Machines P-RAM et Réseaux de tri
TD – Enoncé

Plan

1. Tri-fusion de Batcher
2. Principe du 0-1
3. Tri sur une grille 2D
4. Machine à trier P-RAM de Cole

1 Tri-fusion de Batcher

Réseau de fusion

- On note $\textsc{tri}(c_1, c_2 \ldots , c_n)$ la liste triée des c_i
- On écrit $\textsc{triée}(c_1, c_2 \ldots , c_n)$ pour $c_1 \leq c_2 \leq \cdots \leq c_n$
- Si $\textsc{triée}(a_1, \ldots , a_n)$ et $\textsc{triée}(b_1, \ldots , b_n)$, on note

$$\textsc{fusion}((a_1, \ldots , a_n), (b_1, \ldots , b_n)) = \textsc{tri}(a_1, \ldots , a_n, b_1, \ldots , b_n) .$$

1. Concevoir un réseau $\underline{\textsc{fusion}_1}$, ($\underline{\textsc{fusion}_2}$) à partir de comparateurs pour la fusion de deux listes triées de longueur 2 (2^2).
2. Montrer la propriété suivante :
   $\textsc{triée}(a_1, \ldots , a_{2n})$ et $\textsc{triée}(b_1, \ldots , b_{2n})$
   $$\begin{array}{rcl} (d_1, \ldots , d_{2n}) & = & \textsc{fusion}((a_1, a_3, \ldots , a_{2n-1}), (b_1, b_3, \ldots , b_{2n-1})) \\ (e_1, \ldots , e_{2n}) & = & \textsc{fusion}((a_2, a_4, \ldots , a_{2n}), (b_2, b_4, \ldots , b_{2n})) \end{array}$$
   $\Rightarrow$
   $$\textsc{triée}(d_1, \min(d_2, e_1), \max(d_2, e_1), \ldots , \\ \min(d_{2n}, e_{2n-1}), \max(d_{2n}, e_{2n-1}), e_{2n})$$
3. Définir par récurrence un réseau $\underline{\textsc{fusion}_n}$ pour la fusion de deux listes de longueur 2^n. Combien de comparateurs comprend ce réseau, et quel en est le temps de calcul ?

Réseau de tri

- Construire par récurrence un réseau $\underline{\textsc{tri}_n}$ pour trier 2^n éléments
- Analyser la complexité (temps de calcul, nombre de processeurs, coût).

2 Principe du 0-1

Enoncé

– Montrer qu'un réseau de comparateurs implante correctement le tri si et seulement si il calcule bien cette fonction pour toute suite de type 0-1 (i.e. constituée uniquement de 0 et de 1) en entrée
– En déduire une nouvelle preuve de la correction du réseau de fusion pair-impair de Batcher

3 Tri sur une grille 2D

Définitions

– On s'intéresse au tri par transposition pair impair sur une grille

– Un tableau carré $A = ((a_{i,j}))$ de taille $n \times n$, $n = 2^m$ est ordonné en serpent si les éléments du tableau sont ordonnés comme suit :

$$a_{2i,j} \leq a_{2i,j+1}, \quad \text{si} \quad 1 \leq j \leq n-1$$

$$a_{2i+1,j+1} \leq a_{2i+1,j}, \quad \text{si} \quad 1 \leq j \leq n-1$$

$$a_{2i-1,n} \leq a_{2i,n}, \quad \text{si} \quad 1 \leq i \leq 2^{m-1}$$

$$a_{2i,1} \leq a_{2i+1,1}, \quad \text{si} \quad 1 \leq j \leq 2^{m-1} - 1$$

– Ce serpent induit un réseau linéaire à l'intérieur de la grille

– Un *shuffle* transforme la séquence de $n = 2p$ éléments $< z_1, \ldots, z_n >$ en la séquence $< z_1, z_{p+1}, \ldots, z_p, z_{2p} >$.

Algorithme de fusion

– Données : 4 tableaux de taille $2^{m-1} \times 2^{m-1}$ ordonnés en serpent à fusionner en un tableau de taille $2^m \times 2^m$ ordonné en serpent

– 1. *Shuffle* de chaque ligne du tableau en utilisant les transpositions pair-impair sur chaque ligne (ce qui revient à appliquer la transformation *shuffle* sur les colonnes)

– 2. Trier les paires de colonnes, c'est à dire les tableaux de taille $n \times 2$ en ordre serpent, en utilisant $2n$ étapes de transposition pair-impair sur le réseau linéaire induit sur chaque serpent de longueur $2n$

– 3. Appliquer $2n$ étapes de transposition pair-impair sur le réseau linéaire induit par le serpent de taille n^2.

Questions

1. Faire tourner l'algorithme de tri induit avec $n = 4$ et $a_{i,j} = 21 - 4i - j, \quad 1 \leq i, j \leq 4$
2. Montrer que la première étape de l'algorithme peut s'effectuer en temps $2^{m-1}-1$, l'unité étant un échange entre voisins. En déduire que l'algorithme global de fusion s'effectue en temps $\leq 4.5n$
3. En supposant l'algorithme de fusion correct, construire un algorithme qui trie une séquence de longueur 2^{2m} sur une grille $2^m \times 2^m$. Estimer sa complexité
4. Montrer que le tri par transposition pair-impair sur une grille est correct : il s'agit de montrer que $2n$ étapes de transposition pair-impair dans la troisième phase de l'algorithme de fusion suffisent à obtenir un serpent correctement ordonné.

4 Machine à trier P-RAM de Cole

Enoncé

– Lire le polycopié sur la machine à trier P-RAM de Cole
– Comprendre le principe du fonctionnement en pipeline
– Comprendre comment fusionner des listes en temps constant à l'aide de listes auxiliaires
– Admirer la beauté algorithmique de la construction :-)