

02 - Complexité - Vérification des algorithmes

Corrigé des exercices non traités en cours

Exercice 9 (*Extrait CCINP PSI 2026*) : .

Q7.

```
1 def distance(S1,S2):
2     n=len(S1)
3     somme=0
4     for i in range(n):
5         somme+=(S1[i]-S2[i])**2
6     return (somme/n)**(1/2)
```

Q8.

```
1 def positions_a_rejeter (S1,S2):
2     n=len(S1)
3     liste=[]
4     for i in range(n):
5         if S1[i]==None or S2[i]==None:
6             liste.append(i)
7     return liste
```

Q9.

```
1 def est_absent(element,L):
2     for x in L:
3         if x==L:
4             return False
5     return True
```

Q10.

On énonce l'invariant de boucle $\mathcal{P}(i)$ suivant : A l'issue du k ème tour de boucle (avec $0 \leq k \leq N$), on a :

$$d = \sum_{\substack{j=0 \\ j \notin \text{a_rejeter}}}^{k-1} (S1[j] - S2[j])^2$$

Remarque sur la correction : À l'issue de la dernière itération (lorsque $k = N$), l'invariant garantit que la variable

d vaut : $d = \sum_{\substack{j=0 \\ j \notin \text{a_rejeter}}}^{N-1} (S1[j] - S2[j])^2$

L'instruction finale renvoie alors $\sqrt{\frac{d}{m}}$, où $m = N - \text{len}(\text{a_rejeter})$ représente le nombre d'éléments conservés, ce qui assure la correction du résultat.

Q11. Complexité de distance2 :

- L'affectation de d , N et le calcul de m s'exécutent en $\mathcal{O}(1)$.
- La boucle `for i in range(N)` s'exécute N fois.
- À chaque itération i , la fonction `est_absent(i, a_rejeter)` parcourt la liste `a_rejeter` de taille r . Dans le pire des cas (recherche linéaire), cette vérification requiert $\mathcal{O}(r)$ opérations.

- Les opérations d'accumulation et d'élévation au carré s'exécutent en $\mathcal{O}(1)$.

Le coût temporel total de la boucle est donc $N \times \mathcal{O}(r) = \mathcal{O}(N \cdot r)$, soit finalement $C_{d1}(N, r) = \mathcal{O}(N \cdot r)$

Complexité de distance3 :

- On suppose que la duplication des listes avec `S1.copy()` et `S2.copy()` requiert le parcours et la copie de N éléments chacune, soit un coût de $\mathcal{O}(N)$.
- La première boucle `for i in a_rejeter` effectue r itérations. Les affectations par accès direct `S1c[i] = 0` et `S2c[i] = 0` s'effectuent en $\mathcal{O}(1)$, soit un coût total de $\mathcal{O}(r)$ pour cette boucle.
- La seconde boucle `for i in range(N)` effectue N itérations avec des opérations arithmétiques élémentaires en $\mathcal{O}(1)$, soit un coût de $\mathcal{O}(N)$.

Le coût global est donc la somme : $\mathcal{O}(N) + \mathcal{O}(r) + \mathcal{O}(N)$, soit finalement $C_{d2}(N, r) = \mathcal{O}(N) + \mathcal{O}(r)$

Exercice 11 (Extrait CCINP PSI 2021) :

Un point de mesure contient 38 caractères (5 horaire, 9 latitude, 10 longitude, 6 altitude, 3 fréquence, 5 virgule/saut de ligne), soit 38 octets. Ainsi une heure d'enregistrement représenterait environ 140Ko ($38 \times 3600 = 136800$).

Ainsi 200 heures requièrent donc 28Mo.

Exercice 12 : 1. (a) Un variant de boucle adapté pour la fonction **puissance** est la quantité : $v = k$

(b) Démontrons la terminaison de la boucle :

- Appartenance à $\mathbb{N}$:** Avant le premier tour de boucle, $k = n \geq 0$. Tant que la condition de boucle $k > 0$ est vérifiée, $v = k$ est un entier strictement positif ($v \in \mathbb{N}^*$).
- Décroissance stricte :** À chaque itération de la boucle, l'instruction `k = k - 1` diminue la valeur de k d'une unité. La nouvelle valeur v' du variant vérifie donc : $v' = v - 1 < v$

Le variant v est une quantité à valeurs entières positives qui diminue strictement à chaque passage dans la boucle. L'algorithme se termine donc nécessairement en un nombre fini d'étapes.

2. (a) Un invariant de boucle $\mathcal{P}$ reliant `res`, `x`, `k` et `n` s'énonce :

$$\mathcal{P} : (res = x^{n-k}) \quad \text{et} \quad (0 \leq k \leq n)$$

(b) Prouvons la correction de la fonction :

- Initialisation :** Avant d'entrer dans la boucle, `res = 1.0` et $k = n$. On a : $x^{n-k} = x^{n-n} = x^0 = 1.0 = res$
De plus, comme $n \in \mathbb{N}$, on a bien $0 \leq k \leq n$. L'invariant $\mathcal{P}$ est donc vrai initialement.
- Conservation :** Supposons $\mathcal{P}$ vrai au début d'une itération où la condition $k > 0$ est vérifiée (on a donc $res = x^{n-k}$ et $0 < k \leq n$). Notons res' et k' les valeurs à la fin de cette itération :

$$res' = res \cdot x \quad \text{et} \quad k' = k - 1$$

On calcule : $res' = (res) \cdot x = x^{n-k} \cdot x = x^{n-k+1} = x^{n-(k-1)} = x^{n-k'}$ De plus, comme $0 < k \leq n$, on a $0 \leq k - 1 < n$, c'est-à-dire $0 \leq k' \leq n$. L'invariant $\mathcal{P}$ est donc conservé à la fin de l'itération.

- Terminaison :** À la sortie de la boucle, la condition $k > 0$ devient fausse, ce qui donne $k \leq 0$. Comme la propriété $0 \leq k \leq n$ est conservée par l'invariant, on en déduit qu'à la sortie $k = 0$. En réinjectant $k = 0$ dans la relation d'invariant :

$$res = x^{n-0} = x^n$$

La fonction renvoie la valeur de `res`, soit x^n . La fonction est donc totalement correcte.