

Complexité

Exercices

1. **a.** Montrer que g est dans $O(f)$ si et seulement si $\limsup_{x \rightarrow \infty} \frac{|g(x)|}{f(x)} = c < \infty$.
b. Montrer que $O(f)$ est un espace vectoriel réel.
2. **a.** Montrer que g est dans $o(f)$ si et seulement si $\lim_{x \rightarrow \infty} \frac{g(x)}{f(x)} = 0$.
b. Montrer que $o(f)$ est un sous-espace vectoriel réel de $O(f)$.
3. Soient f et g des fonctions positives.
a. Montrer que $O(g) \leq O(f)$ si et seulement si g est une fonction positive de $O(f)$.
b. Si $g \in O(f)$ est positive, alors $O(g) + O(f) = O(f)$.
4. Pour des fonctions $f_1, \dots, f_t$ positives, et scalaires réels $\alpha_1, \dots, \alpha_t$ positifs, montrer les identités
a. $\sum_{i=1}^t O(f_i) = O(\sum_{i=1}^t \alpha_i f_i)$ et en particulier, $\sum_{i=1}^t O(f_i) = O(\sum_{i=1}^t f_i)$.
b. $\prod_{i=1}^t O(f_i) = O(\prod_{i=1}^t f_i)$.
Conclure que $(O(f_1) + O(g_1))(O(f_2) + O(g_2)) = O((f_1 + g_1)(f_2 + g_2))$ (cf. exercice 6).
5. Définir la somme $(f_1 + O(g_1)) + (f_2 + O(g_2))$ et démontrer que
$$(f_1 + O(g_1)) + (f_2 + O(g_2)) = (f_1 + f_2) + O(g_1 + g_2).$$
L'ensemble $f_i + O(g_i)$ porte quelle structure ?
6. Définir le produit $(f_1 + O(g_1))(f_2 + O(g_2))$ et démontrer que
$$(f_1 + O(g_1))(f_2 + O(g_2)) \subseteq f_1 f_2 + O(f_1 g_2 + f_2 g_1 + g_1 g_2) \\ = (f_1 + O(g_1))(f_2 + O(g_2)) + O(g_1 g_2).$$
Lesquelles des fonctions f_1, f_2, g_1, g_2 doivent être positives ?
Si $g_1 \in O(f_1)$ ou $g_2 \in O(f_2)$, alors
$$(f_1 + O(g_1))(f_2 + O(g_2)) = f_1 f_2 + O(f_1 g_2 + f_2 g_1 + g_1 g_2),$$
et également si $f_1 \in O(g_1)$ où $f_2 \in O(g_2)$ (dans ce dernier cas pour des raisons triviales car $f_1 + O(g_1) = O(g_1)$ ou $f_2 + O(g_2) = O(g_2)$).
7. Trouver une expression pour la classe $(x + O(x^{1/2}))(x + O(\log(x)))^2$.
8. Montrer qu'il existe des fonctions f et $g : \mathbb{R}_{>0} \rightarrow \mathbb{R}$ telles que $g \notin O(f)$ et $f \notin O(g)$.
Indication. On peut construire des fonctions comme combinaison linéaire avec un terme dominant multiplié par des fonctions périodiques $\sin^2(x)$ et $\cos^2(x)$.
9. Répéter exercice 4 en remplaçant ' O ' par ' o '.
10. Supposer que $f, g : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$. Montrer l'équivalence des expressions suivantes
 - a.** $f \sim g$,
 - b.** $f - g \in o(g)$,
 - c.** $f/g \in 1 + o(1)$,
 - d.** $f/g \sim 1$.

Complexité des programmes

On donne un exercice qui sert comme un mini-texte de modélisation. On peut envisager de préparer un exposé court autour de cet exercice, en intégrant des expériences et experimentation en Python/Sage.

Exercice de modélisation. Soit donné les algorithmes ci-dessous, avec donnée d'entrée n .

a.

```
def runtime1(n):  
    m = 0  
    for i in range(n):  
        m += 1  
    return m
```

b.

```
def runtime2(n):  
    m = 0  
    for i in range(n):  
        m += i  
    return m
```

c.

```
def runtime3(n):  
    m = 0  
    for i in range(n):  
        m *= 2  
        m += 1  
    return m
```

d.

```
def runtime4(n):  
    m = 0  
    for i in range(n):  
        m *= 2  
        m += i  
    return m
```

Déterminer, pour chaque programme, une expression en forme de somme pour la valeur de sortie, la taille (en bits) de cette valeur, et la complexité de l'algorithme, en fonction de $\log_2(n)$. Vérifier que cette complexité correspond bien au temps de calcul à l'ordinateur.

Indication. Pour le programme `runtime4`, on peut formuler et démontrer l'identité suivante pour l'expression de la somme représentant la valeur de sortie.

Lemme.

$$\sum_{i=0}^{n-1} i 2^{n-i-1} = 2^n - n - 1.$$