Réseaux de tri

Résumé: Ces exercices sont en grande partie tirés du Cormen Leiserson Rivest que vous connaissez tous. On démontre d'abord le principe du 0-1 puis on l'applique sur différents réseaux de tris.
Olivier BEAUMONT et Fabrice RASTELLO sont les auteurs originaux de ce document. Mes apports ont été mineurs (coquilles, typographie, petits dessins,...).

1 Principe du 0-1

On donne ici une technique de preuve différente pour la preuve de correction des réseaux de fusion et de tri. On dit qu'une suite est de type 0-1 si elle n'est constituée que de 0 et de 1.

▷ **Question 1.** Montrer qu'un réseau de comparateurs implante correctement le tri si et seulement si il calcule bien cette fonction pour toutes suites de type 0-1 en entrée.

▷ **Question 2.** Montrer qu'un réseau de comparateurs α trie correctement la séquence $< n, n-1, \dots, 1 >$ ssi il trie correctement les séquences $< 1^i 0^{n-i} >$ pour tout $i \geqslant 1$.

2 Réseau de tri bitonique

Définition 1. On appelle **séquence bitonique** une séquence qui est soit croissante puis décroissante, soit décroissante puis croissante. Ainsi, les séquences $< 2, 3, 7, 7, 4, 1 >$ et $< 12, 5, 10, 11, 19 >$ sont bitoniques. Les séquences binaires bitoniques sont de la forme $0^i 1^j 0^k$ ou de la forme $1^i 0^j 1^k$.

Définition 2. Un **réseau de tri bitonique** est un réseau de comparaison triant toute séquence bitonique.

Définition 3. Un réseau de comparaison triant toute séquence binaire bitonique est un réseau de tri bitonique.

Définition 4. On appelle **séparateur** un réseau de comparaison de profondeur 1 dans lequel chaque ligne i est comparée à la ligne $i + \frac{n}{2}$ pour $i \in \{1, 2, ..., \frac{n}{2}\}$ (on considère que n est pair).

▷ **Question 3.** Imaginez comment construire un réseau de tri bitonique à partir de séparateurs. Quelle est sa profondeur et le nombre de comparateurs utilisés ?

▷ **Question 4.** En utilisant des trieuses bitoniques, construire un réseau fusionnant deux listes triées et déduisez-en la construction d'un réseau général de tri dont on déterminera la profondeur et le nombre de comparateurs.

3 Tri sur une grille 2D

Cette partie utilise le tri par transposition pair-impair sur un réseau linéaire vu en cours. On s'intéresse ici au tri par transposition pair impair sur une grille.

Définition 5. 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 :

$$\begin{aligned} a_{2i,j} &\leqslant a_{2i,j+1}, &\text{si}\quad & 1 \leqslant j \leqslant n-1 \\ a_{2i+1,j+1} &\leqslant a_{2i+1,j}, &\text{si}\quad & 1 \leqslant j \leqslant n-1 \\ a_{2i-1,n} &\leqslant a_{2i,n}, &\text{si}\quad & 1 \leqslant i \leqslant 2^{m-1} \\ a_{2i,1} &\leqslant a_{2i+1,1}, &\text{si}\quad & 1 \leqslant i \leqslant 2^{m-1} - 1. \end{aligned}$$

On peut noter que ce serpent induit un réseau linéaire à l'intérieur de la grille (voir Figure 1).

Définition 6. Un «shuffle» transforme la séquence de $n = 2p$ éléments $< z_1, \dots, z_n >$ en la séquence $< z_1, z_{p+1}, \dots, z_p, z_{2p} >$.

On se propose d'étudier l'algorithme suivant, qui réalise la fusion de 4 tableaux de taille $2^{m-1} \times 2^{m-1}$ ordonnés en serpent en un tableau de taille $2^m \times 2^m$ ordonné en serpent.

$$
\begin{array}{ccccccc}
a_{1,1} & \leftarrow & a_{1,2} & \leftarrow & a_{1,3} & \leftarrow & a_{1,4} \\
\downarrow & & & & & & \\
a_{2,1} & \rightarrow & a_{2,2} & \rightarrow & a_{2,3} & \rightarrow & a_{2,4} \\
& & & & & & \downarrow \\
a_{3,1} & \leftarrow & a_{3,2} & \leftarrow & a_{3,3} & \leftarrow & a_{3,4} \\
\downarrow & & & & & & \\
a_{4,1} & \rightarrow & a_{4,2} & \rightarrow & a_{4,3} & \rightarrow & a_{4,4}
\end{array}
$$

FIG. 1 – L'ordre serpent sur une grille 4×4

1. «shuffle» de chaque ligne du tableau (en utilisant des transpositions pair-impair sur les indices des éléments), 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.

▷ **Question 5.** Faire tourner/l'algorithme de tri induit avec $n = 4$ et $a_{i,j} = 21 - 4i - j$ pour $1 \leqslant i, j \leqslant 4$.

▷ **Question 6.** Montrer que la première étape de l'algorithme peut s'effectuer en temps $2^{m-1} - 1$, l'unité étant un échange entre voisins (on pourra effectuer sur les transpositions pair-impair sur un ensemble d'indices astucieusement choisis). En déduire que l'algorithme global de fusion s'effectue en temps $\leqslant 4.5n$.

▷ **Question 7.** 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.

▷ **Question 8.** 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).