



Instituto Politécnico Nacional Escuela Superior de Cómputo



Ejercicios 03: Implemente los algoritmos de búsqueda



Materia: Algoritmos y Estructuras de Datos
Profesor: Martínez Edgardo Franco



Alumno: González Joshua

Grupo: 2CM8

Fecha: 30/03/2023 ISC



Tablas del número de comparaciones realizadas para buscar un elemento “x” en una colección de elementos con los distintos algoritmos de búsqueda:

Número de comparaciones realizadas para N = 5,000,000							
Elemento para buscar	Encontrado	Posición lista desordenada	Posición lista ordenada	Búsqueda lineal desordenada	Búsqueda lineal ordenada	Búsqueda binaria iterativa	Búsqueda binaria recursiva
322486	NO	—	—	5000000	819	23	23
14700764	NO	—	—	5000000	34321	23	23
3128036	SÍ	A[1368035]	A[7394]	1368036	7395	22	22
6337399	NO	—	—	5000000	14799	23	23
61396	NO	—	—	5000000	165	23	23
10393545	NO	—	—	5000000	24260	23	23
2147445644	NO	—	—	5000000	4999909	22	22
1295390003	NO	—	—	5000000	3015079	22	22
450057883	NO	—	—	5000000	1048210	22	22
187645041	NO	—	—	5000000	437741	22	22
1980098116	NO	—	—	5000000	4610072	22	22
152503	NO	—	—	5000000	357	23	23
5000	NO	—	—	5000000	20	22	22
1493283650	SÍ	A[542694]	A[3475906]	542695	3475907	20	20
214826	NO	—	—	5000000	531	22	22
1843349527	SÍ	A[2415203]	A[4291037]	2415204	4291038	22	22
1360839354	NO	—	—	5000000	3167661	22	22
2109248666	NO	—	—	5000000	4911241	22	22
2147470852	NO	—	—	5000000	4999971	23	23
0	NO	—	—	5000000	1	22	22

Número de comparaciones realizadas para N= 10,000,000							
Elemento para buscar	Encontrado	Posición lista desordenada	Posición lista ordenada	Búsqueda lineal desordenada	Búsqueda lineal ordenada	Búsqueda binaria iterativa	Búsqueda binaria recursiva
322486	SI	A[8674905]	A[1611]	8674906	1612	22	22
14700764	NO	-	-	10000000	68497	23	23
3128036	SI	A[1368035]	A[14725]	1368036	14726	22	22
6337399	SI	A[6334598]	A[29544]	6334599	29545	22	22
61396	NO	-	-	10000000	343	24	24
10393545	SI	A[6936855]	A[48374]	6936856	48375	23	23
2147445644	NO	-	-	10000000	9999832	24	24
1295390003	SI	A[8313715]	A[6031093]	8313716	6031094	22	22
450057883	NO	-	-	10000000	2096704	23	23
187645041	NO	-	-	10000000	875979	23	23
1980098116	NO	-	-	10000000	9219791	23	23
152503	NO	-	-	10000000	725	23	23
5000	NO	-	-	10000000	36	23	23
1493283650	SI	A[542694]	A[6952607]	542695	6952608	24	24
214826	NO	-	-	10000000	1047	24	24
1843349527	SI	A[2415203]	A[8582390]	2415204	8582391	22	22
1360839354	NO	-	-	10000000	6335635	23	23
2109248666	SI	A[6630039]	A[9822539]	6630040	9822540	22	22
2147470852	SI	A[7902795]	A[9999942]	7902796	9999943	19	19
0	NO	-	-	10000000	1	23	23

Conclusión

Analizando los resultados de las tablas podemos concluir que la diferencia entre el desempeño de la búsqueda lineal y búsqueda binaria es realmente descomunal, siendo la búsqueda binaria mucho mejor que la búsqueda lineal. Claro, puede que en casos específicos una búsqueda es mejor que otra, es decir: supongamos que vamos a utilizar la búsqueda lineal, con la consulta de un solo elemento en un arreglo desordenado y el elemento que buscamos se encuentra en las primeras posiciones, aquí la búsqueda lineal será mejor, pero aun así en la práctica esto es poco probable, además de que día a día se hacen millones de consultas, y que la búsqueda binaria haga aproximadamente 20 comparaciones para encontrar un elemento en una colección de 10 millones de elementos deja opacada a que, si es que hay suerte, la búsqueda lineal encuentre el elemento a la primera, pues un cálculo o veinte cálculos son nada para la computadora, ahora comparemos millones de comparaciones por miles de consultas y veinte comparaciones por miles de consultas, podemos imaginarnos cuál operación le tomará más tiempo a la computadora. Claro, para que la búsqueda binaria funcione, la colección de elementos sobre la que vamos a trabajar se tiene que encontrar ordenada, es aquí donde podemos observar la importancia de conocer los algoritmos de ordenamiento y que siempre será más fácil trabajar con una colección ordenada, pues si estamos trabajando con cantidades de datos muy grandes, podemos aplicar los algoritmos más eficientes que ya vimos: como merge sort o quicksort que son capaces de ordenar millones de elementos en segundos. Por último, es importante observar que la complejidad en el peor de los casos de la búsqueda lineal es $O(N)$ por consulta, mientras que la complejidad de la búsqueda binaria es $O(N \log(N))$ si el arreglo no se encuentra ordenado y $O(\log(N))$ por consulta.

Funcionamiento del programa

Instrucciones de compilación:

```
Linux: gcc main.c busqueda_lineal.c busqueda_binaria.c quickSort.c -o main -std=c17
```

Compilación del programa

```
nomad@Android:~/Documents/C_projects/ejercicios03$ gcc main.c busqueda_lineal.c busqueda_binaria.c quickSort.c -o main -std=c17
```

Ejemplo de compilación en terminal de linux

Mi programa funciona tomando argumentos de entrada:

1. Tamaño de los datos a leer (N).
2. Inicial del algoritmo de búsqueda a utilizar ("l" o "b").
3. Inicial del tipo búsqueda, es decir: si la búsqueda lineal se realizará en una colección desordenada u ordenada de elementos, si la búsqueda binaria que se realizará será iterativa o recursiva ("d" u "o", "i" o "r").
4. Elemento para buscar en la colección (x).

A continuación, se muestra un ejemplo de la ejecución del programa utilizando los diferentes algoritmos de búsqueda para un valor arbitrario x:

Ejecución del programa

```
nomad@Android:~/Documents/C_projects/ejercicios03$ ./main 10000000 l d 322486 < numeros10millones.txt
Busqueda lineal en una lista desordenada de 10000000 elementos: Elemento [322486] encontrado en la posicion A[8674905] - con 8674906 comparaciones realizadas
nomad@Android:~/Documents/C_projects/ejercicios03$ ./main 10000000 l o 322486 < numeros10millones.txt
Busqueda lineal en una lista ordenada de 10000000 elementos: Elemento [322486] encontrado en la posicion A[1611] - con 1612 comparaciones realizadas
nomad@Android:~/Documents/C_projects/ejercicios03$ ./main 10000000 b i 322486 < numeros10millones.txt
Busqueda binaria iterativa en una lista ordenada de 10000000 elementos: Elemento [322486] encontrado en la posicion A[1611] - con 22 comparaciones realizadas
nomad@Android:~/Documents/C_projects/ejercicios03$ ./main 10000000 b r 322486 < numeros10millones.txt
Busqueda binaria recursiva en una lista ordenada de 10000000 elementos: Elemento [322486] encontrado en la posicion A[1611] - con 22 comparaciones realizadas
nomad@Android:~/Documents/C_projects/ejercicios03$
```

Ejemplo de la ejecución del programa en terminal de linux