Travaux Dirigés de Programmation Impérative n°6

Cours d'informatique de Première Année

—Licence MIEF/SPI—

Tri Fusion

Dans ce TD on aborde un nouvel algorithme de tri, plus efficace que ceux vu en cours. C'est l'occasion de réviser le cours sur les tris et la complexité.

▸ Exercice 1. Fusionner deux tableaux triés

- *Écrire un* **algorithme** *qui prends en entrée deux tableaux triés t_1 et t_2 et leur taille n_1 et n_2 et renvoie un tableau trié de longueur $n_1 + n_2$ contenant les valeurs des tableaux passés en entrée.*
- *Reprendre l'idée de l'algorithme précédent pour écrire une fonction* `fusion` *qui prend en entrée un tableau t de longueur n et un paramètre l tel que :*
  - *les cases d'indice $[0, \ldots, l-1]$ sont triées entre elles,*
  - *les cases d'indice $[l, \ldots, n]$ sont triées entre elles.*

  La fonction renvoit un tableau trié de taille n.
- *Testez votre algorithme sur le tableau suivant ($n = 7$ et $l = 4$)*

1	3	7	9	1	2	8

- *Quelle est la complexité en temps et en espace de votre fonction ?*

▸ Exercice 2. Tri fusion

On remarque qu'un tableau de taille 1 est toujours trié.

Le principe du tri fusion est le suivant :

- *si le tableau n'est pas de taille 1, on sépare le tableau en deux parties égale et on trie ces deux parties.*
- *Une fois que les deux parties sont triées, on les fusionne.*

On va utiliser cette description pour la question suivante.

- *Écrire une fonction* **récursive** `tri_fusion` *qui prend en entrée un tableau t de longueur n et le tri en utilisant la fonction* `fusion`.
- *Testez votre algorithme sur le tableau suivant*

7	1	9	3	1	8	2

- *Quelle est la complexité en temps et en espace de votre fonction ?*