

TD4 : Le tri fusion

Compétences

- Décrire le fonctionnement d'un algorithme à partir de son implémentation.
- Évaluer de manière empirique la complexité d'un algorithme récursif.
- Dessiner l'arbre d'appels du tri fusion à partir d'un exemple.

Un peu d'histoire

Durant la seconde Guerre mondiale, La *Moore School of Engineering* à Philadelphie embauche des femmes pour effectuer des calculs de trajectoires. L'une d'elle Betty Holberton est vite remarquée pour ses aptitudes mathématiques et fut choisie pour être l'une des six programmatrices de l'ENIAC (acronyme de l'expression anglaise Electronic Numerical Integrator And Computer). Elle développe en 1952 le premier algorithme de tri connu.

In 1952, Holberton developed the Sort-Merge Generator" for the UNIVAC I which produced a program to sort and merge files. This was the first step toward actually using a computer to write programs (i.e., a precursor to the concept of a compiler) and was called "the first major *software routine* ever developed for automatic programming

Exercices : tri fusion

Appliquer le tri fusion à un tableau t contenant n éléments consiste à trier récursivement le tableau entre les indices 0 et $\lfloor \frac{n-1}{2} \rfloor$ et entre les indices $\lfloor \frac{n-1}{2} \rfloor + 1$ et $n-1$, puis à fusionner les deux sous-tableaux triés. L'idée est qu'il est facile de fusionner deux tableaux triés en un tableau lui-même trié.

* **Ex. 1** — Appliquer le principe du tri fusion au tableau suivant $t = [7, 3, 5, 4, 2, 1]$

* **Ex. 2** — Soient t_1 et t_2 deux tableaux de nombres entiers triés dans l'ordre croissant. On veut fusionner ces deux tableaux en un troisième tableau t_3 également trié dans l'ordre croissant. Par exemple, si $t_1 = [2, 5, 5, 8, 9]$ et $t_2 = [1, 3, 5]$, le tableau résultant t_3 contiendra $[1, 2, 3, 5, 5, 5, 8, 9]$.

Écrire la fonction `fusionnerV1(t1, n1, t2, n2, t3)` qui fusionne les n_1 éléments (triés) de t_1 et les n_2 éléments (triés) de t_2 dans le tableau t_3 . On suppose que la taille de t_3 est suffisante. La fonction retournera le nombre d'éléments du tableau t_3 .

** **Ex. 3** — Si on applique le triFusion à un tableau de n éléments, où se trouve le premier couple d'éléments qui seront comparés entre eux? Où se trouve le deuxième couple d'éléments qui seront comparés entre eux? Qu'est-ce qu'on peut dire sur les indices respectifs du dernier couple d'éléments qui seront comparés entre eux?

*** **Ex. 4** — Donner le nombre de comparaisons nécessaires entre éléments du tableau pour effectuer le tri fusion d'un tableau de taille n dans le pire des cas (quel est-il?), puis dans le meilleur des cas (quel est-il?).

*** **Ex. 5** — Donner le nombre d'affectations à une case du tableau nécessaires pour effectuer le tri fusion d'un tableau de taille n dans le pire des cas (quel est-il?), puis dans le meilleur des cas (quel est-il?).

* **Ex. 6** — En fonction du nombre n d'éléments à trier, quel est l'ordre de grandeur de la complexité en temps du tri fusion?

* **Ex. 7** — Si le tableau contient deux occurrences du même élément, est-ce qu'après le tri l'ordre initial des deux occurrences entre elles est encore respecté?

* **Ex. 8** — Dessiner l'arbre d'appel de la fonction `triFusion` pour le tableau $t = [10, 4, 7, 3, 9, 11, 5]$

*** **Ex. 9** — En s'inspirant de la fonction `fusionnerV1`, écrire une fonction `fusionnerV2(t, debut, milieu, fin)` qui, étant donné un tableau t trié entre les indices `debut` et `milieu` et entre les indices `milieu+1` et `fin`, trie t entre les indices `debut` et `fin`.

Par exemple, soit $t = [4, 1, 5, 0, 3, 8, 2]$. On remarque que $[1, 5]$ est trié donc t est trié entre les indices 1 et 2. Pareillement, $[0, 3, 8]$ est trié donc t est trié entre les indices 3 et 5. L'appel de la fonction : `fusionnerV2(t, 1, 2, 5)` va ainsi fusionner ces deux sous-tableaux et on aura $t = [4, 0, 1, 3, 5, 8, 2]$ qui est trié entre les indices 1 et 5.

Exercices : tri rapide

L'idée du tri rapide est de choisir dans le tableau à trier un élément qu'on appelle le pivot (on peut par exemple prendre le premier élément du tableau) et de mettre à sa gauche tous les éléments plus petits que lui et à sa droite tous

les éléments plus grands ou égaux que lui. Pour trier le tableau initial, il ne reste alors plus qu'à trier les sous-tableaux à gauche et à droite du pivot, ce qu'on fait en itérant récursivement.

* **Ex. 10** — soit $t = [7, 3, 5, 1, 8, 4]$ le tableau à trier. Décrire les différentes étapes nécessaires pour trier ce tableau sur le principe du tri rapide.

*** **Ex. 11** — Écrire une fonction `réorganiser(t, debut, fin)` qui utilise comme pivot l'élément d'indice `debut` et réorganise le sous-tableau situé entre les indices `debut` et `fin` de manière que l'on obtienne :

éléments < pivot	pivot	éléments >= pivot
------------------	-------	-------------------

Cette fonction retourne l'indice de la nouvelle position du pivot.

*** **Ex. 12** — Écrire une fonction récursive `triRapideRec(t, debut, fin)` qui trie récursivement le tableau `t` entre les indices `debut` et `fin`.

* **Ex. 13** — Écrire une fonction `triRapide(t, n)` qui trie le tableau `t` de taille `n` en utilisant `triRapideRec`.

** **Ex. 14** — Donner le nombre de comparaisons nécessaires entre éléments du tableau pour placer définitivement un élément choisi comme pivot.

** **Ex. 15** — Donner le nombre de comparaisons nécessaires entre éléments du tableau pour effectuer le tri rapide d'un tableau de taille `n` dans le pire des cas (quel est-il?), puis dans le meilleur des cas (quel est-il?).

** **Ex. 16** — En fonction du nombre `n` d'éléments à trier, quel est l'ordre de grandeur de la complexité en temps du tri rapide?