

Devoir en temps libre

Pour lundi 29 avril 2024

Retravailler les cours de complexité et de preuves d'algorithmes (terminaison, corrections partielles, totales) pour tirer un meilleur parti du travail demandé. Recopier les réponses aux questions en mettant en valeur la structure afin d'être capable de la restituer et la mettre en oeuvre dans des situations similaires.

* * *

A. Complexité

1. Déterminer la complexité de la fonction

```

1 def u(n):
2     res = 1
3     for k in range(1, n):
4         res = res/2 + 1/k
5     return res

```

Réponse

Opérations élémentaires prises en compte :

- affectation
- opérations arithmétiques
- "return"

Pour l'appel de $u(n)$, on fait

- avant la boucle for (ligne 2) : une opération élémentaire,
- à chaque passage dans la boucle (pour k de 1 à $n - 1$), 4 opérations élémentaires,
- et une opération élémentaire après la boucle for,

donc on fait au total $C_u(n) = 1 + \sum_{k=1}^{n-1} 4 + 1 = -2 + 4n = O(n)$ opérations élémentaires.

Ainsi la fonction u a une complexité linéaire en la valeur de son paramètre n .

2. Déterminer la complexité de la fonction

```

1 def S_1(n):
2     res = 0
3     for k in range(1, n+1):
4         S = S + u(k)
5     return res

```

Réponse

Opérations élémentaires prises en compte :

- affectation
- opérations arithmétiques
- "return"

Pour l'appel de $S_1(n)$, on fait

- avant la boucle for (ligne 2) : une opération élémentaire,
- à chaque passage dans la boucle pour une valeur de k de 1 à n , $2 + (4k - 2)$ opérations élémentaires,
- et une opération élémentaire après la boucle for,

donc on fait au total $C_u(n) = 1 + \sum_{k=1}^n (2 + 4k - 2) + 1 = 1 + 4 \frac{n(n+1)}{2} = 1 + 2n(n+1) = O(n^2)$ opérations élémentaires.

Ainsi la fonction S_1 a une complexité quadratique en la valeur de son paramètre n .

B. Preuve

On s'intéresse à la fonction

```

1 def div_eucl (a, b):
2     """
3     a : entier >=0
4     b : entier > 0
5     renvoie liste [q, r], quotient et reste
6     dans division euclidienne de a par b """
7     r = a
8     q = 0
9     while r >= b:
10        r = r-b
11        q = q+1
12    return [q, r]

```

On veut en prouver la correction totale.

1. Prouver la terminaison de la fonction `div_eucl`.

Réponse

Le seul risque est une boucle "while" infinie.

Par l'absurde, supposons la boucle while infinie

Soit r_k la valeur de la variable r à la fin du k -ième passage dans la boucle while (donc avant le $k + 1$ -ième passage).

On a alors : $\forall k \in \mathbb{N}, r_{k+1} = r_k - b < r_k$,

donc $(r_k)_{k \in \mathbb{N}}$ est une suite strictement décroissante d'**entiers positifs** ($r_k \geq b$ pour tout k). Une telle suite n'existe pas donc on a une contradiction et la sortie de la boucle "while" est prouvée ainsi que la terminaison de la fonction `div_eucl`.

2. Prouver la correction de la fonction `div_eucl`.

Réponse

On a déjà vu la terminaison de la fonction `div_eucl`. On numérote les itérations de la boucle "while" par i à partir de 1 et on note r_i et q_i le contenu des variables `r` et `q` à la fin de la i -ième itération (r_0 et q_0 sont les valeurs à l'entrée dans la boucle).

On pose $\mathcal{H}(i) = \ll a = bq_i + r_i \text{ et } r_i \geq 0 \gg$ (on espère que c'est un invariant de boucle)

Initialisation.

Avant d'entrer dans la boucle, `r` contient `a` et `q` contient 0.

Ainsi $bq_0 + r_0 = 0 + a = a$, et $r_0 \geq 0$ car a est positif d'après la documentation de la fonction. $\mathcal{H}(0)$ est vraie.

Hérédité.

Supposons $\mathcal{H}(i)$ vraie et que l'itération numéro $i + 1$ a lieu.

On a donc $bq_i + r_i = a$ et $r_i \geq b$ (sinon on serait sorti).

L'exécution de la boucle donne $r_{i+1} = r_i - b \geq 0$ et $q_{i+1} = q_i + 1$.

D'où $bq_{i+1} + r_{i+1} = b(q_i + 1) + (r_i - b) = bq_i + r_i = a$.

Et $\mathcal{H}(i + 1)$ est vraie.

Exploitation.

À la sortie de la boucle (après le p ième passage), $a = bq_p + r_p$ et $r_p \geq 0$, et la condition `r >= b` est fautive, donc $r_p < b$ et $0 \leq r_p < b$. Donc q_p et r_p sont le quotient et le reste de la division euclidienne de a par b , donc la fonction renvoie bien ce qui est attendu.