

Série semaine 1 : Solutions

1 Que font ces algorithmes ?

a) La sortie de l'algorithme 1 est non dans ce cas, et celle de l'algorithme 2 vaut 9 ($= |22 - 13|$).

b) L'algorithme 1 indique si la liste L est ordonnée dans l'ordre croissant ou non.

La sortie de l'algorithme 2 donne quant à elle la plus grande différence en valeur absolue entre deux éléments consécutifs de la liste.

2 Quel est le bon algorithme ?

Le bon algorithme est l'algorithme 2. L'algorithme 1 ne calcule que la somme des $n/2$ premiers nombres pairs; l'algorithme 3 calcule la somme des $2n$ premiers nombres entiers, et l'algorithme 4 produit une erreur à la première exécution de la ligne $s \leftarrow s + i$, car la valeur de s n'est pas initialisée.

3 Réparez-moi ces algorithmes !

- Le problème de l'algorithme de Jeanne est le suivant : tout se passe bien jusqu'à ce que $i = 1$ et $j = 5$, moment auquel l'algorithme rajoute 5 à s et recommence la boucle, en passant à $i = 2$. A ce 2^e passage, comme la valeur de j n'a pas changé, elle est toujours égale à 5, donc l'algorithme n'entre pas dans la boucle "Tant que" et ajoute de nouveau 5 à s , et ainsi de suite. La sortie de l'algorithme sera donc $s = 5n$ et non le résultat escompté. Pour remédier à ce problème, il faut incrémenter j après avoir mis à jour la valeur de s , comme suit :

algorithme de Jeanne corrigé
entrée : <i>nombre entier positif</i> n
sortie : <i>nombre entier positif</i> s
<pre> $s \leftarrow 0$ $j \leftarrow 1$ Pour i allant de 1 à n Tant que j n'est ni un multiple de 5 ni un multiple de 7 $j \leftarrow j + 1$ $s \leftarrow s + j$ $j \leftarrow j + 1$ Sortir : s </pre>

Ainsi, après avoir pris la valeur $j = 5$, j passe à 6, qui n'est plus ni un multiple de 5 ni de 7, et l'algorithme rentre dans la prochaine boucle "Tant que".

- Le problème de l'algorithme de Jean est le suivant : il existe des nombres qui sont à la fois des multiples de 5 et de 7 (35, par exemple). De tels nombres seront comptabilisés deux fois dans la somme s par l'algorithme. On peut réparer ce problème de deux manières :

algorithme de Jean corrigé, version 1
entrée : <i>nombre entier positif</i> n sortie : <i>nombre entier positif</i> s
<pre> s ← 0 i ← 1 j ← 1 Tant que $i \leq n$ j ← j + 1 Si j est un multiple de 5 s ← s + j i ← i + 1 Sinon Si j est un multiple de 7 s ← s + j i ← i + 1 Sortir : s </pre>

algorithme de Jean corrigé, version 2
entrée : <i>nombre entier positif</i> n sortie : <i>nombre entier positif</i> s
<pre> s ← 0 i ← 1 j ← 1 Tant que $i \leq n$ j ← j + 1 Si j est un multiple de 5 ou de 7 s ← s + j i ← i + 1 Sortir : s </pre>

4 Ecrivez un algorithme

a) Comme mentionné dans l'énoncé, il existe plusieurs façons de résoudre le problème. En voici deux :

algorithme 1
entrée : <i>liste</i> L <i>de nombres entiers, de taille</i> n , <i>nombre entier positif</i> x sortie : <i>variable binaire</i> s (<i>oui/non</i>)
<pre> s ← non Pour i allant de 1 à $n - 1$ Pour j allant de $i + 1$ à n Si $L(j) - L(i) > x$ s ← oui Sortir : s </pre>

algorithme 2
entrée : <i>liste</i> L <i>de nombres entiers, de taille</i> n , <i>nombre entier positif</i> x sortie : <i>variable binaire</i> s (<i>oui/non</i>)
<pre> s ← non min ← $L(1)$ max ← $L(1)$ Pour i allant de 2 à n Si $L(i) < \min$ min ← $L(i)$ Sinon, si $L(i) > \max$ max ← $L(i)$ Si $\max - \min > x$ s ← oui Sortir : s </pre>

Il en existe une troisième qui consiste à trier d'abord la liste L dans l'ordre croissant (nous verrons comment faire cela dans les semaines à venir), puis à comparer la différence $L(n) - L(1)$ (qui est égale à $\max - \min$ ci-dessus à droite) avec la valeur x .

b) Même s'il est légèrement plus facile à écrire, l'algorithme 1 effectue considérablement plus d'opérations que l'algorithme 2. En effet, tandis que l'algorithme 2 ne parcourt qu'une seule fois la liste L , l'algorithme 1 parcourt toutes les paires (i, j) d'éléments de la liste, qui sont beaucoup plus nombreuses que les éléments eux-mêmes. En effet, pour $n=1'000$, le nombre d'éléments de la liste vaut...1'000, tandis que le nombre de paires d'éléments vaut approximativement 500'000. Nous y reviendrons la semaine prochaine.

Note : Si on utilise la troisième possibilité ci-dessus qui consiste à trier d'abord la liste L , alors l'efficacité de l'algorithme dépend directement de l'efficacité de l'algorithme de tri utilisé.

5 Palindrome

On considère l'algorithme suivant :

algorithme
entrée : liste L de caractères, de taille n sortie : booléen s
$s \leftarrow \text{Vrai}$ Pour i allant de 1 à $\lfloor n/2 \rfloor$ Si $L(i) \neq L(n - i + 1)$ $s \leftarrow \text{Faux}$ Sortir : s

Pour quelle liste de lettres en entrée la sortie de l'algorithme ci-dessus est-elle Vrai ?

- ☐ MALAYALA
☐ CHACHA
☒ HAHHAH
☐ ALABAMA

6 Somme spéciale

On considère l'algorithme suivant :

somme spéciale
entrée : nombre entier n sortie : nombre entier $total$
$total \leftarrow 0$ Pour i allant de 0 à $n - 1$ par pas de 2 $total \leftarrow total + i$ Sortir : $total$

Quelle est la sortie de l'algorithme pour $n = 10$?

- ☐ 8
☐ 10
☒ 20
☐ 30