Série N°02 – Algorithmes de tri

Exercice 01 : Tri un peu spécial (compétition – solve it 2023)

Ecrire une fonction ***« valeurMax »*** qui prend comme paramètre un tableau ***« T »*** d'entiers positifs (ou nuls), et qui réorganise ses cases tel que le tableau forme la plus grande valeur possible. Vu que la valeur maximale peut être très grande, retournez un tableau de caractères (string) plutôt qu'un entier.

Exemples d'entrées et de sorties :

```
Entrée : T = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0}
Sortie : "9876543210"

Entrée : T = {54, 14, 89, 2, 301, 45, 7}
Sortie : "8975432143012"
```

La fonction doit avoir la signature suivante :

```
char* valeurMax(int T[], int n);
```

Indication : utiliser n'importe quel algorithme de tri (idéalement « quick-sort » ou « merge-sort »), mais il faut bien choisir le comparateur

Exercice 02 : La valeur minimale possible

Ecrire une fonction ***« minMemeTaille »*** qui prend comme paramètre un nombre entier, et qui réorganise ses chiffres tel que :

- La valeur du résultat est la plus petite possible en conservant les mêmes chiffres.
- Il ne doit pas y avoir de zéro en début de nombre (sauf si le nombre est égal à 0).

Indications

- Penser à utiliser une version légèrement modifiée d'un algorithme de tri (supposez que vous avez déjà un algorithme de tri implémenté).
- Attention aux nombres négatifs (dans ce cas, vous devez maximiser la valeur absolue).

Exercice 03 : La valeur manquante

Soit ***T*** un tableau d'entier de taille ***n***. On suppose que ***T*** contient tous les entiers de ***0*** à ***n*** sauf un. L'objectif de cet exercice est d'écrire une fonction qui renvoie la valeur manquante.

- La solution naïve est de parcourir le tableau et de tester pour chaque entier de ***0*** à ***n*** s'il est présent dans le tableau. Donner l'implémentation de cette solution. Quelle est sa complexité ?
- Comment peut-on réduire la complexité de cet algorithme ? Implémenter cette solution.

Exercice 04 : Chercher deux nombres avec une somme donnée

L'objectif de cet exercice est d'écrire une fonction ***« somme2 »*** qui prend comme argument un tableau ***T*** et un entier S et qui retourne vrai si le tableau T contient deux nombres dont la somme est égale à S. Par exemple, si T = [1, 2, 3, 4, 5] et S = 7, la fonction doit retourner vrai car 3 + 4 = 7.

- Commencer par implémenter une version naïve de cette fonction qui teste toutes les paires possibles de nombres dans le tableau. Quelle est la complexité de cette fonction ?
- Comment peut-on utiliser les algorithmes de tri pour améliorer la complexité de cette fonction en complexité sous-quadratique ? Implémenter cette version.