

Exercice 3: Analyse de la Complexité et Ajustement de Courbes pour les Fonctions en Python

1. Analyse de la Complexité des Fonctions :

1.1 Analyse de function_1:

function_1 contient des boucles imbriquées dont les complexités peuvent être analysées comme suit:

- La boucle externe s'exécute n^2 fois.
- La boucle interne s'exécute i fois.

La complexité temporelle de function_1 est $O(n^2)$ $O(n^2)$ à cause des boucles Imbriquées.

1.2 Analyse de function_2:

function_2 effectue des opérations linéaires en fonction de l'entrée n . Elle contient une

boucle qui s'exécute n fois. À l'intérieur de la boucle, une liste de taille n est générée puis

mélangée et sa valeur maximale est calculée.

Ce qui donne une complexité temporelle linéaire: $O(n)$ $O(n)$.

2. Vérification de la Complexité et Ajustement de Courbes :

2.1 Vérification pour function_1:

- Les temps mesurés sont collectés pour différentes valeurs de n .
- Un ajustement polynomial est appliqué, et le degré du polynôme est déterminé.
- La courbe polynomiale obtenue doit être de degré 2, en accord avec la complexité $O(n^2)$ $O(n^2)$.

2.2 Vérification pour function_2:

- Les temps mesurés sont collectés pour différentes valeurs de n .
- Un ajustement polynomial est appliqué, et le degré du polynôme est déterminé.
- La courbe polynomiale obtenue doit être de degré 1, en accord avec la complexité $O(n)$ $O(n)$.

3. Résultats :

3.1 Résultats de fonction_1:

- Les temps mesurés sont représentés en fonction de n .
- L'ajustement polynomial donne une courbe de degré 2.
- La complexité de fonction_1 est vérifiée comme étant $O(n^2)$ $O(n^2)$.

3.2 Résultats de fonction_2:

- Les temps mesurés sont représentés en fonction de n .
- L'ajustement polynomial donne une courbe de degré 1.
- La complexité de fonction_2 est vérifiée comme étant $O(n)$ $O(n)$.