Tarea 1

Búsqueda Binaria y Doblada

Jérémy Barbay, Mauricio Quezada

2011-03-28 Lunes → 2011-04-13 Miércoles

CC4102/CC40A/CC53A: Diseño y Análisis de Algoritmos – 2011

Profesor: Jérémy Barbay <jbarbay@dcc.uchile.cl>
Auxiliar: Mauricio Quezada <mquezada@dcc.uchile.cl>
Ayudantes: David Contreras <dcontrer@dcc.uchile.cl>
Sergio Villarroel <svillarr@dcc.uchile.cl>

Resumen

El objetivo de esta tarea es comparar el rendimiento de diferentes algoritmos de búsqueda en conjuntos ordenados. Se consideran cuatro algoritmos (dos variantes de la búsqueda binaria, búsqueda doblada y búsqueda secuencial) y dos experimentos (posición p con probabilidad $1/n$ y posición p con probabilidad proporcional a $1/p$). Se espera que el alumno implemente individualmente los algoritmos, los evalúe en la práctica y presente los resultados de sus experimentos en un reporte completo (e.g. hipótesis, diseño experimental, descripción de las medidas de rendimiento elegidas, presentación e interpretación de los resultados, conclusión).

1. Descripción de la tarea

El objetivo de esta tarea es implementar distintos algoritmos de búsqueda y comparar su rendimiento en la práctica.

2. Algoritmos

Se consideran cuatro algoritmos, que buscan un elemento x en un arreglo de tamaño n , ubicado en la posición p :

1. **Búsqueda binaria con $2\lceil\lg n\rceil$ comparaciones**, que en cada etapa hace una comparación $=$ entre los dos elementos considerados, seguida de una comparación $<$ si no son iguales.
2. **Búsqueda binaria con $1 + \lceil\lg n\rceil$ comparaciones**, que en cada etapa hace una comparación $<$ entre los dos elementos considerados, y hace una comparación $=$ solamente cuando el rango de inserción del elemento x es definido.
3. **Búsqueda doblada con $2 + 2\lceil\lg p\rceil$ comparaciones**, que encuentra en $1 + \lceil\lg p\rceil$ comparaciones un intervalo de tamaño $p/2$ que contiene a x , y encuentra la posición p en $1 + \lceil\lg p\rceil$ comparaciones adicionales (usando la segunda variante de la búsqueda binaria).
4. **Búsqueda secuencial con p comparaciones**, que prueba cada posición en orden.

3. Instancias

Se consideran dos tipos de instancias (que corresponden a dos experimentos distintos):

1. **Al azar uniformemente**: instancias donde la posición de x es igual a p con probabilidad $\frac{1}{n}$; y
2. **Al azar con respecto al inicio**: instancias donde la posición de x es igual a p con probabilidad $\frac{a}{b+p}$, $a, b > 0$. El alumno deberá determinar las constantes adecuadas.

Note que la probabilidad es tomada no sobre los valores del arreglo, sino sobre la posición donde se encuentra el elemento x .

Se consideran instancias de tamaño creciente exponencialmente, desde $2^{10} = 1024$ hasta $2^{20} = 1048576$ elementos, o hasta que los límites de memoria de la máquina usada, o del tiempo de experimentación lo permitan. Cada una de las instancias se deberá repetir 100 veces, y sobre los resultados obtenidos se deberá documentar el máximo, mínimo, y promedio de las medidas de rendimiento.

4. Medidas de Rendimiento

Para esta tarea sólo se considerará como medida de rendimiento la cantidad de comparaciones realizadas por los algoritmos.

5. Implementación

Se puede elegir el lenguaje de programación de su preferencia, aunque se sugiere Java, C o Python. Se debe describir la máquina usada por la experimentación, en términos de:

- sistema operativo,
- cantidad de memoria (activa y en caché).
- cantidad y características de los procesadores.

6. Reporte

El reporte debe seguir los pasos descritos en clase. En particular, el reporte debe permitir de identificar claramente los puntos siguientes:

1. las **hipótesis** elegidas *antes* del inicio de la realización del experimento,
2. el **diseño experimental**, incluyendo los detalles de la implementación de los algoritmos, de la generación de las instancias y la descripción del proceso de medida de rendimiento
3. la **presentación de los resultados** en forma de una descripción textual, tablas y gráficos en dos dimensiones,
4. la **interpretación** de los resultados,

5. y, por último, **conclusiones** criticando sus resultados y proponiendo experimentos adicionales.

En el caso donde los datos no verifican la hipótesis, no se pide modificar el ciclo de diseño, sino de correctamente identificar el problema y presentarlo en la conclusión (e.g. sea una error en la hipótesis, en el diseño experimental o en su implementación).

Se espera que el informe tenga una extensión de *máximo* 4 páginas, (sin considerar anexos) y que incluya 4 gráficos 2D que sumerizen los resultados más importantes. Si tienen más gráficos se pueden colocar en la sección Anexos.

Se puede elegir el software de redacción de su preferencia, aunque se sugiere el uso de L^AT_EX. El reporte debe ser entregado en U-Cursos en un archivo de formato pdf con el reporte textual, y un archivo zip con la totalidad del código usado para los experimentos con un archivo README con instrucciones de cómo ejecutarlo, de manera que potencialmente se puedan reproducir los experimentos.

7. Colaboración y Uso de Librerías

El proyecto es individual: uno de los objetivos es desarrollar las capacidades de programación de los alumnos. La comunicación entre alumnos sobre el proyecto está autorizada, en límites razonables. En particular, cada alumno debe ser capaz de describir los detalles mas técnicos de las implementaciones y experimentos descritos en su reporte. De igual manera, el uso de librerías de software es autorizado, solamente si el alumno es capaz de describir los detalles más técnicos de la implementación de todas las funciones involucradas en la librería. Se pedirá una evaluación oral en caso de duda sobre la originalidad del trabajo.

8. Atraso y otros problemas

Con la finalidad de dar un *feedback* antes de la entrega de la tarea siguiente, los ayudantes tienen solamente una semana para corregir las tareas. Por lo tanto, no se admite atrasos en la entrega de tareas, en general. Los atrasos o dificultades para la entrega de las tareas con excusas válidas se negocian **hasta 24 horas antes** de la fecha de entrega, y directamente con el auxiliar encargado de la tarea (definitivamente **NO** con el profesor del curso). Todo atraso o dificultad para la entrega sin negociación o excusa válida se convierte en un **1 (uno)**.