CC4102 - Diseño y Análisis de Algoritmos
Auxiliar 11

Prof. Gonzalo Navarro; Aux. Mauricio Quezada

29 de noviembre de 2012

1 Bin Packing (C3 2011/2)

El problema de *bin packing* consiste en empaquetar ítems de tamaños $X = \langle x_1, x_2, \ldots, x_n \rangle$ en la menor cantidad posible de cajas, donde éstas tienen un tamaño fijo b. Este problema es NP-completo, pero es fácil obtener una 2-aproximación.

Muestre que no existe un Esquema de Aproximación Polinomial (PTAS) para este problema a menos que P = NP. En particular, muestre que no es posible lograr un factor de aproximación menor a 3/2 en tiempo polinomial si P $\neq$ NP. Para ello, recuerde que el problema de partir X en dos conjuntos de igual suma es NP-completo.

2 Multiplicación de matrices (Ex. 2011/2)

Considere paralelizar el problema de multiplicar dos matrices de $n \times n$. Considere que $T(n,1) = O(n^3)$ (es decir, ignorando algoritmos mejores que el estándar).

1. Proponga un algoritmo de tiempo $O(n)$ usando n^2 procesadores. Calcule $T(n), W(n), T(n,p)$, y $E(n,p)$. ¿Qué modelo PRAM usa?
2. Proponga un algoritmo de tiempo $O(\log n)$ usando n^3 procesadores. Calcule $T(n), W(n), T(n,p)$, y $E(n,p)$.
3. Mejore la eficiencia del algoritmo anterior para que sea $\Theta(1)$. ¿Qué tiempo obtiene, y con cuántos procesadores?
4. (Propuesto) Modifique el algoritmo eficiente para que esté dentro del modelo EREW. ¿Cuál es la desventaja del nuevo algoritmo? ¿Cuál es la ventaja?

3 Parallel Search

Sea $X = \langle x_1, x_2, \ldots, x_n \rangle$ un conjunto de n elementos distintos de un conjunto S tal que $x_1 < x_2 < \ldots < x_n$.

Dado $y \in S$, queremos encontrar i tal que $x_i \leq y < x_{i+1}$. Agregamos $x_0 = -\infty$ y $x_{n+1} = +\infty$ por simplicidad.

Diseñe un algoritmo paralelo para buscar un y dado en X usando p procesadores. Determine $T(n,p)$.

4 Convex Hull

La *envoltura convexa* de un conjunto de puntos $S = \{p_1, p_2, \ldots, p_n\}$ de coordenadas $p_i = (x_i, y_i)$ es el polígono convexo más pequeño que cubre todos los puntos de S.

Proponga un algoritmo paralelo con $O(n)$ procesadores que calcule la envoltura convexa de un conjunto de puntos S. Para ello, considere calcular primero la *envoltura convexa superior* y luego la *inferior*, y unirlas para dar el resultado. ¿Puede extender esta idea recursivamente?. Calcule $T(n)$, $T(n,p)$ (para $O(n)$ procesadores), $W(n)$ y $E(n,p)$.