
TD8 structures algébriques pour l'informatique

Exercice 1

1. On obtient dans l'ordre :

$$[4,5,1,2,3],[4,1,5,2,3],[4,1,2,5,3],[4,1,2,3,5],[1,4,2,3,5],[1,2,4,3,5],[1,2,3,4,5].$$

2. L'algorithme termine. En effet l'instruction

   Si $t_j > t_{j+1}$ alors échanger t_j et t_{j+1}.

   s'exécute exactement $\sum_{i=1}^{n-1}(n-i) = \sum_{i=1}^{n-1}(i) = n(n-1)/2$ fois.
   Après la $i$ème exécution de la boucle

   Pour i allant de 1 à $n-1$,

   les i derniers éléments de la liste sont à leurs bonnes places (on peut le montrer formellement par récurrence), comme des bulles remontant à la surface. Ainsi après l'exécution de l'algorithme, les $n-1$ derniers éléments de la liste sont à leurs bonnes places, ce qui entraine que le premier élément est aussi à sa bonne place et que la liste est bien triée.

3. A chaque liste $[t_1, \dots t_n]$ manipulée par l'algorithme, on peut associer canoniquement la permutation $\begin{pmatrix} 1 & \dots & n \\ t_1 & \dots & t_n \end{pmatrix}$.

   Échanger les éléments d'indices j et $j+1$ dans la liste revient à multiplier la permutation à droite par la transposition $(j\ j+1)$.

   Par exemple, dans la question 1 pour passer de la première séquence à la deuxième, on a échangé les deuxièmes et troisièmes éléments de la liste, ce qui revient à multiplier la permutation à droite par la transposition $(2\ 3)$. On peut vérifier en effet que :

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 1 & 2 & 3 \end{pmatrix} (2\ 3) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 5 & 2 & 3 \end{pmatrix}$$

   et plus généralement, pour $1 \leq j \leq n-1$:

$$\begin{pmatrix} 1 & \dots j & j+1 & \dots & n \\ t_1 & \dots t_j & t_{j+1} & \dots & i_n \end{pmatrix} (j\ j+1) = \begin{pmatrix} 1 & \dots j & j+1 & \dots & n \\ i_1 & \dots t_{j+1} & t_j & \dots & i_n \end{pmatrix}$$

   Par ailleurs, on sait que l'algorithme finit toujours et quelque soit la liste (ou encore la permutation) d'entrée, on obtient à la fin, la liste triée qui correspond à la permutation identité. Comme on sait de plus que le nombre d'itérations de l'algorithme est $n(n-1)/2$, on en déduit qu'il existe $k \leq n(n-1)/2$, et $\tau_1, \dots, \tau_k \in \{(j\ j+1) : 1 \leq j \leq n-1\}$ tels que :

$$\begin{pmatrix} 1 & \dots & n \\ t_1 & \dots & t_n \end{pmatrix} \circ \tau_1 \circ \tau_2 \circ \dots \circ \tau_{k-1} \circ \tau_k = Id$$

   En multipliant à gauche par les transpositions inverses (qui sont les mêmes car une transposition est son propre élément symétrique), on obtient:

$$\begin{pmatrix} 1 & \dots & n \\ t_1 & \dots & t_n \end{pmatrix} = \tau_k \circ \tau_{k-1} \circ \dots \circ \tau_2 \circ \tau_1.$$

Ainsi,
$$\begin{pmatrix}1&2&3&4&5\\4&5&1&2&3\end{pmatrix}(2\ 3)(3\ 4)(4\ 5)(1\ 2)(2\ 3)(3\ 4)=\begin{pmatrix}1&2&3&4&5\\1&2&3&4&5\end{pmatrix}=Id.$$
Et en multipliant à gauche par les transpositions inverses, on obtient:
$$\begin{pmatrix}1&2&3&4&5\\4&5&1&2&3\end{pmatrix}=(3\ 4)(2\ 3)(1\ 2)(4\ 5)(3\ 4)(2\ 3).$$

4. Comme tout élément de S_n s'écrit comme produit des transpositions $((j,j+1))_{1\leq j<n}$, on en déduit que le sous–groupe de S_n engendré par les transpositions est S_n tout entier:
$$\langle\ ((j,j+1))_{1\leq j<n}\ \rangle=S_n.$$

5. Il n' y avait pas de question.

6. Les mêmes arguments que dans le cas du tri bulle s'applique ici. L'algorithme termine toujours. Comme dans le cas du tri bulle, ici l'instruction
$$\text{Si } t_j>t_{indice} \text{ alors } indice=j.$$
s'exécute exactement $\sum_{i=1}^{n-1}(n-i)=\sum_{i=1}^{n-1}(i)=n(n-1)/2$ fois. Ici encore, après la $i$ème exécution de la boucle
$$\text{Pour } i \text{ allant de 1 à } n-1,$$
les i derniers éléments de la liste sont à leurs bonnes places (on peut le montrer formellement par récurrence). Ainsi après l'exécution de l'algorithme, les $n-1$ derniers éléments de la liste sont à leurs bonnes places, ce qui entraine que le premier élément est aussi à sa bonne place et que la liste est bien triée.

7. On obtient ici:
$$\begin{pmatrix}1&2&3&4&5\\4&5&1&2&3\end{pmatrix}(2\ 5)(1\ 4)(2\ 3)(1\ 2)=Id$$
et finalement:
$$\begin{pmatrix}1&2&3&4&5\\4&5&1&2&3\end{pmatrix}=(1\ 2)(2\ 3)(1\ 4)(2\ 5).$$

8. On a par exemple,
$$\begin{pmatrix}1&2&3&4&5\\4&5&1&2&3\end{pmatrix}=(1\ 2)(2\ 3)(1\ 4)(2\ 5)=(3\ 4)(2\ 3)(1\ 2)(4\ 5)(3\ 4)(2\ 3),$$
ce qui montre que la décomposition d'une permutation en produit de transpositions n'est pas unique.

Exercice 2 .

1. vu en cours. On peut également le voir par la décomposition fournie par une variation du tri sélection, où au lieu de rechercher les maximums et les mettre en fin de liste, on recherche les minimums et on les met en début de liste.

2. vu ci-dessus (décomposition fournie par le tri bulle)

3. D'après 2, tout élément de S_n de décompose en produit de transpositions de type $(i\ i+1)$. On remarque de plus que chaque transposition $(i\ i+1)$ se décompose en produit des deux cycles $(1\ 2)$ et $(1\ 2\ \dots\ n)$:
$$(i\ i+1)=(1\ 2\ \dots\ n)^{i-1}(1\ 2)(1\ 2\ \dots\ n)^{n-i+1}$$
On en déduit que tout élément de S_n se décompose bien en un produit où les facteurs sont un des deux cycles $(1\ 2)$ ou $(1\ 2\ \dots\ n)$.