

PROG

TD#1: Tableaux, récursion

2020-W37

Exercice 1 : Parcours élémentaires

Q.1 : Écrivez une fonction qui prend en entrée un tableau d'entiers signés de 64 bits de taille arbitraire¹ et renvoie la valeur minimum stockée dans le tableau.

Q.2 : Écrivez une fonction qui prend en entrée un tableau d'entiers non signés de 32 bits de taille arbitraire et renvoie une approximation flottante en simple précision de sa moyenne.

Exercice 2 : Application simple

Q.1 : Écrivez une fonction qui prend en entrée un tableau d'entiers signés de 8 bits de taille arbitraire et le modifie pour que chaque entrée soit égale à la valeur absolue de l'entrée initiale.

Exercice 3 : Primalité

Q.1 : Écrivez une fonction qui prend en entrée un entier non signé de 64 bits et renvoie 1 (resp. 0) s'il est premier (resp. composé), en testant sa divisibilité par les entiers plus petits que sa racine carrée.

Q.2 : Quelle est la complexité de cet algorithme ? Pensez-vous qu'il soit possible de faire mieux ?

Exercice 4 : Parcours avec prétraitement

On souhaite écrire une fonction qui prend en entrée un multiensemble d'entiers positifs représenté par un tableau d'entiers signés de 16 bits de taille arbitraire et «supprime» tous les éléments de multiplicité supérieure à 1. On considérera pour cela que toutes les entrées initialement stockées dans le tableau sont positives, et qu'un entier négatif indique une entrée supprimée.

Q.1 : Écrivez cette fonction de façon naïve avec une double boucle. Quelle est la complexité de l'algorithme utilisé ?

Q.2 : On suppose qu'on dispose d'une fonction `sort()` qui prend en entrée un tableau et sa taille et trie le tableau en place en utilisant un tri par comparaison de complexité minimale. Écrivez une nouvelle fonction de suppression utilisant un tri. Quelle est la complexité de votre algorithme (exprimée en fonction de celle de `sort()`) ?

1. Ou plus précisément, comprise entre 1 et $2^{64} - 1$.

Exercice 5 : Factorielle

Q.1 : Écrivez une fonction récursive qui calcule la factorielle de son entrée, en effectuant le calcul sur des entiers non signés de 64 bits.

Q.2 : Même question pour un calcul effectué sur des nombres flottants à double précision.

Q.3 : Expliquez comment calculer pour chacune des implémentations précédentes le nombre maximum pour lequel la valeur renvoyée est garantie exacte. De même pour une valeur renvoyée « approximativement correcte ». (On rappelle qu'un flottant double précision a une mantisse de 53 bits et un exposant de 11 bits.)