

## Introduction à l'algorithmique. Exercices

1. Écrire un algorithme qui renvoie l'entier le plus grand d'un tableau d'entiers. Exemple :  
`t(4, 77, 52, 23)`  
`max` vaut 77
2. Proposer un algorithme "du champion" qui trouve la plus petite valeur d'un tableau. Calculer sa complexité.
3. Écrire un algorithme qui échange le contenu de deux variables. Exemple :  
Avant, `a` vaut 4, `b` vaut 9  
Après, `a` vaut 9, `b` vaut 4
4. 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 ?
5. Exprimer la fonction suivante en termes de la notation  $O()$   
$$n^3/1000 - 100n^2 - 100n + 3$$
6. Comment peut-on modifier la plupart des algorithmes pour avoir un bon temps de calcul dans le meilleur cas ?
7. Ecrire 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.
8. 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 ?
9. Une première méthode de tri.
  - (a) Proposer un algorithme de tri qui utilise les algorithmes `échange` et `donneMax` vus plus haut.
  - (b) Faire « tourner » cet algorithme de tri avec les jeux de données suivants : (3, 65, 12, 6, 23, 77, 7, 1) et (1, 3, 6, 7, 12, 23, 65, 77). Compter le nombre d'étapes nécessaire dans chaque cas. Comment pourrait-on « optimiser » cet algorithme ?