Exercices d'algorithmique

Benjamin Dallard

September 12, 2022

1 Question

Donner une définition de la complexité algorithmique et donner trois exemples de types de complexité temporelle en notation grand O.

2 Algorithmes

Pour chaque algorithme donner sa complexité ainsi qu'une justification.

2.1 Condition et produit

Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toute fois : on ne doit pas calculer le produit des deux nombres.

2.2 Additivté

Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer : $1 + 2 + 3 + 4 + 5 = 15$.

2.3 Somme des éléments d'un tableau

Ecrire un algorithme calculant la somme des valeurs d'un tableau (on suppose bien sur que le tableau a été préalablement saisi).

2.4 FacProduct

Toujours à partir de deux tableaux précédemment saisis, écrire un algorithme qui calcule le 'FacProduct' des deux tableaux. Pour calculer le 'FacProduct', il faut multiplier deux à deux les éléments des tableaux et additionner le tout (le résultat final doit donc est un réel).

2.5 Le plus grand

Ecrire un algorithme permettant à l'utilisateur de renvoyer la plus grande valeur du tableau passer en input en précisant quelle position elle occupe dans le tableau.

2.6 Plus grand que la moyenne

Ecrire un algorithme permettant à l'utilisateur de renvoyer le nombre de valeur supérieures à la moyenne d'un tableau passé en input de taille n.

2.7 Multiplier des vecteurs

Ecrire un algorithme permettant d'effectuer la multiplication de deux vecteurs de taille N connue à l'avance et d'afficher le résultat.

2.8 Tris

Écrire un algorithme permettant de trier un tableau de taille n.

2.9 Factorielle

Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle. On rappel qu'on calcul factorielle n tel que : $n! = \prod_{1in} i = 1 \times 2 \times 3 \times \ldots \times (n-1) \times n$.. Par exemple $4! = 1 \times 2 \times 3 \times 4$.

2.10 La multiplication matricielle

Ecrire un algorithme qui prend deux matrices carrées A et B de taille n et qui renvoie un élement $c_{i,j}$ (l'élement de la ligne i colonne j) de la matrice C tel que, $A \times B = C$.

2.11 Suite de Fibonacci

La suite de Fibonacci est définit récursivement par $F_0 = 0$, $F_1 = 1$ et $F_{n+2} = F_{n+1} + F_n$ pour $n \geq 0$. Ecrire une fonction FIBO(n) qui implémente la formule ci-dessus.

2.12 Le temps d'arrêt

On note H_n la somme $H_n = \sum_{k=1}^{n} \frac{1}{k}$. On admet que H tend vers $+\infty$. Écrire un algorithme qui détermine le plus petit entier n tel que (H) dépasse un réel α donné.