

Validation d'algorithmes, devoir sur table

Important :

- Vous avez droit à une feuille recto-verso avec des notes manuscrites pour cette épreuve. Aucun autre document externe n'est autorisé. De même, tout appareil électronique est interdit.
 - Pour chaque question, une indication sur le temps maximal à investir est indiquée.
 - Pensez à vous relire.
 - Soyez clair, soignez la présentation. Un texte bref mais précis est plus utile qu'un discours long et confus.
-

Rappel de cours

On rappelle la syntaxe des expressions algébriques :

- AB : A puis B
 - $A+B$: A ou B
 - A^* : A plusieurs fois (potentiellement zéro fois)
 - ϵ : l'expression vide (si besoin)
 - $A?$: A zéro ou une fois, soit $(A|\epsilon)$
 - (A) : utilisez des parenthèses, les opérateurs ne se rapportent qu'au dernier élément, exemple AB^* veut dire A suivi de plusieurs B, mais $(AB)^*$ veut dire AB répété plusieurs fois.
-

Partie 1: Preuves formelles

Question 1.1 : Restitution de connaissances (15 minutes)

Répondez à chacune des questions en détaillant votre réponse.

- a - On a défini un variant comme une fonction d'un état mémoire vers un ensemble muni d'une relation d'ordre bien fondé, pourquoi la relation doit-elle être bien fondée ?
- b - Comment un invariant s'écrit-il ? À quoi sert-il exactement ? Pouvez vous expliciter en quoi est-il analogue à une hypothèse de récurrence ?

Entrées : l une lettre, m un mot
Output : Le nombre de l dans m

```
1  $n \leftarrow \text{longueur}(m)$ ;  
2  $r \leftarrow 0$ ;  
3  $i \leftarrow 1$ ;  
4 tant que  $i \leq n$  faire  
5   | si  $m[i] == l$  alors  
6   |   |  $r \leftarrow r + 1$ ;  
7   | fin  
8   |  $i \leftarrow i + 1$ ;  
9 fin  
10 retourner  $r$ 
```

Algorithme 1 : Comptage de lettre

Entrées : $a, b \in \mathbb{N}^*$
Output : $\text{pgcd}(a, b)$

```
1 tant que  $a \neq b$  faire  
2   | si  $a > b$  alors  
3   |   |  $a \leftarrow a - b$ ;  
4   | sinon  
5   |   |  $b \leftarrow b - a$ ;  
6   | fin  
7 fin  
8 retourner  $a$ 
```

Algorithme 2 : L'algorithme d'Euclide

Entrées : l liste d'entiers
Output : la liste l triée par ordre croissant.

```
1  $\text{stop} \leftarrow \text{faux}$ ;  
2  $n \leftarrow \text{length}(l)$ ;  
3 tant que  $\neg \text{stop}$  faire  
4   |  $\text{stop} \leftarrow \text{vrai}$ ;  
5   | pour  $i$  de 1 à  $n - 1$  faire  
6   |   | si  $l[i - 1] > l[i]$  alors  
7   |   |   |  $\text{intervertir}(l[i - 1], l[i])$ ;  
8   |   |   |  $\text{stop} \leftarrow \text{faux}$ ;  
9   |   | fin  
10  | fin  
11 fin  
12 retourner  $r$ 
```

Algorithme 3 : Tri bulle

Entrées : $a, b \in \mathbb{N}$
Output : $a \times b$

```
1  $r \leftarrow 0$ ;  
2 tant que  $a \neq 0$  faire  
3   | si  $a = 0 \bmod 2$  alors  
4   |   |  $a \leftarrow a/2$ ;  
5   | sinon  
6   |   |  $r \leftarrow r + b$ ;  
7   |   |  $a \leftarrow (a - 1)/2$ ;  
8   | fin  
9   |  $b \leftarrow 2b$ ;  
10 fin  
11 retourner  $r$ 
```

Algorithme 4 : Multiplication rapide

Question 1.2 : Variants et applications (15 minutes)

Pour les algorithmes suivants, veuillez expliquer pourquoi ils terminent. (Ils sont classés par ordre croissant de difficulté)

- a - l'algorithme de comptage d'une lettre dans un mot (algo 1)
- b - l'algorithme d'Euclide (algo 2)
- c - l'algorithme tri bulle (algo 3). Pour celui-ci on suppose que la boucle lignes 6 à 9 termine.

Question 1.3 : Invariant (15 minutes)

Pour les algorithmes suivants : expliquez pourquoi ils renvoient ce qu'il faut. (Ils sont classés par ordre croissant de difficulté)

- a - l'algorithme de comptage d'une lettre dans un mot (algo 1)
- b - l'algorithme multiplication rapide (algo 4). On supposera pour cette question avoir prouvé la terminaison.

Partie 2: Boite blanche

```
1 commandes1;  
2 tant que condition1 faire  
3   commandes2;  
4   si condition2 alors  
5     break;  
6   fin  
7   commandes3;  
8 fin  
9 commandes4;
```

Algorithme 5 : Bidouille abstraite

Question 2.1 : Graphe de flot de contrôle (10 minutes)

Veuillez tracer le graphe de flot de contrôle de l'algorithme 4. Nous nommons ce graphe \mathcal{G}_1 pour les prochaines questions. Ajoutez des lettres pour nommer ces nœuds.

Indication : Pour gagner du temps, pour les nœuds de commandes **uniquement**, vous pouvez écrire seulement les numéros de ligne. Pour tout nœud de condition, **recopiez** clairement la condition.

Question 2.2 : Graphe de flot de contrôle, variation (10 minutes)

Nous n'avons pas vu d'exemples avec l'instruction **break** en cour mais c'est assez facile à gérer sans ajouter de nœuds d'un nouveau type, pouvez-vous donner le graphe de flot de contrôle correspondant à l'algorithme 5. Nous nommons ce graphe \mathcal{G}_2 pour les prochaines questions. Ajoutez des lettres pour nommer ces nœuds.

Question 2.3 : Chemins (5 minutes)

Donnez les chemins des graphes \mathcal{G}_1 et \mathcal{G}_2 des questions précédentes sous forme d'expression algébrique.

Indication : Vous avez un rappel de cours en page 1.

Question 2.4 : Tests (10 minutes)

Pour l'algorithme 4, trouvez des données de tests pour couvrir le critère «tout les nœuds». **Justifiez** bien votre réponse.

Indication : Si vous avez, utilisez des crayons de couleur : vous pouvez peut-être les utiliser pour ajouter une information sur le graphe \mathcal{G}_1 .