STRUCTURES DE DONNÉES ET ALGORITHMES · CHAPITRE 2

Mesurer la croissance du travail

Définissez taille, modèle de coût, O, Ω et Θ. Distinguez meilleur cas, pire cas, moyenne et coût amorti avant de comparer des structures.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Définir les bornes O, Ω et Θ et le modèle de coût.
- Annoncez la taille et l'opération comptée avant de comparer les croissances.
- O donne une borne supérieure, Ω une borne inférieure et Θ les deux ; les cas moyen, pire et amorti sont des distinctions séparées.

Prérequis. Avoir étudié les chapitres DSA précédents ; savoir suivre une trace et un invariant.

Partir d'un besoin

Une méthode rapide sur dix valeurs peut devenir impraticable sur un million. Pour prévoir cette évolution, comptons une opération précise avant de chronométrer un programme.

Définir la structure et les conventions

La **taille n** mesure ici le nombre de valeurs. Un **modèle de coût** fixe ce que l'on compte et ce que l'on suppose constant : accès à un élément et comparaison de petits nombres, par exemple. Un chronomètre mesure aussi la machine et l'environnement ; un comptage explique le travail de la méthode.

Pour des fonctions de coût non négatives, **O(g(n))** signifie qu'il existe une constante C > 0 et un seuil n_0 tels que le coût ne dépasse pas C × g(n) pour n ≥ n_0. **Ω(g(n))** donne une borne inférieure de même forme ; **Θ(g(n))** réunit les deux. Une borne O n'est pas forcément serrée : n est aussi O(n²), mais Θ(n) est plus précis.

Le **pire cas** maximise le coût parmi les entrées de taille n ; le **meilleur cas** le minimise. Un coût **moyen** exige un modèle de distribution des entrées. Un coût **amorti** répartit le coût total d'une suite d'opérations : il n'exige pas de supposer des entrées aléatoires. Ces qualificatifs sont distincts de O et Θ.

La **mémoire auxiliaire** est le stockage ajouté à l'entrée, en précisant si l'on exclut la sortie. Elle compte aussi la pile des appels récursifs. Les tailles de très grands entiers et les opérations sur des textes longs peuvent rendre notre hypothèse de coût constant inadaptée.

Suivre les états

Exemples de croissance

n	n	n(n - 1) / 2	Divisions par deux jusqu'à zéro
4	4	6	3
8	8	28	4
16	16	120	5

Comparer les réalisations

Les paires valent 28 et les divisions par deux valent 4 pour n = 8. Une plage dont le début dépasse la fin ne produit aucun tour dans ces exemples. Les indices du pseudo-code commencent à 0, ceux de MATLAB à 1.

Pseudo-code

```
// example.pseudo
// count_growth
pairs <- 0
FOR left FROM 0 TO n - 1 INCLUSIVE DO
    FOR right FROM left + 1 TO n - 1 INCLUSIVE DO
        pairs <- pairs + 1
    END FOR
END FOR
PRINT pairs
```

Python

```
# count_growth.py
n = 8
pairs = 0
for left in range(n):
    for right in range(left + 1, n):
        pairs = pairs + 1
remaining = n
halvings = 0
while remaining > 0:
    remaining = remaining // 2
    halvings = halvings + 1
print(pairs, halvings)
```

MATLAB

```
% count_growth.m
n = 8;
pairs = 0;
for left = 1:n
    for right = left + 1:n
        pairs = pairs + 1;
    end
end
remaining = n;
halvings = 0;
while remaining > 0
    remaining = floor(remaining / 2);
    halvings = halvings + 1;
end
disp([pairs, halvings]);
```

Justifier et compter

Le comptage des paires vaut n(n - 1)/2 et est Θ(n²). Pour n ≥ 1, le nombre de divisions vaut floor($\log_2(n)$) + 1 : il est Θ(log n) lorsque n grandit. $\log_2(n)$ est l'exposant qui donne n comme puissance de deux ; sa base change seulement un facteur constant. Les deux compteurs utilisent une mémoire auxiliaire Θ(1).

Prévoir puis expérimenter

Prévoyez les résultats pour n = 0, 1, 4 et 16. Doubler n quadruple approximativement le nombre de paires pour de grandes tailles, mais ajoute un seul tour à la division successive par deux.

Exercice corrigé

▼ **Une opération O(1) amorti est-elle toujours instantanée ?**

Non. Une opération particulière peut coûter Θ(n), par exemple lors d'une réallocation. C'est le coût total d'une suite d'opérations, divisé par leur nombre, qui reste borné par une constante.

À retenir

- Annoncez la taille et l'opération comptée avant de comparer les croissances.
- O donne une borne supérieure, Ω une borne inférieure et Θ les deux ; les cas moyen, pire et amorti sont des distinctions séparées.
- La mémoire au pic inclut les appels et les copies encore vivantes ; elle ne se confond pas avec la somme de toutes les allocations.

Version web : https://www.clearcode.fr/courses/programming/dsa/foundations/complexity