

Algorithmique et Programmation 3

Semestre 3

TP4 : Algorithmes de tri

Algorithmes de tri

1. Implémentez les algorithmes de tri simples vus en cours : tri par sélection, tri par insertion et tri à bulles.
 - (a) Générez aléatoirement plusieurs tableaux de tailles différentes et comparez le temps d'exécution des trois algorithmes. Que remarquez-vous ?
2. Le **tri Shaker** est une variante du tri à bulles qui alterne le sens du parcours à chaque passe. Écrivez cet algorithme.
3. Écrivez un algorithme de tri de piles utilisant le principe du tri par sélection. La pile est la seule structure de données autorisée (ni files ni listes).
4. Générez aléatoirement des listes et comparez les tris suivants : tri fusion et tri par insertion.
5. Le tri par insertion est plus rapide que le tri fusion pour de petites listes. Modifiez le tri fusion pour qu'il utilise le tri par insertion en dessous d'une certaine taille (par exemple, moins de 10 éléments).
6. Le **tri mixte** consiste à diviser une liste en k sous-listes, à trier chacune par insertion, puis à fusionner les sous-listes triées. Implémentez ce tri et déterminez la meilleure valeur de k pour optimiser la complexité.
7. Implémentez le tri rapide, puis générez des listes aléatoires et comparez ses performances au tri fusion.
8. Modifiez le tri rapide pour compter le nombre de comparaisons effectuées. Faites la même chose avec le tri fusion et comparez les deux algorithmes.
9. Modifiez le tri rapide en choisissant comme pivot le dernier élément de la liste.
10. Implémentez une fonction `trie_chaine` de type `string → string` qui retourne la chaîne obtenue en triant les caractères de la chaîne passée en paramètre.

Tri et classes

1. Écrivez un programme qui trie en ordre croissant une liste de points. Un **Point** est défini par son abscisse et son ordonnée. Un point A est considéré inférieur à un point B si sa distance à l'origine est plus petite. Implémentez au moins deux algorithmes de tri parmi ceux vus en cours.