CC4102 - Examen

Prof. Gonzalo Navarro

4 de Diciembre de 2012

P1 (2.0 pt)

Está usted en el comité de contratación de una prestigiosa universidad, para un concurso al que se han presentado varios candidatos. A éstos se les ha dado un año para que investiguen sobre un cierto problema P, para el cual sólo se conoce una cota inferior de $\Omega(n \log^2 n)$ y un algoritmo tipo Las Vegas de costo $O(n^3)$. Hoy los candidatos presentan sus resultados. Evalúe, a través del título de sus presentaciones, cuáles deben ser expulsados inmediatamente por los guardias por contradecir las cotas existentes, cuáles deben ser invitados amablemente a irse por no mejorar los resultados existentes, y cuáles deben contratarse inmediatamente por obtener los mejores resultados del grupo (es decir, dentro de los que aportan, los que no son superados en todos los aspectos por algún otro). Argumente.

1. "Una 2-aproximación de tiempo $O(n^4)$ para P".
2. "El problema P es $\Omega(n\sqrt{n}/\log n)$".
3. "Un algoritmo $O(n^3)$ en promedio para P, para entradas uniformemente distribuidas".
4. "P se puede resolver mediante n operaciones de costo amortizado $O(\log n)$".
5. "P es $O(n^6)$".
6. "Un esquema de aproximación de tiempo $O(\frac{1}{\epsilon^2} n^{2+2/\epsilon})$ para P".
7. "El problema P es $\Omega(n^{1+1/\sqrt{\log n}})$".
8. "Un algoritmo PRAM de tiempo $T(n,p) = O(n \log(n/p))$ para P".

P2 (2.0 pt)

Considere dos matrices esparsas de $n \times n$, almacenadas en disco en forma de una secuencia (i, j, v) para cada valor $A[i, j] = v \neq 0$. Se requiere obtener el producto de las dos matrices, pero se dispone solamente de una RAM de tamaño $M = \Theta(n)$. Diseñe y analice un algoritmo eficiente en términos de p y p', las cantidades de celdas no cero en las dos matrices (Hint: se puede conseguir básicamente $O(pp'/(MB))$).

P3 (2.0 pt)

Un *árbol α-balanceado*, para $1/2 < \alpha < 1$, es un árbol binario de búsqueda donde todo subárbol $T = (root, T_l, T_r)$ cumple $|T_l| \leq \alpha \cdot |T|$ y $|T_r| \leq \alpha \cdot |T|$. Las operaciones para buscar y mantener un árbol α-balanceado son las mismas que para un árbol binario de búsqueda, excepto que luego de insertar o borrar un nodo, se busca el nodo más alto en el camino del punto de inserción/borrado hacia la raíz, que no esté α-balanceado, y se lo reconstruye como árbol perfectamente balanceado (el costo es proporcional al tamaño del subárbol que se reconstruye).

1. Muestre que la búsqueda en un árbol α-balanceado cuesta $O(\log n)$, y que lo mismo ocurre con las inserciones y borrados, si no consideramos las reconstrucciones. ¿Qué constante obtiene multiplicando el $\log n$?

2. Muestre que el costo amortizado de las inserciones y borrados, ahora considerando reconstrucciones, es también $O(\log n)$. Para ello, considere la función potencial

$$\Phi(T) \;=\; \frac{1}{2\alpha - 1} \sum_{T' \in T} abs(|T'_l| - |T'_r| - 1),$$

donde $abs(\cdot)$ es el valor absoluto, y $T' \in T$ significa que T' es un subárbol de T.

P4 (2.0 pt)

Se tienen k arreglos apuntados por los punteros $L_1 \ldots L_k$, cada arreglo en un área distinta de memoria. También se sabe el largo de cada arreglo, $\ell_1 \ldots \ell_k$. Sea $n = \ell_1 + \ldots + \ell_k$. Se desea concatenar los k arreglos en orden en un gran arreglo L de largo n. Note que los ℓ_i pueden ser muy distintos.

Proponga un algoritmo CREW PRAM eficiente para resolver este problema y analícelo (tiempo, trabajo, speedup, eficiencia). Indique cuántos procesadores podría utilizar como mínimo para obtener el mejor tiempo posible de su solución.

Tiempo: 3 horas — Con dos hojas de apuntes