Análisis de Algoritmos 2018/2019

Práctica 3

Nicolás Serrano Salas

Miguel Ángel Luque López

Grupo 1212

Código	Gráficas	Memoria	Total

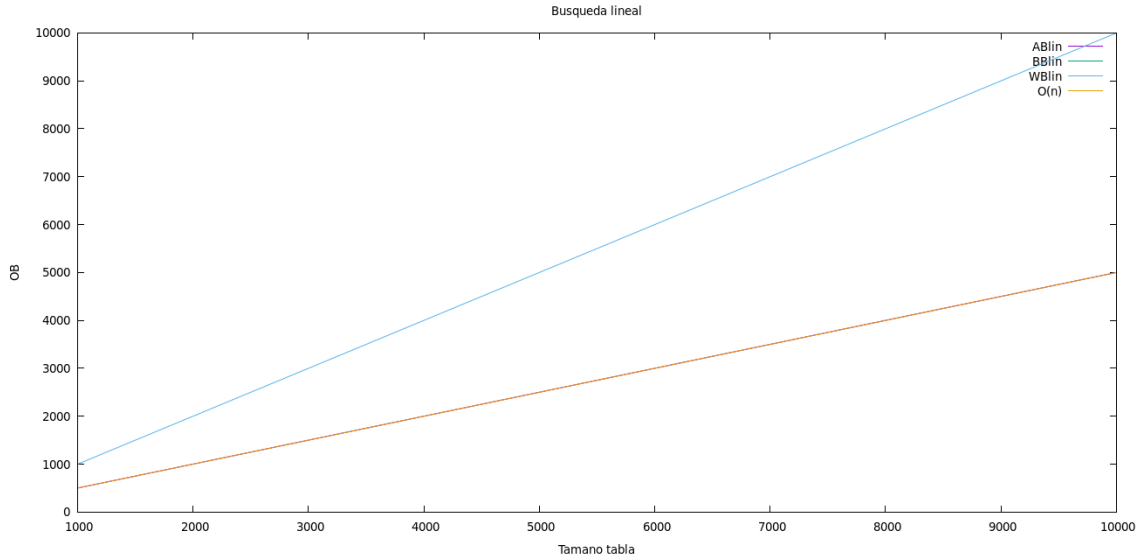
5. Resultados, Gráficas

5.2 Apartado 2

Gráfica con el número máximo, mínimo y medio de OBs de la búsqueda lineal.

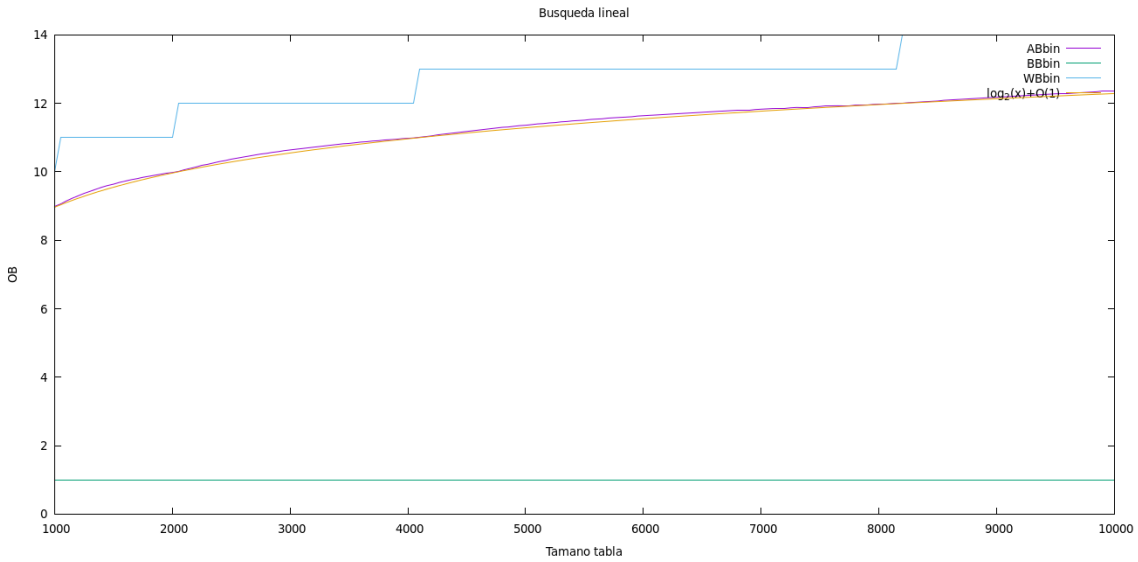
BBlin está muy pegado al cero como para ser visible, pues es del orden de uno.

En esta gráfica también cabe declarar que O(n) toma un valor de N/2 que es a lo que tiende el caso medio de búsqueda lineal.



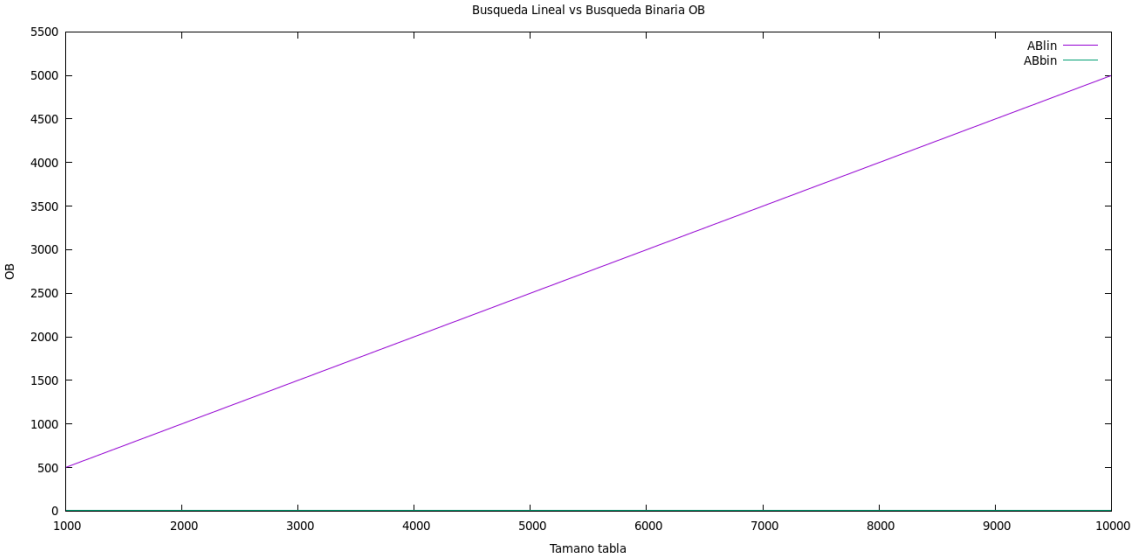
Gráfica con el número máximo, mínimo y medio de OBs de la búsqueda binaria.

Los saltos que se observan en el peor caso se producen a causa de un aumento en la profundidad del árbol, que ocurre al aumentar el tamaño de la tabla en una de las ejecuciones, después de este salto el número de operaciones básicas se estabiliza hasta el próximo aumento de profundidad.en el "$\log_2(x)+O(1)$" que define el caso medio O(1) toma el valor de -1 para ajustarse al valor al que tiende este caso medio.



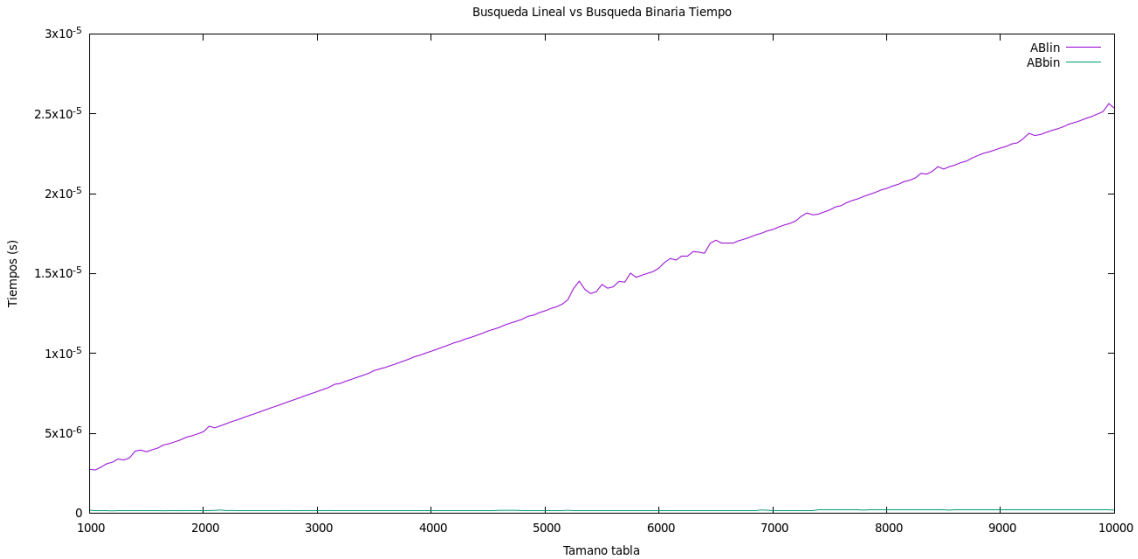
Gráfica comparando el número promedio de OBs entre la búsqueda lineal y la búsqueda binaria, comentarios a la gráfica.

Como vemos el número de operaciones básicas promedio de la búsqueda lineal se representa como una recta, sin embargo el caso de la búsqueda binaria tiende a crecer cada vez más lentamente, al ser un logaritmo, esto hace que las OBs de la búsqueda binaria crezcan a un ritmo extremadamente inferior, viéndose así en la parte inferior de la tabla con un número de operaciones ridículo en comparación a las de búsqueda lineal



Gráfica comparando el tiempo promedio de reloj entre la búsqueda lineal y la búsqueda binaria, comentarios a la gráfica.

En esta gráfica observamos el mismo efecto de antes, la búsqueda binaria es mucho más eficiente, cabe destacar que además el tiempo de ejecución de la búsqueda binaria es tan pequeño que nos fue necesario modificar nuestra función de medición de tiempos para poder medir algo superior a cero.



5. Respuesta a las preguntas teóricas.

Aquí respondéis a las preguntas teóricas que se os han planteado en la práctica.

5.1 Pregunta 1

En ambos casos la operación básica es la comparación de claves, que en el código estaría localizado en el fragmento: “if(clave == tabla[(*ppos)])”

5.2 Pregunta 2

Bbbin: O(1)

Bblin: O(N)

Wbbin: log(N)+O(1)

Wblin: O(N)

5.3 Pregunta 3

Abbin = log(N) + O(1)

Ablin = O(N)

5.4 Pregunta 4

El algoritmo de búsqueda lineal comienza seleccionando el primer elemento de la tabla, a partir de este irá comparando uno por uno secuencialmente y en el orden del array hasta encontrar el elemento que busca, esto hace que como máximo pueda comparar la clave con todos los elementos y como mínimo solo con el primero, siendo entonces la media de comparaciones la mitad del número de elementos.

El algoritmo de búsqueda binaria en cambio empezará por el elemento en la mitad del array de la tabla, en esta ocasión la tabla estará ordenada, así que en función de si el elemento es menor o mayor el siguiente elemento a comparar será la mitad de la mitad izquierda o la mitad de la mitad derecha correspondientemente, este proceso continuará hasta que no queden divisiones posibles o se encuentre el elemento que se busca. Este algoritmo a diferencia del anterior en ningún caso tendrá que comparar la clave con todos los elementos, pues con cada ciclo reduce a la mitad los elementos de la tabla en los que buscar, siendo así el máximo de comparaciones log(N)+ O(1).