

Exercice 1 : Considérer deux algorithmes A1 et A2 avec leurs temps d'exécution $T1(n) = 9n^2$ et $T2(n) = 100n + 96$ respectives.

1. déterminer la complexité asymptotique des deux algorithmes dans la notation Grand-O.
2. quel algorithme a la meilleure complexité asymptotique?
3. montrer que vos solutions sont correctes en spécifiant un c et un n_0 par algorithme afin que la notation du majorant soit satisfaite (voir le support de cours).
4. pour quelle longueur de donnée n A1 ou A2 est le plus efficace ?
5. quelle est la complexité asymptotique de l'algorithme suivant? Quelle règle avez vous appliqué ?

Debut

Appeler A1

Appeler A2

Fin

Exercice 2 : Considérons le problème de nettoyage de données dont la spécification est la suivante :
TAB [N]

$\forall i : N - k < i \leq N - 1 \ a[i] = 0$

$\forall i : 0 \leq i \leq N - k \ a[i] \neq 0$

Un algorithme pour résoudre ce problème a donc pour objectif de déplacer toutes les occurrences de 0 vers la fin du tableau. A la fin de l'exécution, la variable k doit représenter le nombre de zéro se trouvant à la fin du tableau.

Q1 : proposez deux variantes pour ce nettoyage sans utiliser un tableau auxiliaire.

Q2 : mesurer les temps d'exécution.

Q3 : calculer la complexité de chaque variante (à faire en TD).

Exercice 3 : recherche séquentielle vs. dichotomique

1. Comparer les temps d'exécution des deux méthodes pour des tailles : 10^3 , 10^4 , 10^5 , 10^6
2. Comparer la complexité asymptotique des deux variantes (à faire en Cours-TD).

Exercice 4 : problème de l'empilement (stacking problem).

Un câblage n'a plus en stock qu'une seule bobine de câble de K mètres. Il a reçu n commandes consistant chacune en une certaine longueur de câble, qui ne peut être fragmentée. Il veut maximiser la longueur total de câble qu'il peut servir avec l'unique bobine restante.

Voici une application numérique :

$n = 10$

$c = \{2463, 2597, 315, 1089, 759, 599, 1283, 1683, 1625, 2292, 211\}$

$k = 67910$

une solution : $67910 = 2597 + 315 + 759 + 1283 + 1625 + 211$

Mesurer le temps d'exécution pour des ensembles de taille 10, 20, 30, ... -

Quelle est la complexité de votre algorithme ? (à faire en TD).

Exercice 5 : Tris Bulle/Sélection/Insertion vs. Fusion

Mesurer les temps d'exécution des deux variantes Bulle/Sélection/Insertion et Fusion sur des instances/tailles 10^4 , 10^5 , 10^6 , 10^7

à faire en TD : Calculer la complexité asymptotique des deux algorithmes.

Exercise 6 :

Consider the following algorithms with execution time $T(n)$ for data size n .
Determine their respective asymptotic complexity.

- **Algorithm A1:** $T(n) = 3n + 2$

- **Algorithm A2:** $T(n) = 6$

- **Algorithm A3:** $T(n) = 4n^2 + n + 2$

- **Algorithm A4 :**

execute A1;

execute A2;

execute A3;

- **Algorithm A5 :**

For i from 1 to n do

execute A3;

Endfor

execute A1;

- **Algorithm A6 :**

For i from 1 to 5 do

execute A1;

Endfor

Exercise 7 :

Consider the following algorithms

Algorithm 1	Algorithm 2
<pre> 1. x = n ; 2. while (x > 0) { 3. y = n ; 4. while (y > 0) { 5. y = y - 1 ; 6. } 7. x = x / 2 ; 8. }</pre>	<pre> 1. x = n ; 2. while (x > 0) { 3. y = x ; 4. while (y > 0) { 5. y = y - 1 ; 6. } 7. x = x / 2 ; 8. }</pre>

Calculate the asymptotic complexity of the two algorithms? Justify?

