TD Algorithmique - Complexité et Algorithmes de Tri

Pr. Lamia ZIAD

EST Essaouira

Exercice 1 : Analyse de complexité

Soient les fonctions de complexité suivantes :

- $f(n) = 3n^3 + 2n^2 + 5n + 1$
- $g(n) = n \log n + 2^n$
- $h(n) = 2^n + n!$
- $i(n) = \log(n!) + n^2$

Questions :

1. Donner la notation O (grand O) de chaque fonction
2. Classer ces fonctions par ordre de complexité croissante
3. Pour chaque fonction, déterminer si elle est :
   - Polynomiale
   - Exponentielle
   - Factorielle
   - Logarithmique

Exercice 2 : Tri par sélection

Algorithme :

```
def tri_selection(tableau):
    n = len(tableau)
    for i in range(n):
        min_index = i
        for j in range(i+1, n):
            if tableau[j] < tableau[min_index]:
                min_index = j
        tableau[i], tableau[min_index] = tableau[min_index], tableau[i]
    return tableau
```

Questions :

1. Appliquer l'algorithme manuellement sur le tableau : [64, 25, 12, 22, 11]
2. Compter le nombre de comparaisons et d'échanges
3. Donner la complexité temporelle dans le pire cas, meilleur cas et cas moyen
4. Proposer une optimisation possible

Exercice 3 : Tri rapide (Quicksort)

Principe :

1. Choisir un pivot
2. Partitionner le tableau : éléments $\leq$ pivot à gauche, éléments $>$ pivot à droite
3. Appliquer récursivement sur les sous-tableaux

Questions :

1. Implémenter la fonction de partition
2. Appliquer Quicksort sur [10, 80, 30, 90, 40, 50, 70] avec pivot = 50
3. Analyser la complexité dans le pire cas (tableau déjà trié)
4. Quelle stratégie de choix de pivot proposez-vous pour éviter le pire cas ?

Exercice 4 : Comparaison d'algorithmes

On considère trois algorithmes de tri :

— Tri par bulles : $O(n^2)$

— Tri par fusion : $O(n \log n)$

— Tri rapide : $O(n \log n)$ en moyenne

Questions :

1. Pour $n = 1000$, quel algorithme choisiriez-vous et pourquoi ?
2. Calculer le temps d'exécution pour $n = 10^6$ si une opération prend 1 ns
3. Quel algorithme est le plus adapté pour de petites tailles de tableau ?
4. Discuter de la stabilité et de la mémoire utilisée pour chaque algorithme