TD 17 d'informatique - MPSI - Complexités et preuves de programmes - 22/04/2025

Exercice 1. Fonction mystère

Soit `f` la fonction :

```python
def f(L):
    """f(L: list[int]) -> int"""
    k = 0
    for i in range(1, len(L)):
        if L[i] < L[k]:
            k = i
    return L[k]
```

1. Quelle est la condition sur `L` pour que la fonction `f` ne déclenche pas d'erreur ? Que fait la fonction `f` ?
2. Déterminer la complexité en temps pire cas et meilleur cas de cette fonction.
3. Trouver un invariant de boucle permettant de prouver la correction de la fonction `f` (on demande seulement l'invariant, pas la preuve de la correction).

Exercice 2. Somme des produits

Soit `som_prod` la fonction :

```python
1 # n >= 1
2 def som_prod(n):
3     """som_prod(n: int) -> int"""
4     res = 0
5     for i in range(1,n+1):
6         for j in range(i, n+1):
7             res += i*j
8     return res
```

1. Que fait la fonction `som_prod` ?
2. Montrer la terminaison de la fonction.
3. Déterminer la complexité de cette fonction. On donnera le nombre exact d'exécutions de la ligne 7.

Afin de prouver la correction de la fonction, on se propose de trouver un invariant de boucle $(\mathcal{P})$ pour la boucle extérieure puis de montrer $(\mathcal{P})$ à l'aide d'un invariant de boucle $(\mathcal{Q})$ pour la boucle intérieure.

4. (a) Trouver un invariant de boucle $(\mathcal{P}_{\texttt{i}})$ pour la boucle `for i`.
   (b) On fixe un entier $\texttt{i} \in [\![1 ; \texttt{n}]\!]$. Trouver une invariant de boucle $(\mathcal{Q}_{\texttt{i},\texttt{j}})$ pour la boucle `for j`.
   (c) Montrer la correction de la fonction `som_prod`.

Exercice 3. Moyenne et écart type

Soit $n \in \mathbb{N}^*$ et $x_0, x_1, \ldots, x_{n-1} \in \mathbb{R}^n$. La moyenne et l'écart type des x_i notés m et σ sont définis par :

$$m = \frac{1}{n}\sum_{i=0}^{n-1} x_i, \qquad \sigma = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(x_i - m)^2}.$$

En d'autres termes, $\sigma = \sqrt{m'}$ où m' est la moyenne des y_i définis par $y_i = (x_i - m)^2$.

Afin de calculer la moyenne et l'écart type des valeurs d'un dictionnaire non vide, on utilise les deux fonctions suivantes :

```python
# d est non vide
def moy(d):
    """
    d: dict[_, float]
    Return: float
    """
    s = 0
    for c in d:
        s += d[c]
    return s/len(d)
```

```python
# d est non vide
import math
def ecart_type(d):
    """
    d: dict[_, float]
    Return: float
    """
    d1 = {}
    for c in d:
        y = (d[c] - moy(d))*(d[c] - moy(d))
        d1[c] = y
    return math.sqrt(moy(d1))
```

1. Trouver un invariant de boucle permettant de prouver la correction de la fonction `moy` (on demande seulement l'invariant, pas la preuve de la correction).
2. Donner la complexité de la fonction `moy`.
3. Trouver un invariant de boucle permettant de prouver la correction de la fonction `ecart_type` (on demande seulement l'invariant, pas la preuve de la correction).
4. Quel est le nombre exact d'appels à la fonction `moy` dans la fonction `ecart_type`. Donner la complexité de la fonction `ecart_type`.
5. Comment modifier la fonction `ecart_type` pour ne faire que deux appels à la fonction `moy` ? Qu'obtient-on comme complexité avec ces modifications ?

Exercice 4. Calcul du PGCD

Pour calculer le PGCD de deux entiers `(m, n)` $\in \mathbb{N}^2$, on utilise l'algorithme d'Euclide :

```python
# n >= 0, m >= 0
def PGCD(n,m):
    """PGCD(n: int,m: int) -> int"""
    a,b = n,m
    while b != 0:
        a,b = b, a % b
    return a
```

1. Montrer la terminaison de la fonction `PGCD`.
2. Montrer la correction de la fonction `PGCD`.

Pour calculer la complexité de la fonction `PGCD`, on va utiliser la suite de Fibonacci $(f_k)_{k\in\mathbb{N}}$ définie par :

$$f_0 = 0, \qquad f_1 = 1, \qquad f_{k+2} = f_{k+1} + f_k, \text{ pour } k \geqslant 0.$$

On utilisera également $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$ le nombre d'or. À partir de la relation $\phi^2 = 1 + \phi$, il est possible de montrer par récurrence sur $k \in \mathbb{N}$ que $\phi^k \leqslant f_{k+2} \leqslant \phi^{k+1}$ (ces inégalités sont admises dans la suite de l'exercice).

Étant donnés $(n, m) \in \mathbb{N}^2$ deux entiers, on note $c(n, m)$ le nombre de tours de boucle lors du calcul du PGCD de n et m. Dans un premier temps, on souhaite montrer que si n et m sont deux termes consécutifs de la suite de Fibonacci alors $c(n, m)$ est de l'ordre de $\log(n)$ (en fait c'est le pire cas pour l'algorithme d'Euclide, comme montré dans la fin de l'exercice).

3. (a) Si $k \geqslant 2$, montrer que f_k est le reste dans la division Euclidienne de f_{k+2} par f_{k+1}.
   (b) En déduire la valeur de $c(f_{k+1}, f_k)$ pour $k \in \mathbb{N}$.
   (c) On pose $n = f_{k+1}$ et $m = f_k$. Montrer que $c(n, m) = \Theta\Big(\log(n)\Big)$.

Pour terminer, on souhaite montrer que pour n et m quelconques, le nombre de tours de boucle dans l'algorithme d'Euclide est en $\mathcal{O}(\log(n))$.

4. On suppose $0 \leqslant m < n$.
   (a) Montrer par récurrence sur $c(n, m)$ que $n \geqslant f_{c(n,m)+2}$.
   (b) En déduire que la complexité temporelle de la fonction `PGCD` est en $\mathcal{O}(\log n)$. Est-elle en $\Theta(\log n)$?
5. Qu'en est-il du cas $0 \leqslant n \leqslant m$?

Exercices à faire pour la séance du 29/04/2025

Faites ces exercices sur feuille et apportez les lors de la séance de TP du 29/04/2025. Je ramasserai certaines copies au hasard.

Exercice 5.

Pour cet exercice, on pourra s'inspirer des preuves faites dans l'exercice 2. On s'intéresse à la fonction :

```python
def somLL(L):
    """somLL(L: list[list[int]]) -> int"""
    s = 0
    for i in range(len(L)):
        for j in range(len(L[i])):
            s += L[i][j]
    return s
```

1. Que renvoie la fonction `somLL` ?
2. Montrer la terminaison de la fonction.

Afin de prouver la correction de la fonction, on se propose de trouver un invariant de boucle $(\mathcal{P})$ pour la boucle extérieure puis de montrer $(\mathcal{P})$ à l'aide d'un invariant de boucle $(\mathcal{Q})$ pour la boucle intérieure. Pour cela :

→ Pour chaque $\texttt{i} \in \llbracket \texttt{0,len(L)-1} \rrbracket$, on imagine que le tour `i` de la boucle « `for i` » est en cours d'exécution. On note alors $\texttt{s}_\texttt{i}$ la valeur de la variable `s` à la fin du tour `i` (c'est à dire après que la boucle « `for j` » soit terminée).

→ Pour chaque $\texttt{i} \in \llbracket \texttt{0,len(L)-1} \rrbracket$ et chaque $\texttt{j} \in \llbracket \texttt{0,len(L[i])-1} \rrbracket$, on imagine que le tour `i` de la boucle « `for i` » est en cours d'exécution et que le tour `j` de la boucle « `for j` » est en cours d'exécution. On note alors $\texttt{s}'_{\texttt{i,j}}$ la valeur de la variable `s` à la fin du tour de boucle `j` (c'est à dire juste après l'exécution de la ligne 6).

3. (a) Trouver un invariant de boucle $(\mathcal{P}_\texttt{i})$ pour la boucle « `for i` ». Votre invariant doit exprimer la valeur de $\texttt{s}_\texttt{i}$ en fonction de `L` et de `i`.

   (b) Trouver un invariant de boucle $(\mathcal{Q}_{\texttt{i,j}})$ pour la boucle « `for j` ». Votre invariant doit exprimer la valeur de $\texttt{s}'_{\texttt{i,j}}$ en fonction de `L`, $\texttt{s}_{\texttt{i-1}}$, `i` et `j`.

   (c) À l'aide des invariants précédents, montrer la correction de la fonction `somLL`. Vous devez donc montrer que $(\mathcal{P}_i)$ et $(\mathcal{Q}_{i,j})$ sont bien des invariants, puis en déduire la correction de la fonction.

Exercice 6. Exponentiation rapide (facultatif)

Dans cet exercice, on considère $x \in \mathbb{R}$ et $n \in \mathbb{N}$ et on souhaite calculer x^n en temps $\Theta(\log n)$. Soit `puiss` la fonction suivante :

```
# n >= 0
def puiss(x,n):
    """puiss(x: float,n: int) -> float"""
    res = 1; y = x; m = n
    while m != 0:
        if m % 2 == 1:
            res = res * y
        m = m//2
        y = y*y
    return res
```

1. Montrer la terminaison de la fonction `puiss`.

Soit $K \in \mathbb{N}$ le nombre de tours de boucle. On numérote les tours de boucle par un entier $k \in \llbracket 0, K-1 \rrbracket$ ($k = 0$ correspond au premier tour de boucle, $k = 1$ correspond au deuxième tour et ainsi de suite). Pour $k \in \llbracket 0, K-1 \rrbracket$, on note res_k, y_k et m_k les valeurs des variables `res`, `y` et `m` au début du tour de boucle numéro k. De plus, on note res_K, y_K et m_K les valeurs des variables `res`, `y` et `m` à la fin de la fonction. Pour $k \in \llbracket 0, K \rrbracket$, on note $(\mathcal{P}_k)$ la relation suivante :

$$res_k \times y_k^{m_k} = x^n.$$

2. Montrer que $(\mathcal{P}_k)$ est un invariant de boucle et en déduire la correction de la fonction `puiss`.

On va maintenant montrer que la complexité temporelle de `puiss` est en $\Theta(\log n)$:

3. Montrer que pour tout $k \in \llbracket 0, K-1 \rrbracket$:

$$\frac{n}{2^k} - 1 \leqslant \frac{n}{2^k} - \sum_{i=1}^{k} \frac{1}{2^i} \leqslant m_k \leqslant \frac{n}{2^k}.$$

4. En déduire que la complexité en temps de `puiss` est en $\Theta(\log n)$.