

Instituto Politécnico Nacional

Escuela Superior de Cómputo



Análisis de algoritmos

Practica 02 : Análisis temporal y notación de orden (Algoritmos de búsqueda)

M. en C. Edgardo Adrián Franco Martínez

<http://www.eafranco.com>

edfrancom@ipn.mx

[@edfrancom](#) [f edgardoadrianfrancom](#)





Contenido

- Definición del problema
- Actividades
- Observaciones
- Reporte de práctica
- Rubrica de evaluación del reporte
- Entrega vía Web
- Fechas de Entrega



Definición del problema

- **Con base en el ordenamiento obtenido a partir del archivo de entrada de la practica 01 que tiene 10,000,000 de números diferentes. Realizar la búsqueda de elementos bajo 3 métodos de búsqueda, realizar el análisis teórico y experimental de las complejidades; así como encontrar las cotas de los algoritmos.**

- Búsqueda lineal o secuencial
- Búsqueda binaria o dicotómica
- Búsqueda en un árbol binario de búsqueda
- Implementación de las tres búsquedas con su variante de Threads





Actividades

1. **Programar en ANSI C**, cada uno de los 3 algoritmos de búsqueda mencionados y su variante con hilos (Elegir entre 2 y 4 Threads).



2. Adaptar los programas para que sean capaz de recibir un parámetro “n” que indica el numero de enteros a considerar para el **espacio de búsqueda** a partir de un archivo con un máximo 10,000,000 de **números ordenados**, en el caso del árbol binario de búsqueda, trabajar con el árbol resultante de la practica 01.
3. Realizar un **análisis teórico** de la **complejidad temporal de cada algoritmo** (*Mejor, peor y medio; tal cual se implementara*).



4. Realizar un registro del tiempo para cada algoritmo y su variante con hilos buscando diferentes números en varios tamaños de problema:

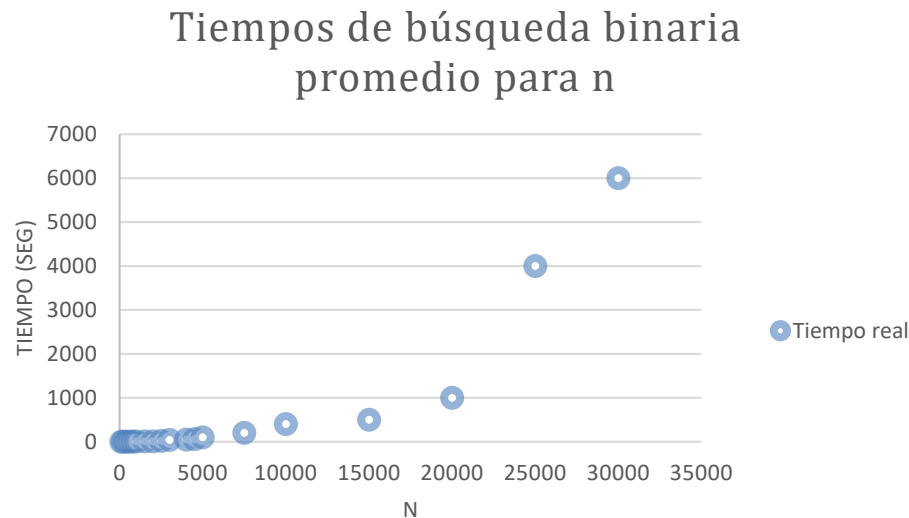
- Los números A=[322486, 14700764, 3128036, 6337399, 61396, 10393545, 2147445644, 1295390003, 450057883, 187645041, 1980098116, 152503, 5000, 1493283650, 214826, 1843349527, 1360839354, 2109248666 , 2147470852 y 0].
- Considerar para tamaño de problema “n” primeramente los primeros 100, 1000, 5000, 10000, 50000, 100000, 200000, 400000, 600000, 800000, 1000000, 2000000, 3000000, 4000000, 5000000, 6000000, 7000000, 8000000, 9000000 y 10000000 elementos del arreglo.
- Registrar los tiempos de búsqueda registrados para cada n considerado de cada algoritmo y su variante de Hilos si a elegido desarrollarlos.

Búsqueda binaria (Sin hilos)

Número a buscar	Tamaño de n	Tiempo real	Encontrado
322486	1000	0.23 s	no
322486	35000	1.11 s	no
322486	75000	2.23 s	si



5. Graficar el comportamiento temporal de cada algoritmo, considerando el tiempo de búsqueda promedio de todos los números $A[]$ en cada n .



6. Graficar una comparativa de los 3 algoritmos y sus variantes con hilos considerando el tiempo promedio de búsqueda de cada uno de ellos para cada tamaño de problema n .



7. Realizar una **aproximación polinomial** del comportamiento temporal (tiempo real promedio), de cada uno de los algoritmos probados según el punto 4.

- Encontrar para cada algoritmo un polinomio que refleje de mejor manera su comportamiento.

8. Mostrar gráficamente la comparativa de las aproximaciones de cada algoritmo y determinar si el comportamiento experimental se aproxima a lo esperado teóricamente.

9. Determine **según sus pruebas y el análisis teórico del peor caso, cuanto le lleva a la computadora ejecutar cada paso básico** de cada algoritmo. ***(Aproximar la función teórica a las aproximadas)***



10. Determine con base en el análisis teórico y la aproximación experimental obtenida cual será el tiempo de búsqueda de cada algoritmo para un tamaño n de 50000000, 100000000, 500000000, 1000000000 y 5000000000. *(Considere el tiempo obtenido en el punto 9 para estimar a partir del análisis teórico del peor caso).*

11. Indique las cotas O mayúscula para cada algoritmo con base en el análisis teórico del peor caso.

12. Indique las cotas O mayúscula para cada aproximación polinomial seleccionada y justifique.



11. Finalmente responda a las siguientes preguntas:

- i. ¿Cuál de los 3 algoritmos es más fácil de implementar?
- ii. ¿Cuál de los 3 algoritmos es el más difícil de implementar?
- iii. ¿Cuál de los 3 algoritmos es el más difícil de implementar en su variante con hilos?
- iv. ¿Cuál de los 3 algoritmos en su variante con hilos resulto ser mas rápido?
- v. ¿Cuál algoritmo tiene menor complejidad temporal?
- vi. ¿Cuál algoritmo tiene mayor complejidad temporal?
- vii. ¿El comportamiento experimental de los algoritmos era el esperado? ¿Por que?
- viii. ¿Sus resultados experimentales difieren mucho de los análisis teóricos que realizo? ¿A que se debe?
- ix. ¿Los resultados experimentales de las implementación con hilos de los algoritmos realmente tardaron $F(t)/\#hilos$ de su implementación sin hilos?
- x. ¿Cuál es el % de mejora que tiene cada uno de los algoritmos en su variante con hilos? ¿Es lo que esperabas? ¿Por qué?
- xi. ¿Existió un entorno controlado para realizar las pruebas experimentales? ¿Cuál fue?
- xii. ¿Si solo se realizará el análisis teórico de un algoritmo antes de implementarlo, podrías asegurar cual es el mejor?
- xiii. ¿Qué tan difícil fue realizar el análisis teórico de cada algoritmo?
- xiv. ¿Qué recomendaciones darían a nuevos equipos para realizar esta practica?

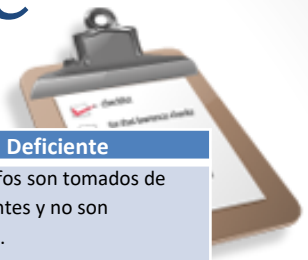




Observaciones

- El análisis teórico se realizara a priori (es decir antes de implementar cualquier algoritmo), por lo tanto el **día de laboratorio deberán mostrar estos análisis**.
- Utilizar solo ANSI C, los programas deberán de funcionar en Linux.
- Indique cual fue su plataforma experimental (Características del hardware, compilador, sistema operativo, entorno controlado, etc.)
- Se sugiere crear scripts que faciliten la experimentación.
- En el laboratorio mostrar el análisis teórico y funcionamiento de los algoritmos
 - Autodocumentación del código
 - Documentación de funciones y algoritmos
- Las variantes empleando hilos, deberá reportar como funcionan sus particiones de las búsquedas y los tiempos de búsqueda considerando el tiempo de CPU de la búsqueda de todo el programa y en cada hilo. Este análisis y conclusiones permitirán definir el % de mejora de la implementación con hilos.





Rubrica de evaluación del reporte

Indicador	Excelente	Muy bien	Bien	Deficiente
Construcción de párrafos	Todos los párrafos incluyen una introducción, explicaciones o detalles y una conclusión	Los párrafos incluyen información relacionada pero no fueron generalmente bien organizados	La estructura del párrafo no estaba clara y las oraciones no estaban generalmente relacionadas	Los párrafos son tomados de otras fuentes y no son originales.
Redacción	No hay errores de gramática, ortografía y puntuación y la redacción es coherentemente	No hay errores de gramática, ortografía y puntuación, pero la redacción presenta incoherencias	Pocos errores de gramática, ortografía y puntuación	Muchos errores de gramática, ortografía y puntuación
Cantidad de información <small>Portada, Introducción, Planteamiento del problema, algoritmos e implementación, actividades y pruebas, errores detectados, posibles mejoras, conclusiones y anexos</small>	Todos los temas son tratados de manera clara y precisa, según lo solicitado.	La mayoría de los temas son tratados de manera clara y precisa	Dos temas no están tratados o están imprecisos y no cumplen lo solicitado.	Tres o más temas no están tratados o están imprecisos y no cumplen lo solicitado.
Calidad de la información	La información está claramente relacionada con el tema principal y proporciona varias ideas secundarias y/o ejemplos	La información da respuestas a las preguntas principales, y solo da algunos detalles y/o ejemplos	La información da respuestas a las preguntas principales, pero no da detalles y/o ejemplos	La información tiene poco o nada que ver con las preguntas planteadas.
Algoritmos	Los algoritmos dan solución apoyándose de pseudocódigo, diagramas y/o figuras en un lenguaje claro.	La mayoría de los algoritmos dan solución apoyándose de pseudocódigo, pero diagramas y/o figuras.	Los algoritmos son mencionados textualmente pero no se describen	Los algoritmos no son expresados en el reporte.
Organización	La información está muy bien organizada con párrafos bien redactados y con subtítulos con estilos adecuados	La información está organizada, pero no se distingue en estilos adecuados	La información está organizada, pero los párrafos no están bien redactados	La información proporcionada no parece estar organizada o es copiada de referencias externas de manera literal





Reporte de practica



- Portada
- Planteamiento del problema
- Actividades y Pruebas (Verificación de la solución, pruebas y resultados de la práctica según lo solicitado)
- Errores detectados (Si existe algún error detectado, el cuál no fue posible resolver o se desconoce el motivo y solo ocurre con ciertas condiciones es necesario describirlo)
- Anexo (Códigos fuente *con colores e instrucciones de compilación)
- Bibliografía (En formato IEEE)





Entrega vía Web



Grupo	Contraseña
3CM2	Analisis3cm2
3CM3	analisis3cm3

- **En un solo archivo comprimido** (ZIP, RAR, TAR, JAR o GZIP)
 - Reporte (DOC, DOCX o PDF)
 - Códigos fuente (.C, .H, etc.)
 - **Código documentado:** Titulo, descripción, fecha, versión, autor.
 - *(Funciones y Algoritmos: ¿Qué hace?, ¿Cómo lo hace?, ¿Qué recibe?, ¿Qué devuelve?, ¿Causa de errores?).*
 - OBSERVACIONES
 - *NO enviar ejecutables o archivos innecesarios, las instrucciones de compilación van en el anexo del reporte. (Yo compilare los fuente).
 - *NO enviar archivo de números



Fechas de entrega



- **Demostración** *Laboratorio de Programación 2 (2107)*

- 3CM2 “Miércoles 19 de septiembre o miércoles 26 de septiembre de 2018”.
- 3CM3 “Lunes 17 de septiembre o lunes 24 de septiembre 2018”.



- **Entrega de reporte y código**

- En un solo archivo comprimido.



- **Fecha y hora limite de entrega vía Web**

- Miércoles 03 de octubre de 2018 a las 23:59:59 hrs.

