

[PROF] Tris

Fiche 4

Ce document a pour objectif de guider les enseignants pour le cours d'algorithmique. Il est déconseillé de le fournir aux étudiants, car il vise surtout à vous permettre de guider la séance. Vous n'êtes évidemment pas obligé de le suivre à la lettre (c'est même déconseillé, car cela va autant vous perturber vous que la classe : suivez votre chemin de pensée et/ou celui de la classe, et ensuite vérifiez que vous n'avez rien oublié).

La quatrième partie présente les tris. Vous pouvez introduire cette séance par la vidéo suivante : YouTube - 15 Sorting Algorithms in 6 Minutes (*attention au son et au risque d'épilepsie*).

1 Définitions

Rappelez les informations classiques suivantes :

Définition informelle d'un algorithme de tri¹ : « *Un algorithme de tri est, en informatique ou en mathématiques, un algorithme qui permet d'organiser une collection d'objets selon une relation d'ordre déterminée.* ».

En pratique cela implique d'organiser des objets comparables les uns avec les autres selon un critère. On peut ordonner des fruits selon leur taille, leur poids, etc. Le critère pour ordonner doit absolument disposer d'une *relation d'ordre* du type « plus grand que » ou « inférieur ou égal à », afin de trier les objets (c'est-à-dire leur donner un numéro d'ordre).

Très concrètement, les algorithmes de tri ont comme objectif de réorganiser en ordre croissant ou décroissant un tableau composé d'éléments comparables deux à deux. Les principaux problèmes rencontrés concernent :

- la *complexité temporelle*, c'est-à-dire le temps nécessaire pour trier (souvent associé au nombre de comparaisons et déplacements nécessaires),
- la *complexité spatiale*, c'est-à-dire l'espace nécessaire pour réorganiser le tableau (déplacement d'au plus 2 éléments à la fois, ou construction complète d'un nouveau tableau en mémoire en parallèle de l'ancien),
- la *stabilité*, c'est-à-dire que l'on conserve l'ordre des éléments considérés comme égaux dans le tableau initial (si deux pommes P1 suivie de P2 ont le même poids, on doit conserver l'ordre en sortie P1 suivie de P2 et surtout pas P2 suivie de P1 : en effet, un tri sur un autre critère que le poids a peut être été effectué préalablement, et ainsi, les pommes seront triées principalement sur leur poids, et ensuite sur chaque critère précédent).

Rappelez que l'on travaille sur des tableaux.

Définissez le *tri en place* : on modifie le tableau donné en paramètre sans jamais créer de tableau supplémentaire ou intermédiaire.

Explicitez pourquoi les tris ne sont en fait que des procédures qui ne renvoient rien : les tris sont *en place*.

1. Wikipedia (octobre 2023) : Algorithme de tri

2 Swap

- Expliquez que puisque les tris se font en place, alors il faut une procédure ou fonction qui permette d'échanger deux valeurs entre elles dans un tableau
- Demandez aux étudiants de rédiger une telle fonction (elle renvoie faux si les index à échanger sont hors du tableau)

3 Tri à Bulles

- Donnez la logique générale oralement :
« *Le tri à bulles vise à faire remonter tour à tour les plus grandes valeurs vers la fin du tableau.* »
« *Si deux valeurs côte à côte sont dans le désordre, on les inverse, et on teste les suivantes jusqu'à la fin du tableau. On recommence ce traitement sur chaque case du tableau autant de fois que nécessaire (c'est-à-dire autant de fois qu'il y a de cases).* »
- Faire un exemple en démarrant avec le tableau suivant, et en faisant 3-4 tours de tris :

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 18 | 22 | 15 | 32 | 42 | 10 | 24 | 29 |
|----|----|----|----|----|----|----|----|

- Faire remarquer aux étudiants que la dernière case se retrouve fixée à chaque tour, car la plus grande valeur y a été placée
- Demandez aux étudiants de réfléchir à comment faire l'algorithme
- Au bout de quelques minutes, indiquez qu'il *faut* utiliser 2 boucles imbriquées, et explicitez-les :
 - Une boucle pour comparer toutes les cases 2 à 2
 - Une boucle pour répéter cette opération autant de fois qu'il y a de cases (*mais aussi pour limiter le nombre de cases testées à chaque fois*)
- Implémentez le tri / Faire la correction

4 Tri par Sélection

- Donnez la logique générale oralement :
« *Le tri par sélection est le tri le plus évident humainement parlant : il suffit de chercher à chaque tour le plus grand élément restant, pour le placer en fin de tableau.* »
« *On cherche donc la valeur la plus grande en parcourant toutes les cases, puis, on échange la place de cet élément avec le tout dernier du tableau (pour le mettre à sa place définitive). Et on recommence ce traitement (sans lire la dernière case à chaque fois) jusqu'à avoir ordonné tous les éléments.* »
- Faire un exemple en démarrant avec le tableau précédent, en faisant 3-4 tours de tris
- Faire remarquer aux étudiants que tous les algorithmes de tri ont au moins 2 boucles
- Demandez aux étudiants de réfléchir à comment faire l'algorithme

- Implémentez le tri / Faire la correction
- Demandez aux étudiants le cas d'un tableau à 1 élément, 2 éléments... 0 élément
- Faire remarquer que ce tri ne peut pas bien gérer le cas à 2 éléments, et qu'il faut donc faire une fonction chapeau :
 - La fonction chapeau doit vérifier si le tableau a suffisamment de cases pour faire un tri (donc cas 1 et 0 sont laissés intacts)
 - La fonction chapeau doit regarder si, dans le cas à 2 cases, les cases sont dans le désordre, et si oui, il faut les inverser
 - Dans tous les autres cas, on doit appeler la fonction de tri précédemment écrite

5 Tri par Insertion

- Donnez la logique générale oralement :
 - « *Le tri par insertion considère que le tableau donné en paramètre n'est pas trié, et seuls les éléments qu'il a manipulés successivement le sont. Ainsi, chaque élément du tableau est comparé à tous ceux déjà triés, et on le place là où il devrait être.* »
 - « *Pour le premier élément, celui-ci est considéré comme déjà trié, on peut donc passer au deuxième. Le deuxième est comparé avec l'unique élément déjà trié : on échange leurs deux places si nécessaire. Le troisième élément est comparé au plus grand des deux éléments triés, s'il est plus grand ou égal, il reste à sa place, sinon on le décale vers la gauche d'un cran, et on le compare à l'élément suivant (et ainsi de suite pour trouver sa place finale).* »
- Faire un exemple en démarrant avec le tableau précédent, en faisant 3-4 tours de tris
- Faire remarquer aux étudiants que tous les algorithmes de tri ont au moins 2 boucles
- Demandez aux étudiants de réfléchir à comment faire l'algorithme
- Implémentez le tri / Faire la correction
- Demandez aux étudiants si l'algorithme arrive à traiter le cas à 2 éléments

6 Les fonctions de tri dans nos outils

Le début de cette partie est accessible à tous, mais la suite est optionnelle (elle peut être rapidement abordée avec les étudiants ayant déjà vu les tris).

- Demandez comment faire pour trier dans l'ordre *descendant*
- Demandez comment faire pour trier autre chose que des nombres (ou des caractères) : comment trier des structures de données ?
- *[Optionnel]* Présentez l'idée d'une fonction renvoyant vrai/faux si *elt1* est plus grand que *elt2*
- *[Optionnel]* Présentez comment rendre la fonction de tri indépendante de la fonction de comparaison
- *[Optionnel]* Montrez le manuel de *qsort(3)*, et mentionnez les *pointeurs sur fonction*

*Ce document et ses illustrations ont été réalisés par Fabrice BOISSIER en décembre 2024
(dernière mise à jour décembre 2024)*