



PRACTICA 1 FAA

Rafael Mesa Nombela

Ingeniería Informática

2022

Índice

Introducción y Algoritmo Búsqueda Secuencial	3
Cálculo del Tiempo Teórico	4
Seudocódigo	4
Caso Mejor	5
Caso Peor	5
Caso Medio	5
Grafica y Tabla de corte	6
Conclusión	7
Calculo de tiempo experimental	8
Gráficas y Tablas de coste	
Conclusión	
Comparación de resultados teóricos con resultados experimentales	11
Diseño de Aplicación	11
Conclusiones finales	12

Introducción al Algoritmo de Búsqueda Secuencial

Lo primero que debemos preguntarnos ¿Es cuál es la finalidad de un algoritmo? En este caso la finalidad del algoritmo de Búsqueda Secuencial es buscar de un elemento n dentro de un arreglo de N elementos. Lo segundo que nos preguntamos es ¿Cómo lo hace? El algoritmo va recorriendo el arreglo y comparando cada elemento con n hasta encontrarlo. Lo tercero que nos debemos preguntar es ¿Cuan eficiente es dicho algoritmo? Como todo algoritmo, se miden tres casos diferentes: Caso Mejor, Caso Peor, Caso Medio. Siendo el primero cuando el elemento se encuentra el primero del arreglo y se ejecuta 1 vez. El segundo cuando n no se encuentra en el arreglo. Y el Caso Medio cuando dicho elemento se encuentra en una posición aleatoria dentro del arreglo.

Ahora nos dispondremos a estudiar en profundidad los tres casos.

Código del algoritmo en c++

```
int BusquedaSecuencial(int V[],int numelementos,int valor){ ///Numero Operaciones Elementales:

    int i=0; //1: Asignación

    while(V[i]!=valor&& i<numelementos){ /*4: 2 Comparaciones, 1 Acceso al vector, 1 logica*/
        i++; //2: Incremento y asignación
    }

    if(V[i]==valor) //2: Condición y Acceso al vector
        return i; /*1: En caso de que la condicion sea correcta*/
    else
        return -1; //1: Si la condicion es falsa
}
```

Cálculo del tiempo teórico

Pseudocódigo:

$$T_{BSequencial}(n) = T_{Asig}(1) + T_{Bucle}(2) + T_{Si}(5)$$

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

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

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

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

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

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

$$T_{BSequencial}(n) = 8 + 6 \sum(i=1;?)$$

Caso Mejor

Para el caso mejor las únicas líneas que se ejecutaran son la 1 y la 2, y posteriormente la 5 y la 7, por lo que no se efectuaría el bucle.

$$TB_{\text{Secuencial}}(n) = 9 + 7 \sum_{i=1}^0 = 9$$

Caso Medio

Para el caso medio el bucle se efectuará una cantidad de veces comprendida entre 1 y n veces teniendo todas igual probabilidad de suceder. Por lo que al final el bucle se ejecuta $n/2$ veces.

$$TB_{\text{Secuencial}}(n) = 9 + 7 \sum_{i=1}^{n/2} = (7/2)n + 9$$

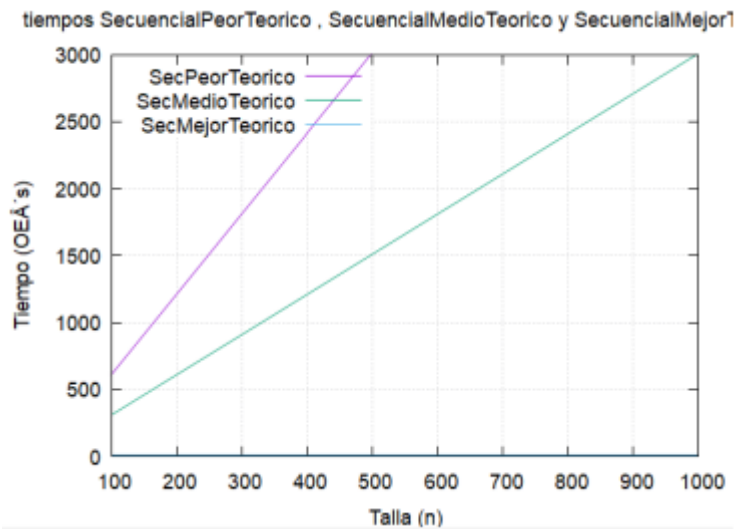
Caso Peor

Este caso se da cuando el elemento que estamos buscando no se encuentra en el arreglo por lo que el algoritmo actuará de la siguiente forma:

Se ejecuta la primera línea y a continuación entra en el bucle y se ejecuta n veces hasta que deja de cumplirse la segunda condición. Tras esto se ejecuta la línea 5 y finaliza ejecutando la línea 7.

$$TB_{\text{Secuencial}}(n) = 9 + 7 \sum_{i=1}^n = 7n + 9$$

Gráfica y Tabla de costes



```
C:\Users\jesus\Downloads\fuentes\x64\Debug\practica 1.exe

Busqueda Secuencial Teorico. Tiempos de ejecucion

    Talla      Tiempo (oe)
    100        6.1e+02    3.1e+02
    200        1.2e+03    6.1e+02
    300        1.8e+03    9.1e+02
    400        2.4e+03    1.2e+03
    500         3e+03     1.5e+03
    600        3.6e+03    1.8e+03
    700        4.2e+03    2.1e+03
    800        4.8e+03    2.4e+03
    900        5.4e+03    2.7e+03
   1000        6e+03     3e+03

Datos guardados en el fichero SecuencialPeorSecuenci
Generar grafica de resultados? (s/n):
```

Conclusion

En la gráfica podemos observar que la diferencia entre los tres casos reside en la pendiente, siendo esta misma la que nos da información del coste de cada caso.

En el caso Mejor tenemos una pendiente = 0, y tiene sentido ya que en este caso el elemento a encontrar se encuentra en la primera posición del arreglo y por lo tanto el tamaño del arreglo no nos importa y se ejecuta siempre las mismas operaciones elementales.

En el caso Peor tenemos la pendiente máxima, vuelve a adecuarse con la realidad ya que es el caso con el coste más elevado.

Por último, tenemos el caso Medio, nuevamente no podría ser de otra forma, con una pendiente menor a la del caso Peor y mayor que el caso Mejor, y por lo tanto con un coste Medio, esto sucede porque a diferencia del caso Mejor, si importa el número de elementos que tenga el arreglo sin embargo no se ejecutara para todas las posiciones como en el caso Peor.

Cálculo de tiempo experimental

Tablas y Graficas de coste

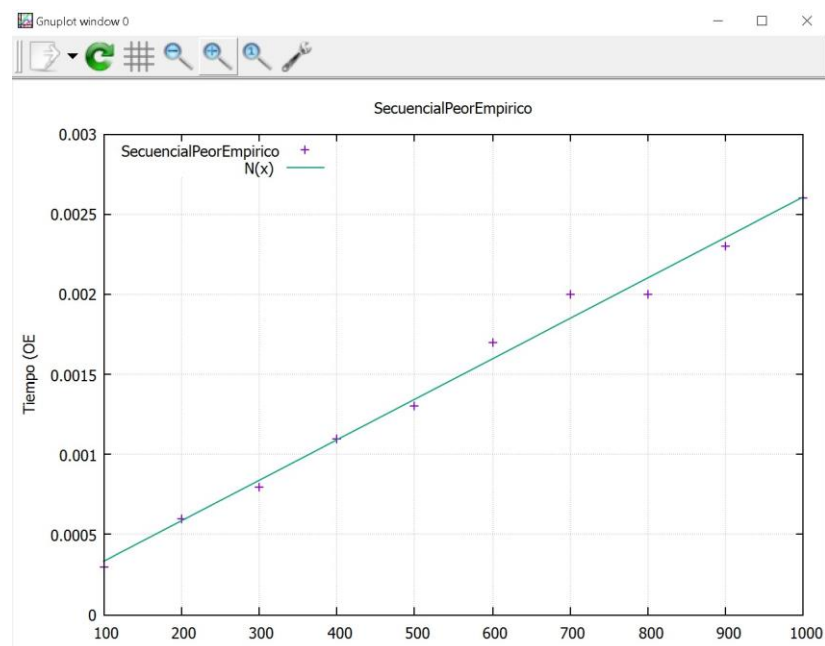
Caso Peor:

```
Busqueda SecuencialSecuencialPeor EmpiricoTiempos de ejecucion

    Talla      Tiempo (ms)

tiempo = 0.03milisegundos
    100          3e-07
tiempo = 0.06milisegundos
    200          6e-07
tiempo = 0.08milisegundos
    300          8e-07
tiempo = 0.11milisegundos
    400         1.1e-06
tiempo = 0.13milisegundos
    500         1.3e-06
tiempo = 0.17milisegundos
    600         1.7e-06
tiempo = 0.2milisegundos
    700         2e-06
tiempo = 0.2milisegundos
    800         2e-06
tiempo = 0.23milisegundos
    900         2.3e-06
tiempo = 0.26milisegundos
   1000         2.6e-06

Datos guardados en el fichero SecuencialPeorEmpirico.dat
```



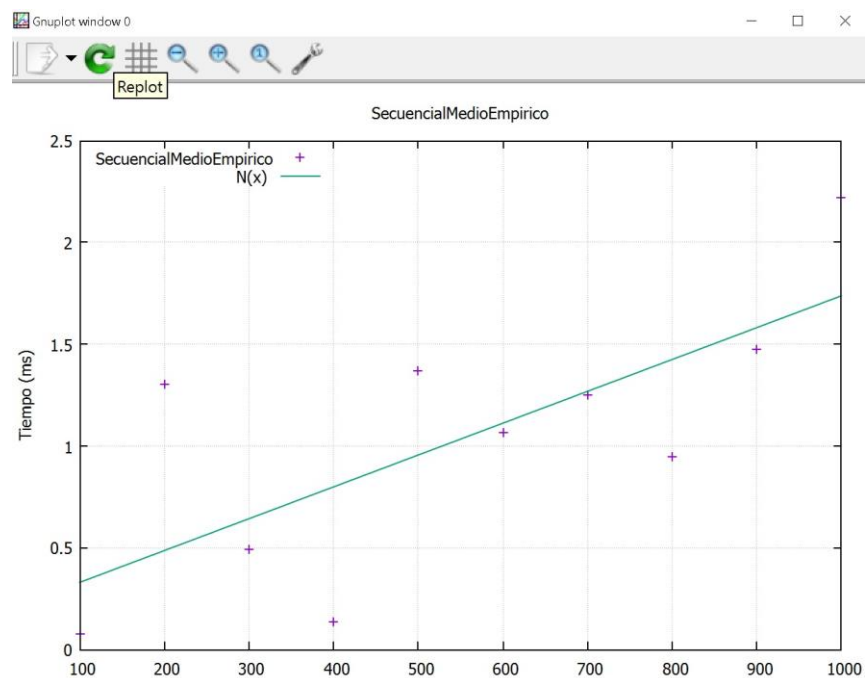
Caso Medio:

Busqueda SecuencialSecuencialMedio EmpiricoTiempos de ejecucion

Talla	Tiempo (ms)
100	7.9e-05
200	0.0013
300	0.00049
400	0.00014
500	0.0014
600	0.0011
700	0.0012
800	0.00095
900	0.0015
1000	0.0022

Datos guardados en el fichero SecuencialMedioEmpirico.dat

Generar grafica de resultados? (s/n):



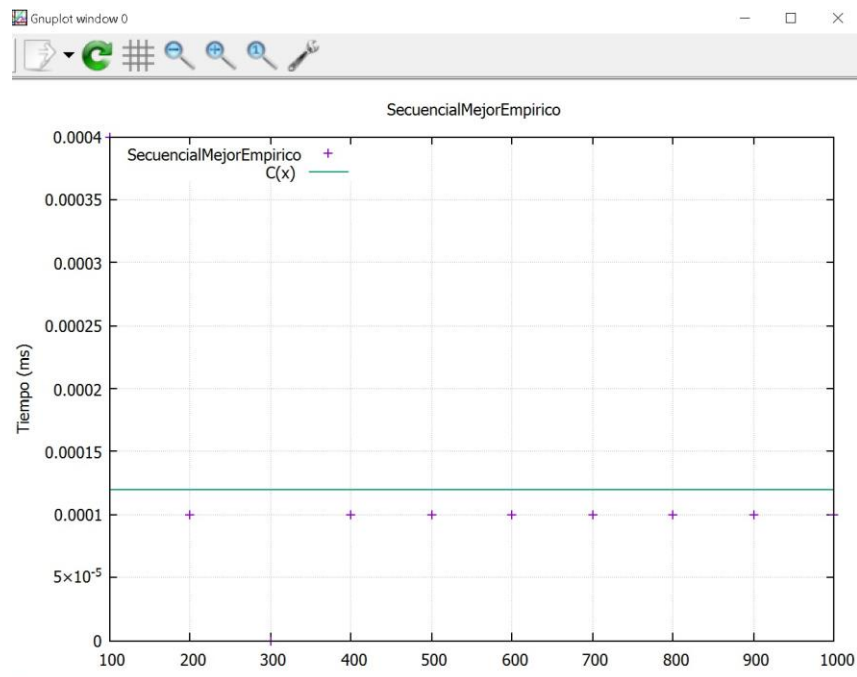
Caso Mejor:

```
Busqueda SecuencialSecuencialMejor EmpiricoTiempos de ejecucion

      Talla      Tiempo (ms)

tiempo = 0.0004milisegundos
      100      4e-07
tiempo = 0.0001milisegundos
      200      1e-07
tiempo = 0milisegundos
      300      0
tiempo = 0.0001milisegundos
      400      1e-07
tiempo = 0.0001milisegundos
      500      1e-07
tiempo = 0.0001milisegundos
      600      1e-07
tiempo = 0.0001milisegundos
      700      1e-07
tiempo = 0.0001milisegundos
      800      1e-07
tiempo = 0.0001milisegundos
      900      1e-07
tiempo = 0.0001milisegundos
     1000      1e-07

Datos guardados en el fichero SecuencialMejorEmpirico.dat
```



Comparación entre resultados teóricos y experimentales

Podemos observar que existen leves incongruencias entre la teoría y la práctica, pero esto se debe a que la teoría siempre está basada en el caso ideal, muy alejado de la realidad, entrópica, la cual siempre presenta imperfecciones debido a una multitud de variables externas que no se nos es posible controlar. En este caso, por ejemplo, dependiendo de la potencia del procesador, o la cantidad de procesos que se están ejecutando en el ordenador que uses saldrán diferentes resultados.

Sin embargo, lo que sí es igual en los dos sistemas, teórico y empírico, es la relación que presentan el caso Mejor el caso Medio y el caso Peor entre si.

Dicha relación es que la pendiente del caso Peor es la más pronunciada, la del caso Peor es 0 y la del caso medio se encuentra entre las dos. Y esto es lo más importante.

Diseño de aplicación

La aplicación, en este caso sigue un diseño modular, estructurado de forma que hay una cabecera con los archivos Constantes.h, donde se encuentran las diferentes constantes que se usaran en el programa, Mtime.h TestAlgoritmo.h Conjuntoint.h, cuyos .cpp se encuentran en los archivos fuente junto al principal.cpp(main).

- Constantes.h: como ya se ha dicho anteriormente dentro de este fichero se han declarado las constantes que usará el programa, estas son secuencialpeor, secuencialmedio, secuencialmejor
- Mtime: es el encargado de devolver la diferencia entre tiempo final y tiempo inicial.
- TestAlgoritmo: contiene todos los métodos que usaremos para estudiar los diferentes casos, teórico y empírico.
- Conjunto: sirve para crear y modificar el tamaño del vector
- Principal: como su propio nombre indica es nuestro main, donde implementaremos el menú y pondremos en funcionamiento los metodos de TestAlgoritmo.

Conclusión final

Gracias a este proyecto, hemos podido ver de forma gráfica, en el sentido literal de la palabra, como interacciona un algoritmo ante diferentes inputs(entradas, estímulos), ya que al final de este proyecto se comprende perfectamente que no es lo mismo que el valor de entrada a buscar esté al final como que este en medio o no este. A parte, al ser el primer programa serio que se realiza en la carrera te abre la mente a como se hacen de verdad los programas, ya que hasta ahora operábamos siempre en el main y programas muy pequeños. Por último y volviendo al algoritmo en concreto, es importante darse cuenta de cómo se comporta un algoritmo a la hora de la práctica, o lo que es lo mismo, en la realidad, cosa que no habíamos tocado hasta ahora