

MEMORIA PRÁCTICA 1

FUNDAMENTOS DE ANÁLISIS DE ALGORITMOS



2021-2022

DANIEL LINFON YE LIU
INGENIERÍA INFORMÁTICA

ÍNDICE

1. Introducción. Algoritmo Búsqueda secuencial. (Pág. 3)
2. Cálculo del tiempo teórico (Pág. 3,4,5):
 - Pseudocódigo (código) y análisis de coste. (Pág. 3)
 - Tablas y Gráficas de coste. (Pág. 4)
 - Conclusiones. (Pág. 5)
3. Cálculo del tiempo experimental: (Pág. 5,6)
 - Tablas y Gráficas de coste. (Pág 5)
 - Conclusiones. (Pág. 6)
4. Comparación de los resultados teórico y experimental.
(Pág. 6)
5. Diseño de la aplicación. (Pág. 7)
6. Conclusiones y valoraciones personales de la práctica.
(Pág. 7)

1. Introducción

El objetivo de esta práctica es realizar el estudio teórico de un algoritmo de búsqueda secuencial, posteriormente implementarlo en un lenguaje de programación y medirlo experimentalmente. El análisis del algoritmo conlleva:

- **Estudio teórico:** Obtener la expresión de la función complejidad temporal ($T(n)$) correspondiente al algoritmo.
- **Estudio Empírico:** Se mide el tiempo para cada tamaño dado de las entradas, obteniendo los puntos de la función complejidad para los casos peor, mejor y medio.
- **Conclusión:** Se realiza una gráfica para cada una de las tres funciones obtenidas en el estudio empírico, si el análisis es por casos, y se comentan justificando la respuesta.

2. Cálculo de tiempo teórico

Se aplicarán las reglas conocidas para contar el número de operaciones que realiza un algoritmo. Este valor será expresado como una función de $T(n)$ que dará el número de operaciones requeridas para un caso concreto del problema caracterizado por tener un tamaño n . El análisis lo realizaremos para los casos mejor, peor y medio.

2.2. Pseudocódigo (código) y análisis de coste

| líneas | <code>int BusquedaSecuencial(int T[], int n, int valor)</code> | <u>Nº</u> <u>OE</u> | |
|--------|--|------------------------|---|
| | { | | |
| 1) | <code>int i = 0;</code> | 1 | asignación |
| 2) | <code>while (T[i] != valor && i<n) {</code> | 4 | Condición del Bucle (2comp., 1ac. vector, 1 lóg.) |
| 3) | <code>i=i+1;</code> | 2 | Incremento y asignación |
| 4) | <code>}</code> | | |
| 5) | <code>if (T[i]==valor)</code> | 2 | 1 condición y 1 acc. al vector |
| 6) | <code>return i;</code> | 1 | si la condición se cumple |
| 7) | <code>else return -1;</code> | 1 | cuando condición es falsa. |
| | } | | |

- El tiempo de ejecución del algoritmo será:

$$T_{\text{BSecuencial}}(n) = T_{\text{Asig}}(1) + T_{\text{Bucle}}(2) + T_{\text{Si}}(5)$$

- Para calcularlo lo haremos por partes:

$$T_{\text{Asig}(1)} = 1$$

$$T_{\text{Si}(5)} = T_{\text{condSi}} + T_{\text{cuerpoSi}} = 2 + \text{máx/mín/medio}(T_{\text{return}(667)}) = 2 + 1 = 3$$

$$T_{\text{Bucle}(2)} = T_{\text{condB}} + \sum_{i=1;?} T_{\text{cicloBucle}} = 4 + \sum_{i=1;?} T_{\text{cicloBucle}}$$

$$T_{\text{cicloBucle}} = T_{\text{condB}} + T_{\text{cuerpoB}} (= 0 \text{ sólo instrucción de incremento del bucle}) + T_{\text{incrementoB}} = 4 + 2 = 6$$

$$T_{\text{Bucle}(2)} = 4 + \sum_{i=1;?} 6 = 4 + 6 \sum_{i=1;?}$$

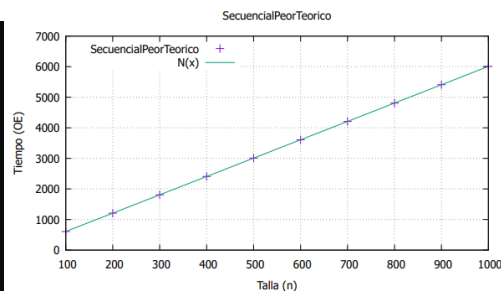
$$T_{\text{BSecuencial}}(n) = T_{\text{Asig}(1)} + T_{\text{Bucle}(2)} + T_{\text{Si}(5)} = 1 + 4 + 6 \sum_{i=1;?} + 3 = 8 + 6 \sum_{i=1;?}$$

$$T_{\text{BSecuencial}}(n) = 8 + 6 \sum_{i=1;?}$$

2.2. Tablas y gráficas de coste

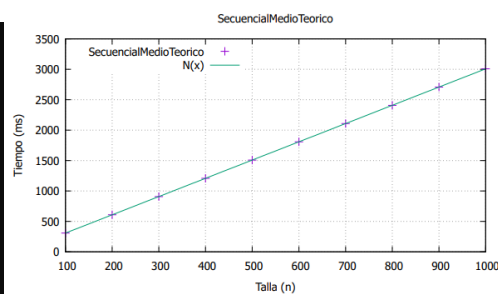
- Caso Peor:** $T_{\text{BSecuencial}}(n) = 8 + 6 \sum_{i=1;n} = 6n + 8$

| Talla | Tiempo (oe) |
|-------|-------------|
| 100 | 6.1e+02 |
| 200 | 1.2e+03 |
| 300 | 1.8e+03 |
| 400 | 2.4e+03 |
| 500 | 3e+03 |
| 600 | 3.6e+03 |
| 700 | 4.2e+03 |
| 800 | 4.8e+03 |
| 900 | 5.4e+03 |
| 1000 | 6e+03 |



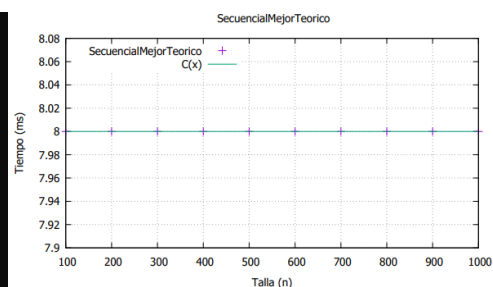
- Caso Medio:** $T_{\text{BSecuencial}}(n) = 8 + 6 \sum_{i=1;n/2} = 8 + 6(n/2) = 3n + 8$

| Talla | Tiempo (oe) |
|-------|-------------|
| 100 | 3.1e+02 |
| 200 | 6.1e+02 |
| 300 | 9.1e+02 |
| 400 | 1.2e+03 |
| 500 | 1.5e+03 |
| 600 | 1.8e+03 |
| 700 | 2.1e+03 |
| 800 | 2.4e+03 |
| 900 | 2.7e+03 |
| 1000 | 3e+03 |



- Caso Mejor:** $T_{\text{BSecuencial}}(n) = 8 + 6 \sum_{i=1;0} = 8$

| Talla | Tiempo (oe) |
|-------|-------------|
| 100 | 8 |
| 200 | 8 |
| 300 | 8 |
| 400 | 8 |
| 500 | 8 |
| 600 | 8 |
| 700 | 8 |
| 800 | 8 |
| 900 | 8 |
| 1000 | 8 |



2.3. Conclusiones

Como se esperaba, el coste temporal del algoritmo depende de la talla del problema y de la instancia en concreto. Búsquedas con la misma talla tienen costes diferentes, dependiendo de dónde se encuentre la clave.

Aún en el peor caso, este algoritmo sigue un modelo de crecimiento lineal. Este crecimiento puede verse representado como rectas en la gráfica. La función complejidad es lineal.

3. Cálculo de tiempo experimental

Se trata de estudiar experimentalmente el comportamiento del algoritmo. Para ello mediremos los recursos empleados (tiempo) para cada tamaño dado de las entradas.

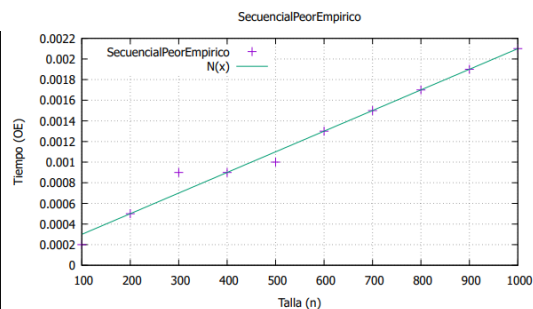
3.3. Tablas y gráficas de coste

- **Caso Peor:**

```
Busqueda SecuencialSecuencialPeor EmpiricoTiempos de ejecucion
```

| Talla | Tiempo (ms) |
|-------|-------------|
| 100 | 0.0002 |
| 200 | 0.0005 |
| 300 | 0.0009 |
| 400 | 0.0009 |
| 500 | 0.001 |
| 600 | 0.0013 |
| 700 | 0.0015 |
| 800 | 0.0017 |
| 900 | 0.0019 |
| 1000 | 0.0021 |

Datos guardados en el fichero SecuencialPeorEmpirico.dat

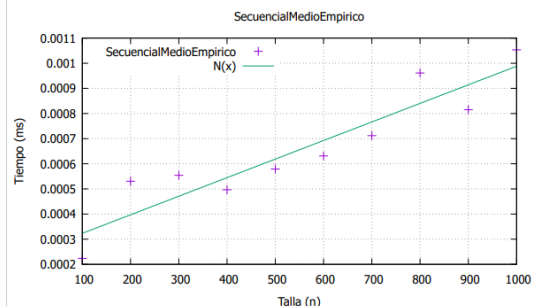


- **Caso Medio:**

```
Busqueda SecuencialSecuencialMedio EmpiricoTiempos de ejecucion
```

| Talla | Tiempo (ms) |
|-------|-------------|
| 100 | 0.00022 |
| 200 | 0.00053 |
| 300 | 0.00055 |
| 400 | 0.0005 |
| 500 | 0.00058 |
| 600 | 0.00063 |
| 700 | 0.00071 |
| 800 | 0.00096 |
| 900 | 0.00081 |
| 1000 | 0.0011 |

Datos guardados en el fichero SecuencialMedioEmpirico.dat

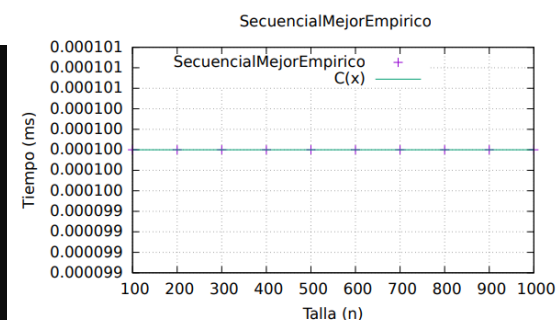


- **Caso Mejor:**

```
Busqueda SecuencialSecuencialMejor EmpiricoTiempos de ejecucion
```

| Talla | Tiempo (ms) |
|-------|-------------|
| 100 | 0.0002 |
| 200 | 0.0001 |
| 300 | 0.0001 |
| 400 | 0.0001 |
| 500 | 0.0001 |
| 600 | 0.0001 |
| 700 | 0.0001 |
| 800 | 0.0001 |
| 900 | 0.0001 |
| 1000 | 0.0001 |

Datos guardados en el fichero SecuencialMejorEmpirico.dat



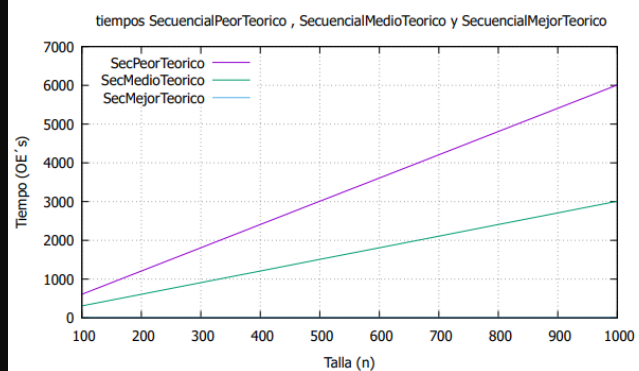
3.2. Conclusiones

Los resultados individuales muestran un grado tolerable de dispersión, sin elementos anómalos. Siendo esta dispersión más apreciable en el caso medio, donde hacen falta repetir la medición varios ordenes de magnitud para datos más próximos a la regresión lineal. En el caso mejor por separado, aparenta una gran dispersión; pero, teniendo en cuenta la escapa, la diferencia es mínima y se debe al límite de precisión con la que se mide el tiempo transcurrido.

4. Comparación de los resultados teórico y experimental

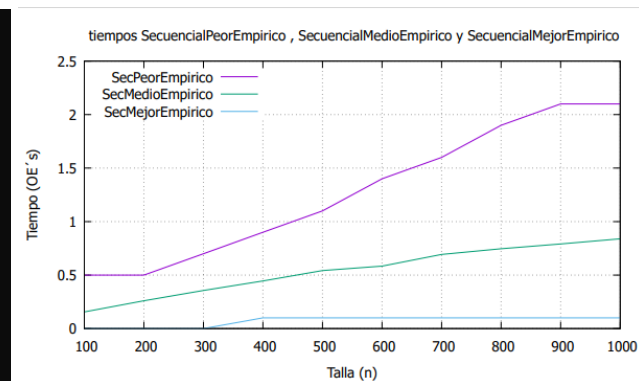
Comparación teórico

| Busqueda Secuencial Teorico. Tiempos de ejecucion | | | |
|---|-------------|---------|---|
| Talla | Tiempo (oe) | | |
| 100 | 6.1e+02 | 3.1e+02 | 8 |
| 200 | 1.2e+03 | 6.1e+02 | 8 |
| 300 | 1.8e+03 | 9.1e+02 | 8 |
| 400 | 2.4e+03 | 1.2e+03 | 8 |
| 500 | 3e+03 | 1.5e+03 | 8 |
| 600 | 3.6e+03 | 1.8e+03 | 8 |
| 700 | 4.2e+03 | 2.1e+03 | 8 |
| 800 | 4.8e+03 | 2.4e+03 | 8 |
| 900 | 5.4e+03 | 2.7e+03 | 8 |
| 1000 | 6e+03 | 3e+03 | 8 |



Comparación experimental

| Busqueda Secuencial Empirico. Tiempos de ejecucion | | | |
|--|-------------|------|-----|
| Talla | Tiempo (ms) | | |
| 100 | 0.5 | 0.16 | 0 |
| 200 | 0.5 | 0.26 | 0 |
| 300 | 0.7 | 0.36 | 0 |
| 400 | 0.9 | 0.44 | 0.1 |
| 500 | 1.1 | 0.54 | 0.1 |
| 600 | 1.4 | 0.58 | 0.1 |
| 700 | 1.6 | 0.69 | 0.1 |
| 800 | 1.9 | 0.75 | 0.1 |
| 900 | 2.1 | 0.79 | 0.1 |
| 1000 | 2.1 | 0.84 | 0.1 |



Como podemos ver en las imágenes, los valores de los casos empíricos se corresponden con sus gráficas, así como los valores de los casos teóricos se corresponden con las suyas. Si nos fijamos, son bastante distintos los resultados teóricos de los experimentales, ya que incluso con la misma talla, los resultados teóricos en comparación con los empíricos se disparan. Esto se debe a que los casos teóricos no están medidos en milisegundos, sino en operaciones elementales, por lo que para los casos teóricos, veríamos cuantas operaciones debe llevar a cabo el programa según la talla, mientras que en los casos empíricos veríamos cuanto tarda en ejecutarse para cada talla.

5. Diseño de la aplicación

Como podemos ver en la imagen, el proyecto se compone de los archivos de encabezado `ConjuntoInt.h`, `Constantes.h`, `Mtime.h` y `TestAlgoritmo.h`, de los archivos de origen `ConjuntoInt.cpp`, `Principal.cpp`, `Mtime.cpp` y `TestAlgoritmo.cpp` y de una serie de archivos de recursos que en este caso son las gráficas que se generan cuando ejecutamos el programa, tanto de los casos teóricos, como de los empíricos. Los archivos que llevan la extensión `.h` son aquellos en los que se encuentran la clase que implementaremos en aquellos con extensión `.cpp`, a excepción de `Constantes.h` que define las constantes que usaremos en el proyecto.



`TestAlgoritmos` tiene los siguientes métodos principales:

- `comprobarAlgoritmo`.
- `costeTeorico`: Recibe por parámetro el caso a estudiar de forma teórica. Muestra por pantalla y guarda en un archivo los valores que toma la función complejidad para distintas tallas y casos.
- `compararTeorico`: Recibe por parámetro los tres casos a comparar teóricamente. Muestra por pantalla y guarda en un archivo los valores de la función complejidad para todos los casos y diferentes tallas.
- `costeEmpirico`: Recibe por parámetro el caso a simular empíricamente. Genera un conjunto de datos aleatorios, y mide el tiempo que transcurre entre el inicio y el fin de la llamada al algoritmo de búsqueda. Repite el proceso para distintas tallas, muestra los resultados por pantalla y los guarda en un fichero.
 - En el caso peor, se le introduce la clave de búsqueda -1, valor que no existe en el conjunto de datos.
 - En el caso medio, se escoge una clave aleatoria de entre los elementos contenidos en el conjunto. Las mediciones se repiten múltiples veces y se toma la media.
 - En el caso mejor, se pasa el primer elemento del conjunto como clave.
- `compararEmpirico`: Recibe por parámetro los tres casos a simular empíricamente. Realiza las mismas mediciones de los tres casos, los muestra por pantalla y los guarda en un archivo.

`Principal.cpp` consta de un menú principal y dos submenús, para el estudio teórico y empírico. En ambos submenús se puede probar el algoritmo de búsqueda, generar resultados para un caso en particular, o comparar los resultados de los tres casos.

6. Conclusiones y valoraciones personales de la práctica.

Las gráficas de los resultados teóricos son coherentes con los resultados obtenidos, puesto que las ecuaciones en las que se basa el programa para formar la gráfica son como la ecuación de una recta, que efectivamente es lo que sale.

En cuanto a los resultados empíricos, en el caso mejor los valores son muy próximos a 0 y por lo tanto la gráfica es prácticamente una recta. En el caso peor los valores están van de menos a más dando lugar a recta ascendente. Sin embargo, en el caso medio los valores que salen son muy lejanos entre ellos. Esta práctica ha servido para visualizar la utilidad del análisis de algoritmos.