EPFL - Information, Calcul et Communication - Partie théorique

Série 1

1 Que font ces algorithmes?

On considère les deux algorithmes suivants:

algorithme 1

entrée : *liste L de nombres entiers, de taille n*
sortie : *variable binaire r (oui/non)*

```
r ⟵ oui
Pour i allant de 2 à n
    Si L(i − 1) > L(i)
        r ⟵ non
Sortir : r
```

algorithme 2

entrée : *liste L de nombres entiers, de taille n*
sortie : *nombre entier positif d*

```
Si n = 1
    Sortir : 0
d ⟵ |L(2) − L(1)|
Pour i allant de 3 à n
    Si |L(i) − L(i − 1)| > d
        d ⟵ |L(i) − L(i − 1)|
Sortir : d
```

a) Quel est la sortie de chacun de ces algorithmes si les données en entrée valent $L = (7, 14, 22, 13, 17)$ et $n = 5$?

b) De manière générale, quelle est la sortie de chacun de ces algorithmes?

2 Quel est le bon algorithme?

Lequel des quatre algorithmes suivants permet de calculer la somme des n premiers nombres pairs? (exemple: si $n = 4$, alors s doit valoir $2 + 4 + 6 + 8 = 20$) Expliquez pourquoi les autres ne fonctionnent pas!

algorithme 1

entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

```
s ⟵ 0
Pour i allant de 1 à n
    Si i est pair
        s ⟵ s + i
Sortir : s
```

algorithme 2

entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

```
s ⟵ 0
Pour i allant de 1 à n
    s ⟵ s + 2i
Sortir : s
```

algorithme 3

entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

```
s ⟵ 0
Pour i allant de 1 à 2n
    s ⟵ s + i
Sortir : s
```

algorithme 4

entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

```
Pour i allant de 1 à n
    s ⟵ s + i
Sortir : 2s
```

3 Réparez-moi ces algorithmes!

Un enseignant du cours ICC a demandé à Jeanne et à Jean d'écrire un algorithme qui calcule la somme des n premiers nombres entiers faisant partie de la liste des multiples de 5 ou de celle des multiples de 7 (ou non-exclusif). Voici ce que Jeanne et Jean ont écrit:

algorithme de Jeanne

entrée : *nombre entier positif* n
sortie : *nombre entier positif* s

```
s ⟵ 0
j ⟵ 1
Pour i allant de 1 à n
    Tant que j n'est ni un multiple de 5 ni un multiple de 7
        j ⟵ j + 1
    s ⟵ s + j
Sortir : s
```

algorithme de Jean

entrée : *nombre entier positif* n
sortie : *nombre entier positif* s

```
s ⟵ 0
i ⟵ 1
j ⟵ 1
Tant que i ≤ n
    j ⟵ j + 1
    Si j est un multiple de 5
        s ⟵ s + j
        i ⟵ i + 1
    Si j est un multiple de 7
        s ⟵ s + j
        i ⟵ i + 1
Sortir : s
```

Malheureusement, chacun des algorithmes ci-dessus a un problème (différent). Dans chacun des cas, voyez-vous lequel? Et pouvez-vous aider Jeanne et Jean à réparer leurs algorithmes respectifs?

Note: Pour identifier concrètement si un nombre j est un multiple de a, on vérifie si $j \bmod a = 0$, où $j \bmod a$ désigne le reste de la division euclidienne de j par a.

4 Ecrivez un algorithme

Soit L une liste de nombres entiers, de taille n, et x un nombre entier positif.

a) Ecrivez un algorithme dont la sortie soit oui s'il existe $i, j \in \{1, \ldots, n\}$ tels que $|L(i) - L(j)| > x$, et non dans le cas contraire.

Note: Il existe plusieurs façons d'écrire un tel algorithme!

b) Estimez maintenant l'efficacité de votre algorithme: combien d'opérations seront effectuées par celui-ci (approximativement) si la taille de la liste L vaut $n = 1'000$?

5 Pour le plaisir: un algorithme de tri inhabituel*

Un magicien a emprisonné cent lutins avec des bonnets rouges et bleus dans une caverne sans lumière. Au matin, les lutins doivent sortir un par un de la caverne et se présenter au magicien en ligne (de face, côte à côte) de sorte à ce que tous les lutins à bonnet rouge (R) soient d'un côté et tous les lutins à bonnet bleu (B) de l'autre; par exemple:

....RRRRRRRRBBBBBBBBB.....

Le problème est qu'aucun lutin ne connaît la couleur de son propre bonnet et qu'ils n'ont pas non plus le droit d'échanger des informations entre eux. Or si un lutin se présente au mauvais endroit, il est instantanément désintégré par le magicien. Comment vont faire ces cent lutins pour réussir à se présenter bien ordonnés et *tous* survivre?