

# Réseaux de tri

## 1 Séquences particulières

On a vu qu'un réseau de comparateurs *primitif* implante correctement le tri si et seulement si il trie correctement la séquence  $\langle n, n-1, \dots, 1 \rangle$ . Pour un réseau de comparateurs général, on a le résultat suivant :

▷ **Question 1** *Montrer qu'un réseau de comparateurs  $\alpha$  trie correctement la séquence  $\langle n, n-1, \dots, 1 \rangle$  si et seulement si il trie correctement les séquences  $\langle 1^i 0^{n-i} \rangle$  pour tout  $i \geq 1$ .*

## 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  $\langle 2, 3, 7, 7, 4, 1 \rangle$  et  $\langle 12, 5, 10, 11, 19 \rangle$  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 comparateurs triant toute séquence binaire bitonique.

**Définition 3.** On appelle **séparateur** un réseau à  $n$  entrées, avec  $n$  pair, composé d'une colonne de comparateurs dans lequel chaque entrée  $i$  est comparée à l'entrée  $i + \frac{n}{2}$  pour  $i \in \{1, 2, \dots, \frac{n}{2}\}$ .

▷ **Question 2** *Comment construire un réseau de tri bitonique à partir de séparateurs ? Quelle est sa profondeur et le nombre de comparateurs utilisés ?*

▷ **Question 3** *En utilisant des trieuses bitoniques, construire un réseau fusionnant deux listes triées. En déduire la construction d'un réseau général de tri dont on déterminera la profondeur et le nombre de comparateurs.*

## 3 Tri sur une grille 2D

Cet exercice étend le tri par transposition pair-impair, déjà étudié sur un réseau linéaire, au cas d'une grille à deux dimensions.

**Définition 4.** 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 :

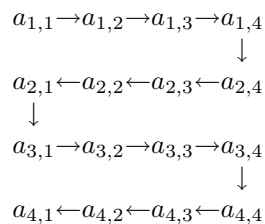
$$\begin{aligned} a_{2i-1,j} &\leq a_{2i-1,j+1}, & \text{si } 1 \leq j \leq n-1, 1 \leq i \leq n/2, \\ a_{2i,j+1} &\leq a_{2i,j}, & \text{si } 1 \leq j \leq n-1, 1 \leq i \leq n/2, \\ a_{2i-1,n} &\leq a_{2i,n}, & \text{si } 1 \leq i \leq n/2, \\ a_{2i,1} &\leq a_{2i+1,1}, & \text{si } 1 \leq i \leq n/2-1. \end{aligned}$$

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

**Définition 5.** Un « shuffle » transforme la séquence de  $n = 2p$  éléments  $\langle z_1, \dots, z_n \rangle$  en la séquence  $\langle z_1, z_{p+1}, z_2, z_{p+2}, \dots, z_p, z_{2p} \rangle$ . Par exemple le « shuffle » de  $(1, 2, 3, 4, 5, 6, 7, 8)$  est  $(1, 5, 2, 6, 3, 7, 4, 8)$ .

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 :

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 transpositions pair-impair sur le réseau linéaire induit sur chaque serpent de longueur  $2n$ .

FIG. 1 – L'ordre serpent sur une grille  $4 \times 4$ .

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

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

▷ **Question 5** *Montrer que la première étape de l'algorithme peut s'effectuer en temps  $2^{m-1} - 1$ , l'unité étant un échange entre voisins (plusieurs échanges entre voisins pouvant être effectués en parallèles pour le même coût). On pourra effectuer les transpositions pair-impair sur un ensemble d'indices astucieusement choisis. En déduire que l'algorithme global de fusion s'effectue en temps  $\leq \frac{9}{2}n$ .*

▷ **Question 6** *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 7** *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 Réponses aux exercices

### ▷ Question 1, page 1

⇒ Supposons tout d'abord que le réseau trie correctement la séquence  $\langle n, n-1, \dots, 1 \rangle$ . Alors (voir la démonstration du lemme 0-1), il trie également correctement les séquences  $\langle f_i(n), f_i(n-1), \dots, f_i(1) \rangle$  pour tout  $i$ , où

$$f_i(x) = 0 \text{ si } x < i \text{ et } 1 \text{ sinon.}$$

⇐ Supposons maintenant que le réseau ne trie pas correctement la séquence  $\langle n, n-1, \dots, 1 \rangle$  et montrons que nécessairement, il ne trie pas correctement une des séquences  $\langle 1^i 0^{n-i} \rangle$ .

Soit  $j$  le plus grand élément de  $\llbracket 1, n \rrbracket$  qui n'est pas à sa place après l'application du réseau de comparateurs  $\alpha$  sur la séquence  $\langle n, n-1, \dots, 1 \rangle$ . Alors, il existe une permutation  $\sigma$  de  $\llbracket 1, j \rrbracket$  telle que :

$$\alpha(\langle n, n-1, \dots, 1 \rangle) = \langle \sigma(1), \dots, \underbrace{\sigma(k)}_{=j}, \dots, \underbrace{\sigma(j)}_{<j}, j+1, \dots, n \rangle.$$

Considérons la fonction  $f_j$  précédemment définie.  $f_j$  étant une fonction monotone croissante, on a

$$\begin{aligned} \alpha(\langle 1^j, 0^{n-j} \rangle) &= \alpha(f_j(\langle n, n-1, \dots, 1 \rangle)) \\ &= f_j(\alpha(\langle n, n-1, \dots, 1 \rangle)) \\ &= \langle 0, \dots, 0, 1, 0, \dots, 0, 1, \dots, 1 \rangle, \end{aligned}$$

et donc le réseau ne trie pas correctement les séquences de types  $\langle 1^i 0^{n-i} \rangle$ .

### ▷ Question 2, page 1

L'effet d'un séparateur sur une séquence bitonique est décrit figure 4. À la sortie du séparateur, on obtient deux séquences bitoniques de même taille dont une au moins est pure (c'est-à-dire composée soit uniquement de 0, soit uniquement de 1).

Pour le montrer, supposons qu'il y ait plus de 1 que de 0 dans la séquence bitonique initiale.

- Supposons la séquence initiale de la forme  $1^i 0^j 1^k$ .
  - Si  $k \geq \frac{n}{2}$ , alors la séquence de sortie est  $1^i 0^j 1^k$ .
  - Si  $i \geq \frac{n}{2}$ , alors la séquence de sortie est  $1^{i-\frac{n}{2}} 0^j 1^{k+\frac{n}{2}}$ .
  - Si  $i < \frac{n}{2}$  et  $k < \frac{n}{2}$  (mais  $i+k \geq \frac{n}{2}$  puisqu'il y a plus de 1), alors la séquence de sortie est de la forme  $0^{\frac{n}{2}-k} 1^{\frac{n}{2}-j} 0^{\frac{n}{2}-i} 1^{\frac{n}{2}}$ .
- Supposons la séquence initiale de la forme  $0^i 1^j 0^k$ .  
Comme  $j \geq \frac{n}{2}$ , la séquence de sortie est de la forme  $0^i 1^{j-\frac{n}{2}} 0^k 1^{\frac{n}{2}}$ .

Dans tous les cas, on vérifie donc que la séquence de sortie a bien la forme souhaitée. Enfin, la démonstration est tout à fait identique s'il y a plus de 0 que de 1.

En utilisant ce résultat, il est facile de construire un réseau de tri bitonique en utilisant des séparateurs, comme indiqué figure 3. Soient  $t_m$  et  $p_m$  respectivement la profondeur et le nombre de comparateurs du réseau de tri bitonique pour une séquence d'entrée bitonique binaire de taille  $n = 2^m$ . On vérifie que :

$$\begin{aligned} t_1 = 1, \quad t_m = t_{m-1} + 1 & \text{ ce qui implique que } t_m = m, \\ p_1 = 1, \quad p_m = t_m 2^{m-1} & \text{ ce qui implique que } p_m = m 2^{m-1}. \end{aligned}$$

Le réseau de tri bitonique ainsi construit a donc pour profondeur  $O(\log n)$  et comporte  $O(n \log n)$  comparateurs.

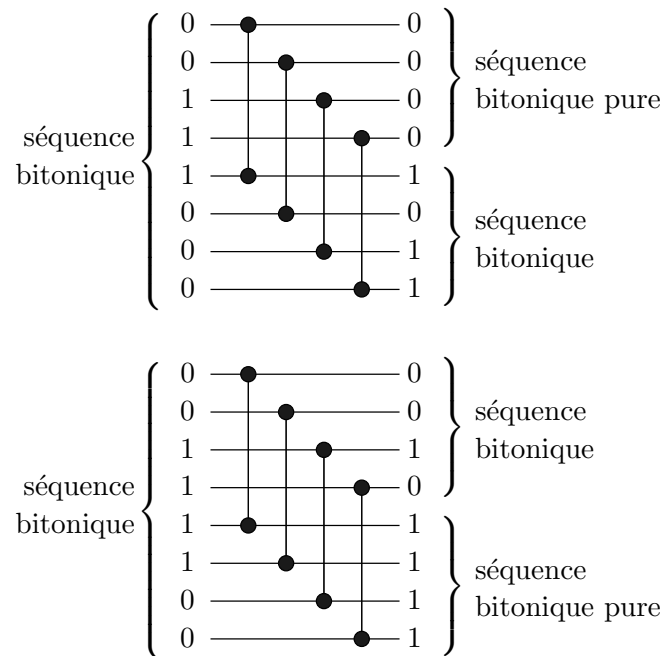


FIG. 2 – Un séparateur de taille 8 appliqué à deux séquences binaires bitoniques différentes.

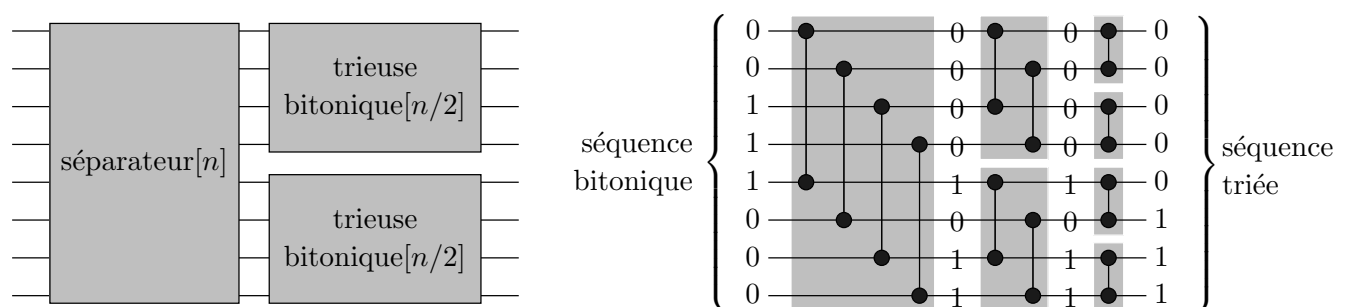


FIG. 3 – Construction d'un réseau d'une trieuse bitonique de séquences de taille  $n$  à partir de séparateurs.

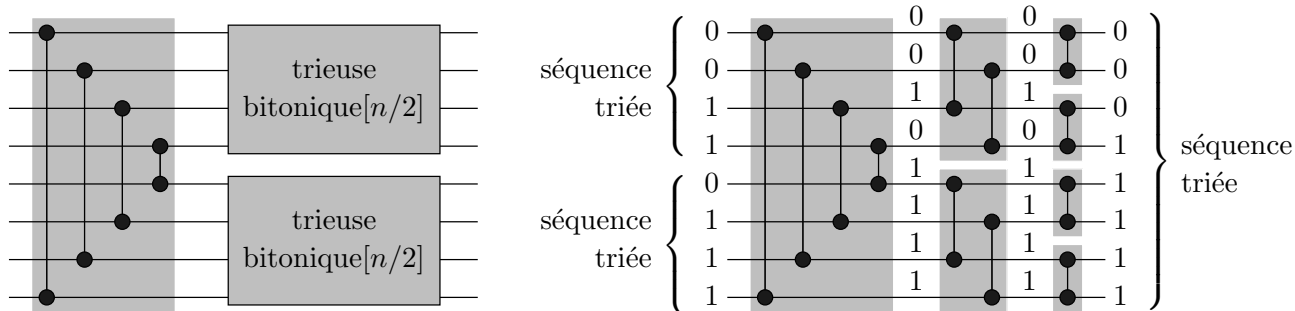


FIG. 4 – Construction d’un réseau de fusion à l’aide de trieuses bitoniques.

▷ **Question 3, page 1**

Pour construire un réseau de fusion à partir de trieuses bitoniques, considérons la fusion de deux listes triées de taille  $n$ ,  $0^i 1^{n-i}$  et  $0^k 1^{n-k}$  respectivement. Si on inverse la deuxième liste triée (ce qui revient à échanger des câbles mais ne nécessite aucun comparateur supplémentaire), on obtient la séquence bitonique  $0^i 1^{2n-i-k} 0^k$  sur laquelle on peut appliquer la trieuse bitonique pour obtenir la fusion des deux listes initiales. Pour inverser la seconde liste, il suffit simplement de modifier le premier étage de la trieuse bitonique comme montré figure 4.

Ensuite, on obtient le réseau général de tri en empilant les réseaux de fusion, comme pour le tri-fusion de Batcher. Le réseau obtenu trie correctement toutes les séquences binaires, et le principe du 0-1 permet de conclure.

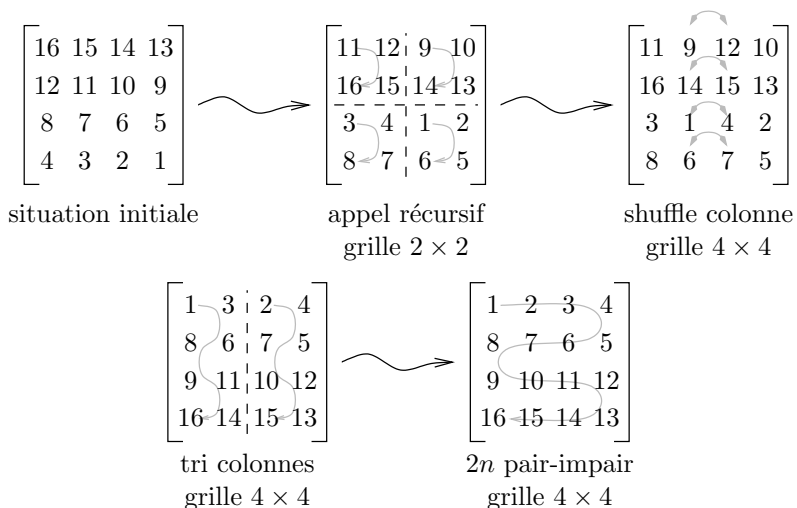
Soient  $t'_m$  et  $p'_m$  respectivement la profondeur et le nombre de comparateurs du réseau de tri ainsi construit pour une séquence d’entrée de taille  $n = 2^m$ . On vérifie que :

$$\begin{aligned} t'_1 &= 1, & t'_m &= t'_{m-1} + t_m & \text{ce qui implique que } & t'_m &= O(m^2), \\ p'_1 &= 1, & p'_m &= 2p'_{m-1} + p_m & \text{ce qui implique que } & p'_m &= O(m^2 2^m). \end{aligned}$$

Le réseau de tri ainsi construit a donc pour profondeur  $O((\log n)^2)$  et comporte  $O(n(\log n)^2)$  comparateurs.

▷ **Question 4, page 1**

Le dessin ci-dessous représente l’évolution de la grille au cours des différentes étapes du tri serpent.





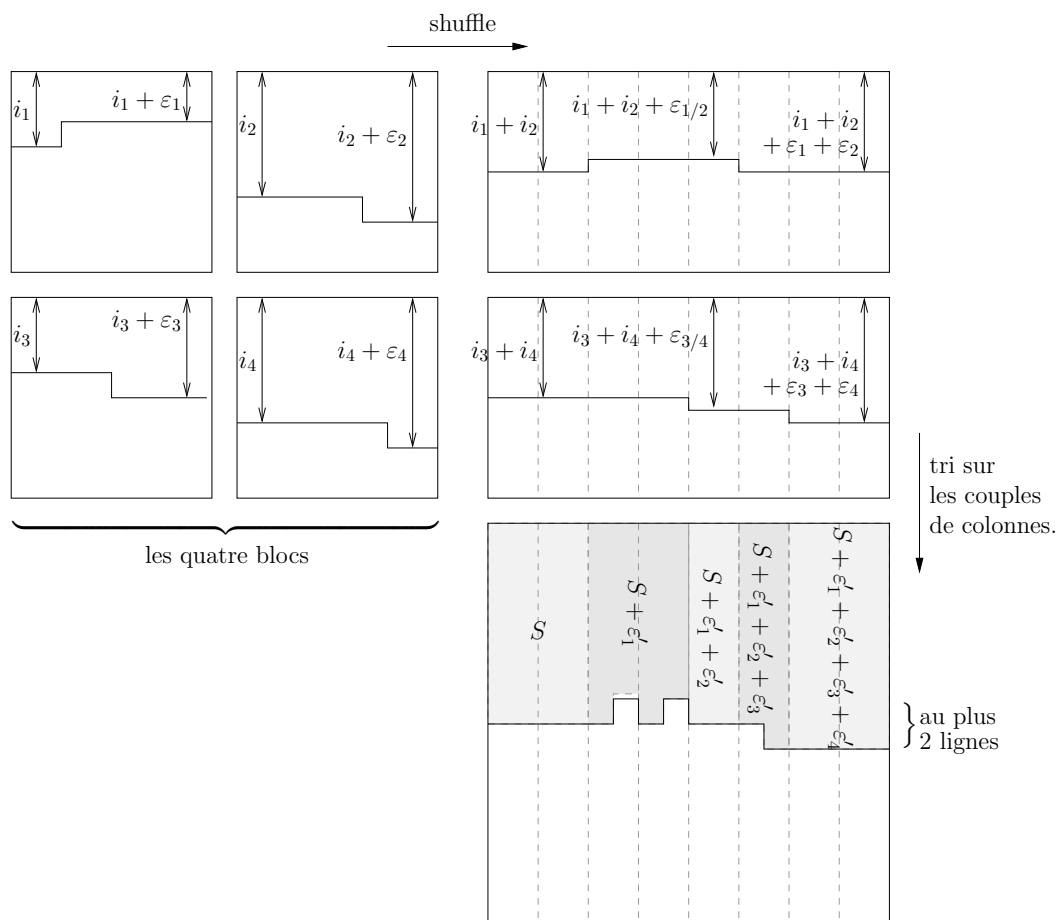


FIG. 6 – Fusion de deux serpents 0-1.

En effet, regardons le nombre de 0 et de 1 dans les couples de colonnes après l'opération de « shuffle ». Dans la moitié supérieure de la grille, il y a au plus trois types de paires de colonnes. Ces paires contiennent chacune  $i_1 + i_2$ ,  $i_1 + i_2 + \varepsilon_{1/2}$  ( $\varepsilon_{1/2}$  valant  $\varepsilon_1$  ou  $\varepsilon_2$  selon les cas) ou  $i_1 + i_2 + \varepsilon_1 + \varepsilon_2$  zéros. On peut faire la même analyse dans la moitié inférieure et en déduire qu'il existe au plus 5 types de paires de colonnes après le tri sur les paires de colonnes. Ces paires de colonnes contiennent donc  $S$ ,  $S + \varepsilon'_1$ ,  $S + \varepsilon'_1 + \varepsilon'_2$ ,  $S + \varepsilon'_1 + \varepsilon'_2 + \varepsilon'_3$  ou  $S + \varepsilon'_1 + \varepsilon'_2 + \varepsilon'_3 + \varepsilon'_4$  zéros (où  $S = i_1 + i_2 + i_3 + i_4$  et les  $\varepsilon'_1, \varepsilon'_2, \varepsilon'_3, \varepsilon'_4$  sont des entiers dans  $\{-1, 1\}$ ).

En remarquant que  $S$  est pair (en tant que somme de quatre nombres impairs) on peut vérifier que, dans tous les cas, au plus deux lignes (de longueur  $n$ ) ne sont pas exclusivement constituées de 0 ou de 1, et donc que le réseau linéaire induit par le serpent de taille  $n^2$  trie la séquence globale en seulement  $2n$  étapes.