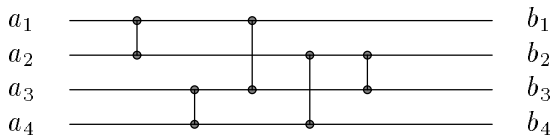
Dea d'Informatique, 1997-98
Algorithme et Combinatoire
Corrigé du problème sur les Réseaux de tri
10 Octobre 1997

Question 1. On peut diviser le réseau en trois étapes la première effectue en parallèle `C(1,2)`



`C(3,4)`, la seconde `C(1,3) C(2,4)` enfin la dernière `C(2,3)` . Après la première étape on a $a_1 < a_2$ et $a_3 < a_4$, ainsi après la deuxième étape le fil 1 contient le plus petit élément et le fil 4 le plus grand. Il reste donc à ordonner les second et troisièmes éléments ce qui est fait au cours de la dernière étape.

Question 2. Le tri bulle s'exprime par l'algorithme

```
for (i = 1; i <= n -1; ++i)
    for (j = 1; j <=  n -i ; ++ j)
         C(j, j+1)
```

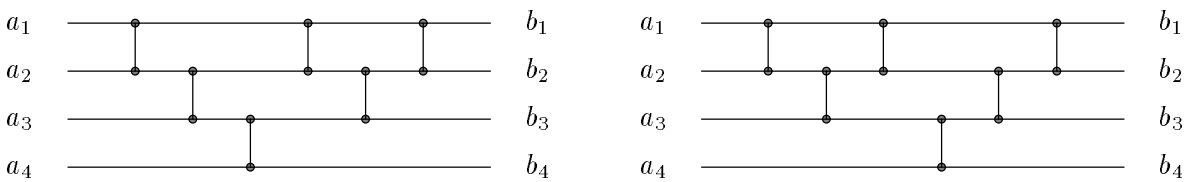


Figure 1: Tri Bulle et Tri par insertion

Le tri par insertion consiste à trier a_1, a_2 puis insérer a_3 et ainsi de suite jusqu'à a_n. Par rapport à ce qui a été vu en cours il faut mettre des comparaisons jusqu'au bout de l'insertion car on n'a pas de test qui permet de savoir si on a trouvé la bonne place. Il s'écrit :

```
for (i = 2; i <= n; ++i)
    for (j = i-1; j >0 ; --j)
         C(j, j+1)
```

Il est représenté sur la figure ci-dessus.

On remarque que les deux tris sont équivalents si on s'autorise à déplacer des comparaisons qui peuvent s'effectuer en parallèle.

Question 3. Dans un réseau primitif chaque opération `C(i,i+1)` diminue d'au plus 1 le nombre d'inversions. Comme le nombre d'inversions peut atteindre $\frac{n(n-1)}{2}$ en entrée et que la sortie ne présente pas d'inversions, le nombre de comparateurs doit être au moins de $\frac{n(n-1)}{2}$.

Question 4. Le texte suggère d'effectuer un raisonnement par récurrence sur le nombre de comparateurs du réseau.

S'il n'y a qu'un seul comparateur `C(i,i+1)` , alors la sortie $c_1, c_2, \ldots c_n$ correspondant à l'entrée $n, n-1, \ldots, 2, 1$ vérifie $c_k = n-k+1$ pour $k \neq i, i+1$ et $c_i = n-i, c_{i+1}$ on a donc pour tous les $p < q$ $c_p > c_q$ sauf si $p = i$ et $q = i+1$. Mais la même relation est vérifiée par b_i, b_{i+1} quelque soit l'entrée puisqu'il y a eu passage dans le comparateur.

Supposons donc la propriété vraie pour k et considérons un réseau R constitué de $k+1$ comparateurs; on peut écrire:

```
R = R'; C(i,i+1)
```

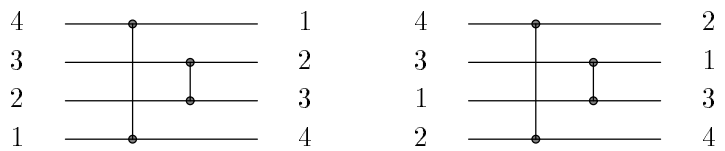
où `R'` ne comporte que k comparateurs. Notons $b'_1, b'_2, \ldots b'_n$ la sortie de R' pour l'entrée $a_1, a_2, \ldots, a_n$; $c_1, c_2, \ldots, c_n$ la sortie de R pour l'entrée $n, n-1, \ldots, 2, 1$ et $b'_1, b'_2, \ldots b'_n$ la sortie de R' pour de cette même entrée. Il faut montrer que si $p < q$ et $b_p > b_q$ alors on a aussi $c_p > c_q$.
Pour cela on commence par remarquer que si ni p ni q ne sont dans l'ensemble $\{i, i+1\}$ alors on a $b_p = b'_p$ et $b_q = b'_q$, l'hypothèse de récurrence donne ainsi $c'_p > c'_q$. Comme le dernier comparateur ne porte pas sur p et q on a aussi $c'_p = c_p$ et $c'_q = c_q$ et le résultat. On remarque aussi que l'on ne peut avoir $p = i$ et $q = i+1$ car ces deux éléments sont dans le bon ordre. Il reste donc deux cas à examiner

- $p \in \{i, i+1\}$ et $q > i+1$
- $q \in \{i, i+1\}$ et $p < i$

1. Supposons $p \in \{i, i+1\}$ et $q > i+1$. Encore deux cas sont à distinguer:

   1.a Le premier est celui où $b'_i > b_q$ et $b'_{i+1} > b_q$. Il y a donc deux inversions et comme $b_q = b'_q$ on peut utiliser l'hypothèse de récurrence et on obtient $c'_i > c'_q$ et $c'_{i+1} > c'_q$. Mais là encore on a $c'_q = c_q$ comme $c_p \in \{c'_i, c'_{i+1}\}$ on obtient $c_p > c_q$. **2.a** Le second est celui où un seul des b'_i, b'_{i+1} est supérieur à b_q dan sce cas on a $p = i+1$ et c_p est le plus grand de c'_i, c'_{i+1}

2. Supposons $q \in \{i, i+1\}$ et $p < i$. Un raisonement symétrique du précédent donne le résultat.



Question 5 La propriété n'est pas vraie pour un réseau non primitif.