

CC4102 Diseño y Análisis de Algoritmos

Tarea 1

Prof. Benjamin Bustos

Aux.: Iván Sipirán

Fecha de entrega: Lunes 22 de abril de 2013 a las 23:59

El objetivo de esta tarea es analizar teórica y experimentalmente tres algoritmos para encontrar la mediana de un conjunto de elementos. Los algoritmos son:

- QuickSelect: Usa el algoritmo de particiones usado en QuickSort para encontrar un pivote aleatorio. Luego, el algoritmo coloca los elementos menores antes del pivote y los mayores después. Como el pivote queda en su posición correspondiente si el arreglo estuviera ordenado, sirve de referencia para encontrar la mediana. De acuerdo a la posición del pivote, se realiza un llamado recursivo a la parte izquierda o derecha del arreglo.
- Mediana de medianas: Este algoritmo se diferencia del QuickSelect en la forma de elegir el pivote. Para este caso, se toman grupos de $2k + 1$ elementos (k es un parámetro del método) y se les encuentra su mediana. Luego, se calcula recursivamente la mediana de medianas, y este elemento es utilizado como pivote.
- Algoritmo de Musser: Este algoritmo es otra variación de QuickSelect. Si después de realizar t llamados recursivos del algoritmo QuickSelect la fracción del subproblema a resolver aún no es α veces más pequeño que el problema original (con α un entero pequeño), se cambia el algoritmo de búsqueda a mediana de medianas (t y α son parámetros del método).

Esta tarea consta de dos partes: Teórica y Experimental

Parte Teórica

- Analizar el peor caso y caso promedio de QuickSelect.
- Analizar el peor caso de mediana de medianas, tomando grupos de $2k + 1$ elementos. Encuentre analíticamente el k óptimo y compruebe que el algoritmo es $\Omega(n)$ cuando $k = 1$. Como algoritmo para encontrar la mediana de los $2k + 1$ elementos utilice ordenación por inserción.
- Para el Algoritmo de Musser, analice el peor caso y caso promedio considerando el k óptimo encontrado para mediana de medianas.

Parte Experimental

- Implemente los tres algoritmos.
- Muestre con gráficos los tiempos de ejecución y número de comparaciones de cada algoritmo, considerando el valor óptimo de k para mediana de medianas. Las pruebas se deben realizar sobre arreglos aleatorios de tamaño 100,000 hasta 1,000,000, repitiendo 100 veces cada experimento. Considere intervalos de 50,000 elementos para los experimentos (es decir: 100,000; 150,000; 200,000; etc.).
- Además, se le pide validar experimentalmente el valor óptimo de k encontrado en el análisis teórico, así como también los valores óptimos experimentales de t y α para el Algoritmo de Musser. Los valores óptimos experimentales son aquellos que minimizan el tiempo de ejecución del algoritmo. Para esta prueba considere arreglos aleatorios de tamaño 1,000,000, repita el experimento 100 veces y promedie los resultados.
- IMPORTANTE: Todos los gráficos deben mostrar barras de error (intervalo de confianza al 95 %).

NOTAS

- En el informe, los análisis teóricos deben ser detallados y bien explicados.
- Para la parte experimental, se espera que cada gráfico tenga su respectivo análisis y discusión (y no simplemente repetir lo que ya muestra el gráfico).

- Se pueden utilizar los siguientes lenguajes de programación: C/C++, Java o Python.
- Se debe entregar el informe (formato PDF) y código fuente (con una breve descripción de cómo compilar y ejecutar) a través de la plataforma U-Cursos, Sección Tareas. No se aceptarán tareas entregadas por otros medios.
- No se aceptarán tareas atrasadas.