

A Introduction à l'algorithmique

1. Exprimer la fonction $n^4 - 128 * n^3 + \log_2(n)$ en notation de Landau
2. Des deux algorithmes suivants lequel a la plus grande complexité ?

Algorithme A a une complexité en $O(\log_2(n^3))$

Algorithme B a une complexité en $O(\log_2(n))$

3. On cherche à traiter des informations sur les communes de France. Deux algorithmes sont proposés :

Algorithme A a une complexité en $O(n.\log_2(n))$

Algorithme B a une complexité en $O(n^2)$

Il s'avère que B est plus rapide que A. Comment est-ce possible ? Serait-ce le cas sur un ensemble plus large de communes ?

4. Proposer un algorithme dit "du champion" qui trouve la plus petite valeur d'un tableau. Calculer sa complexité.
5. Il est bien connu que pour échanger les valeurs de deux variables, il faut une troisième variable intermédiaire. Cependant, pour les données numériques entières, il existe un moyen de ne pas recourir à une variable supplémentaire. Comment faire ? Est-ce que ce "truc" diminue le coût de l'opération ?
6. Écrire un algorithme qui, étant donné un tableau d'entiers *tab*, recherche la position de la première case contenant 0 (soit *k* l'indice de cette case), et qui affiche le contenu des cases de *k* à 1 à rebours. Calculer le coût en temps de cet algorithme.
7. Proposer un algorithme qui décide si un nombre entier est premier (i.e n'a pas d'autre diviseur que 1 et lui-même). Comment caractériser le meilleur cas, le pire cas ? Quelle est la complexité de votre algorithme ? Essayer plusieurs solutions.
8. Comment peut-on modifier la plupart des algorithmes pour avoir un bon temps de calcul dans le meilleur cas ?
9. On cherche à évacuer les lignes en double dans un tableau de chiffres. Proposer un algorithme pour rechercher les lignes en doubles et les supprimer. Quelle est la complexité de cet algorithme dans le meilleur des cas ? Dans le pire des cas ?