

Compléments de Programmation

Licence 1 UPEC 2023/2024

TM6 : Tri

Nous avons vu en cours le tri à Bulles. Vous allez implémenter, avec des fonctions statiques Java, les variants suivants du tri à Bulles :

Question 1 (Tri à Bulles - variant "vérification"):

On fait des passes et on s'arrête dès qu'on ne fait plus d'échanges (comme nous avons vu en cours).

Question 2 (Tri à Bulles - variant "paresseux"):

On a la flemme de vérifier si on fait des échanges ou pas : du coup on fait exactement n passes, où n est la taille du tableau à trier. Cette version est correcte, car, comme nous avons vu en cours, à chaque passe, au moins un élément trouve sa position finale.

Question 3 (Tri à Bulles - variant "optimisé"):

Dans ce variant, on fait aussi exactement n passes, mais à chaque passe on parcourt une partie plus petite du tableau à trier. Pourquoi ce variant est correct ?

Question 4 (Tri à Bulles - variant "roumain"):

Implémentez le variant vu dans la vidéo des danseurs roumains. En quoi ce variant est différent du variant précédent ?

Question 5 (Tri par sélection):

Implémentez en Java le tri par sélection. Dans ce tri, on fait n passes. A la passe i , on cherche l'élément le plus petit du tableau entre $i - 1$ et $tab.length - 1$, et on l'échange avec l'élément à la position $i - 1$. Attention aux détails.

Regardez la vidéo des danseurs roumains pour ce tri (Selection Sort). Est-ce leur version correspondre à la votre ?

Question 6 (Tri par insertion):

Quand on joue à cartes, on prend en main les cartes une par une. Chaque carte est insérée à sa place dans la main jusqu'à ce que toutes les cartes son en main, et la main est triée. Est-ce que vous réussissez à imaginer une façon d'implémenter cet algorithme de tri en Java ?

Cherchez en ligne la vidéo des danseurs roumains pour ce tri (Insertion Sort). Est-ce leur version correspondre à la votre ?