

Devoir 1

Remise : sous forme *papier*, au plus tard le ME 20 septembre à l'heure du *début* de la démo

- Un algorithme A prend $\frac{3^n}{100}$ secondes et un algorithme B prend $\frac{n^6}{100}$ secondes pour résoudre un exemplaire de taille n d'un problème.
 - Quelles tailles maximales d'exemplaires A et B peuvent-ils résoudre en 30 jours? *What are the maximal instance sizes that A and B can solve in 30 days?*
 - Quelles seraient ces tailles sur un ordinateur un million de fois plus rapide? *How about on a computer one million times faster?*
 - B est-il infiniment souvent plus rapide que A ? *Is B infinitely often faster than A ?*
- Résolvez le problème 3.5 de BB :

Problem 3.5. Which of the following statements are true? Prove your answers.

- $n^2 \in O(n^3)$
- $n^2 \in \Omega(n^3)$
- $2^n \in \Theta(2^{n+1})$
- $n! \in \Theta((n+1)!)$

Note : your *proofs* here are only allowed to use the definitions of O , Ω , Θ and basic arithmetic.

- Considérez le problème suivant :

COMPOSITION

DONNÉE: f, f_1, \dots, f_k des applications $\{1, 2, \dots, m\} \rightarrow \{1, 2, \dots, m\}$.

DÉCIDER: s'il existe $\ell \geq 1$ et i_1, i_2, \dots, i_ℓ tels que $f = f_{i_1} \circ f_{i_2} \circ \dots \circ f_{i_\ell}$ où \circ représente la composition de fonctions.

- Donnez une méthode `composition(s,f)` en Python, commentée, résolvant COMPOSITION lorsque s est l'ensemble $\{f_1, \dots, f_k\}$. Plus précisément, f est de type `tuple` et s est un ensemble d'objets de type `tuple`. Par exemple,

```
>>> f11 = tuple([3, 1, 3, 4]) # 1->3 2->1 3->3 4->4
>>> f12 = tuple([3, 4, 2, 1]) # tuples car une liste ne peut pas
>>> s1 = set([f11, f12])      # servir d'élément d'un ensemble
>>> f1 = tuple([1, 1, 1, 2])
>>> composition(s1, f1)
True
```

Indice. Une méthode “bête” vaudra tous ses points. Supposez que l'entrée est de la bonne forme.

- Le fichier `exemplaires.py` fournit 6 exemplaires de COMPOSITION. Vous pouvez copier ce fichier ou importer `exemplaires` dans votre méthode. Complétez ensuite au mieux, avec ou sans l'aide de votre méthode (*sur papier, à la main*) le tableau ci-dessous :

Exemplaire	Valeur réponse sur cet exemplaire
s1, f1	True
s2, f2	
s3, f3	
s4, f4	
s5, f5	
s6, f6	

(c) Imprimez votre méthode et joignez-la à votre devoir.

(d) Déposez votre méthode sur Studium pour que Maëlle puisse la tester.

English translation In the above, f, f_1, \dots, f_k are total functions from $\{1, 2, \dots, m\}$ to $\{1, 2, \dots, m\}$. The COMPOSITION problem is to determine whether there exist $\ell \geq 1$ and i_1, i_2, \dots, i_ℓ such that the composition $f_{i_1} \circ f_{i_2} \circ \dots \circ f_{i_\ell}$ yields the function f .

You are asked to produce a Python method, properly commented, that solves COMPOSITION. See the example above for the input and output expected. Any method, including brute force, is fine. No need to check input formats and such.

Then you are asked to fill in the above table *by hand*, as best you can, with or without the help of your Python program, for the 6 instances provided by the file `exemplaires.py` on Studium.

Please hand in a printed copy of your method together with your assignment, and also upload your method on Studium for Maëlle to be able to test it.

4. Soit $S = O((2n)^3 \log n) \cup \Omega(n^5)$ et

$$\begin{aligned} f : \mathbb{N} &\rightarrow \mathbb{R}^+ \\ n &\mapsto n^4 (\log n)^2. \end{aligned}$$

Justifiez que $f \notin S$, en vous servant des limites (cf. transparents ou encore BB p. 88) si vous le souhaitez. Que pouvez-vous dire de $S \cap \Theta(f)$? De $S \cup \Theta(f)$? Pourquoi?

Justify that $f \notin S$, using the theorem on limits if you wish. What can you say about $S \cap \Theta(f)$? $S \cup \Theta(f)$? Why?