

TECHNOLAB-ISTA

Sujet d'examen : Complexité Algorithmique

Partie 1 : Questions de cours

1. **Définitions :**

- a) Définir la complexité en temps d'un algorithme.
- b) Expliquer la différence entre complexité en pire cas, cas moyen, et meilleur cas.
- c) Donner la définition formelle de la notation $O(\text{grand } O)$.

2. **Notation :**

Ranger les fonctions suivantes de la plus faible à la plus forte complexité :

$n, \log n, n^2, 2^n, n \log n$

Justifier en une phrase ou un exemple pour chaque comparaison.

3. **Vrai ou Faux :** Justifier brièvement chaque réponse.

- a) La complexité d'un algorithme récursif est toujours plus élevée que celle d'un itératif.
- b) Si un algorithme a une complexité $O(n \log n)$, alors il est plus rapide qu'un algorithme de complexité $O(n^2)$ pour tout n .
- c) Une complexité $O(1)$ signifie que le temps d'exécution dépend de la taille des données.
- d) La recherche dichotomique a une complexité en temps linéaire.
- e) La notation $\Theta(n)$ signifie que le temps d'exécution est exactement proportionnel à n .

4. **Exemples :** Donner un algorithme connu (nom et principe) pour chacune des complexités suivantes :

- a) $O(n)$
- b) $O(n^2)$
- c) $O(n \log n)$
- d) $O(\log n)$
- e) $O(2^n)$

Partie 2 : Analyse d'algorithmes

Exercice 1 : Analyse d'un algorithme simple

Considère la fonction suivante :

def mystere(n):

 s = 0

 for i in range(n):

 for j in range(i):

 s += 1

 return s

- 1. Quel est le rôle de cette fonction ?
- 2. Exprimer le nombre total d'itérations (opérations élémentaires) en fonction de n . (4 points)
- 3. Donner la complexité en notation O .
- 4. Peut-on l'optimiser ? Si oui, comment ?

Exercice 2 : Comparaison de deux algorithmes

Deux fonctions cherchent la valeur maximale d'un tableau d'entiers :

A

```
def max1(T):  
    max_val = T[0]  
    for i in range(1, len(T)):  
        if T[i] > max_val:  
            max_val = T[i]  
    return max_val
```

B

```
def max2(T):  
    for i in range(len(T)):  
        est_max = True  
        for j in range(len(T)):  
            if T[j] > T[i]:  
                est_max = False  
                break  
        if est_max:  
            return T[i]
```

1. Quelle est la complexité de chaque fonction ?
2. Laquelle est la plus efficace ? Pourquoi ?
3. Peut-on améliorer max2 ?
4. Quelle stratégie algorithmique utilise max2 ?