

Tarea 45: Algoritmos de búsqueda

El objetivo de la tarea es comparar la eficiencia entre distintos algoritmos de búsqueda, concretamente entre los algoritmos de búsqueda secuencial y búsqueda binaria.

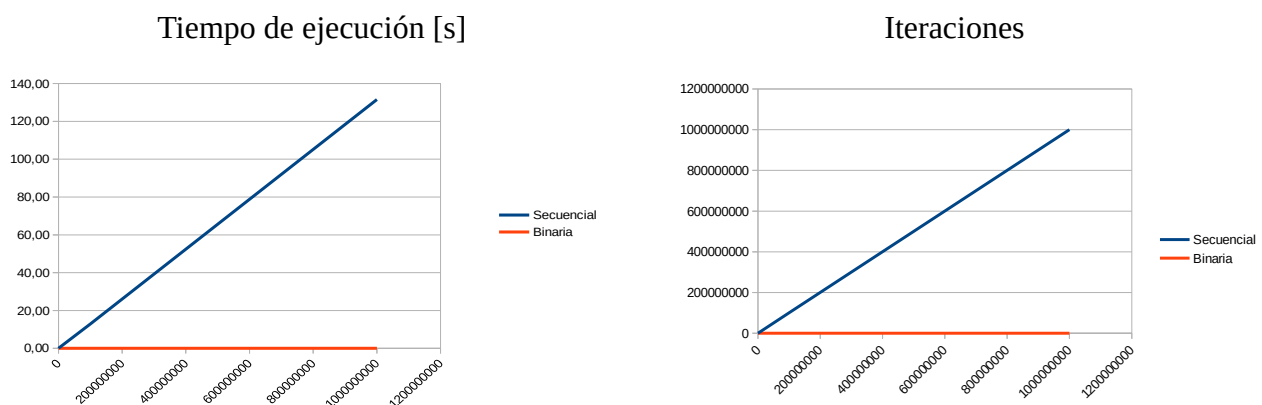
El ejemplo que se presenta en la tarea (lista = [3,56,21,33,874,123,66,1000,23,45,65,56]) es una lista relativamente corta, es decir, la cantidad de datos de entrada es pequeña. Si aplicamos el algoritmo para buscar el número 874 utilizando ambos algoritmos obtenemos el siguiente resultado:

SECUENCIAL --> Posición: 9 (numero de iteraciones: 10 y 2.6226043701171875e-06 segundos)

BINARIA --> Posición: 9 (numero de iteraciones: 3 y 2.6226043701171875e-06 segundos)

En este caso, se puede apreciar que la diferencia en iteraciones es mucho más notable que la diferencia en tiempo de ejecución. Esto es debido a que la variable de entrada tiene un tamaño bastante pequeño (12 datos) y, por tanto, prácticamente no hay diferencias entre el tiempo de ejecución de ambos. Sin embargo, el número de iteraciones sí nos está dando pistas sobre la diferencia de la eficiencia de ambos algoritmos: 3 iteraciones con el algoritmo binario en comparación con las 10 iteraciones del secuencial.

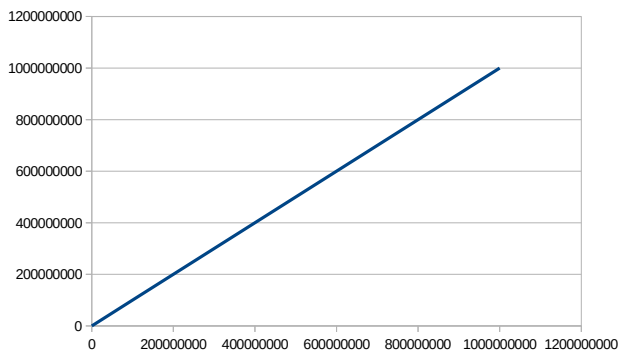
Para un análisis más exhaustivo se ha utilizado un dato de entrada más grande: se ha cuantificado la ejecución de ambos algoritmos para listas de entrada de 100, 1000, 10.000, 100.000 y 1.000.000 de elementos. El elemento a buscar siempre ha sido el último elemento de la lista, una vez verificado que el buscar este elemento coincide con el peor caso de ejecución.



Se puede ver claramente que la diferencia entre ambos algoritmos aumenta de forma drástica conforme aumenta el tamaño de los datos de entrada. En la gráfica comparativa se puede ver que el algoritmo secuencial aumenta de manera lineal con el tamaño del array de entrada y el algoritmo binario apenas aumenta (en esta escala, en comparación con el secuencial). Por otro lado, se puede ver que el tiempo de ejecución y el número de iteraciones son equivalentes a la hora de valorar la eficiencia del algoritmo.

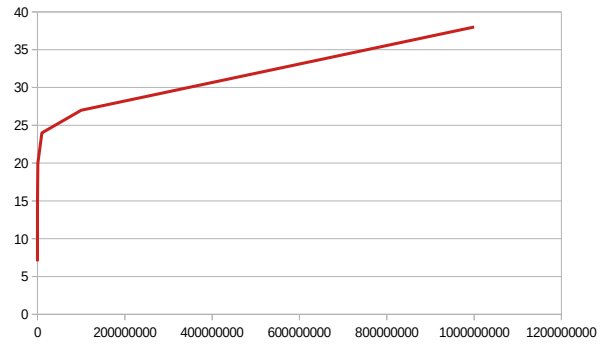
Si miramos cada algoritmo por su lado, en las gráficas inferiores tenemos el resultado del algoritmo secuencial a la izquierda y el del binario a la derecha. Como hemos verificado en los gráficos anteriores, podemos observar que la búsqueda secuencial tiene una relación lineal entre el tamaño de la entrada y el número de iteraciones que se necesitan para encontrar la solución. Por tanto, se confirma que la notación O-Grande del algoritmo secuencial es $O(N)$.

Iteraciones - Búsqueda secuencial



$O(N)$

Iteraciones - Búsqueda binomial



$O(\log N)$

En el caso del algoritmo binomial, la relación es logarítmica entre el tamaño de la variable de entrada y el número de la iteraciones. Por tanto, se confirma la notación O-Grande del algoritmo de búsqueda binomial: $O(\log N)$: el aumento del tamaño del array de entrada afecta mucho menos a la ejecución del programa y por tanto, el algoritmo binario es mucho más eficiente que el secuencial.