = (primero + ultimo) / 2;
        if (key < v[mitad])
            ultimo = mitad - 1;
        else
            primero = mitad + 1;
    }
    return primero;
}
```

buscarPosición se encarga de calcular la posición donde hay que insertar un nuevo elemento dentro de un subvector previamente ordenado, se basa en el algoritmo de búsqueda binaria iterativa.

1.2 Tablas y gráficas de coste



1.3 Conclusiones

Podemos observar que el algoritmo tiene un pequeño grado de dispersión debido a que los valores que genera el programa al azar y hay algunos que tarden más tiempo que otros.

También se puede observar que efectivamente, como se decía en el enunciado de la modificación, el algoritmo es de orden $O(n^2)$, por lo que cuanto más grande sea la talla, más tiempo tardará en realizar el algoritmo.

2. Diseño de la aplicación

En este apartado no hay mucho que decir, ya que la estructura del programa es la misma que la Práctica 3, lo único que se ha modificado en esta práctica es el fuente AlgoritmosOrdenación, donde se ha incluido el nuevo método de ordenación Junio2020. También se ha modificado el fuente Constantes.h para poder añadir a Junio2020 en los métodos de TestOrdenacion.

3. Conclusiones y valoraciones personales de la Práctica

Durante el cuatrimestre hemos realizado una serie de modificaciones en las prácticas que íbamos trabajando en su día, estas modificaciones nos enseñaban a comprender aun más la práctica ya que aprendemos a añadir, quitar o cambiar métodos sin que la práctica sufra ningún error de compilación o traspies.