

IFT2125 automne 2016 - Devoir1

Philippe Caron

21 septembre 2016

1

Soient les entiers $a, b \in \mathbb{N}^{>0}$, on peut exprimer chacun de ses entiers comme le produit de facteurs premiers.

$$a = 2^{a_1} \cdot 3^{a_2} \cdot 5^{a_3} \cdot \dots \quad (1)$$

Le PGCD de deux nombre est donc le nombre formé par le produit des plus petites puissances présentes de facteurs communs aux deux nombres. Il y va donc de soit que le PPCM de deux nombre soit le produit de ceux-ci divisés par le PGCD ; ce faisant on élimine tous les éléments «redondant».

Prenons par exemple les entiers 2940 et 9450 :

$$2940 \times 9450 = \underbrace{2^2}_{\text{PGCD}=210} \cdot \underbrace{3^1} \cdot \underbrace{5^1} \cdot \underbrace{7^2}_{\text{PPCM}=132\,300} \times \underbrace{2^1} \cdot \underbrace{3^3} \cdot \underbrace{5^2} \cdot \underbrace{7^1}$$

Pour calculer **ppcm**(1234, 567, 89) on fait d'abord la factorisation de chacun des nombre :

$$1234 \times 567 \times 89 = 2 \cdot 617 \times 3^4 \cdot 7 \times 89$$

Réponse 1 : On constate que les trois nombres n'ont aucun facteur commun, le PGCD est donc 1 et le PPCM le résultat du produit des trois nombre, soit 62 271 342.

Tel que mentionné plus haut, on divise le produit final par le PGCD afin d'éliminer les facteurs qui sont commun aux nombre qu'on multiplie, mais qu'arrive-t-il quand on multiplie plus que deux nombres ?

Par exemple :

$$6 \times 9 \times 15 = 2^1 \cdot 3^1 \times 3^2 \times 3^1 \cdot 5^1$$

Dans ce cas, la puissance minimale commune aux trois nombres est 3^1 , on pourrait donc croire que :

$$\mathbf{ppcm}(6, 9, 15) = \frac{6 \cdot 9 \cdot 15}{3} = \frac{810}{3} = 270$$

Mais en examinant le résultat, on réalise que tous les résultats des divisions par 6 (45), par 9 (30) et par 15 (18), sont divisibles par 3, un facteur commun aux trois nombre ! Ce qui signifie que 270 n'est pas le PPCM.

Ceci se produit parce qu'en divisant par 3, on élimine la redondance du facteur dans la première multiplication (ex : 6×9), mais il reste encore un facteur 3 dans 15 qui n'a pas été éliminé. Pour arriver à la bonne solution, il suffit d'enlever ce facteur en divisant une fois de plus par 3.

$$\frac{270}{3} = 90$$

On peut vérifier que 90 est bien le PPCM, en divisant par 6 (15), par 9 (10) et par 15 (6) ; les résultats 15, 10, et 6 n'ont aucun facteur commun, on confirme que 90 est bien le PPCM.

Réponse 2 : De manière générale, on peut ajuster la règle pour une multiplication de k entiers en s'assurant que le PGCD est enlevé à chaque fois qu'on rajoute un entier. On a :

$$\forall a_1, \dots, a_k \in \mathbb{N}^{>0} \text{ où } k \geq 2, \mathbf{ppcm}(a_1, \dots, a_k) = \frac{a_1 \times \dots \times a_k}{[\mathbf{pgcd}(a_1, \dots, a_k)]^{k-1}} \quad (2)$$

2

(a)

Voici le code en python :

```
1 def estCongru(n, t):
2     return (n - t[0]) % t[1] == 0
3
4 def nonCongru(p):
5     k = 1
6     l = len(p)
7     for i in range(1):
8         k = ppcm(k, p[i][1])
9     for i in range(1, k):
10        for j in range(1):
11            if estCongru(i, p[j]):
12                break
13            if j == l - 1:
14                return i
15    return 0
16
17 def pgcd(a, b):
18     while b:
19         a, b = b, a % b
20     return a
21
22 def ppcm(a, b):
23     return a* b // pgcd(a, b)
```

(b)

tuple	Valeur réponse de x sur cet exemplaire
p1	
p2	
p3	
p4	
p5	

3

Soit les fonctions de temps respectives pour les algorithmes A et B suivantes :

$$t_A(n) = \frac{2^n}{1000} \qquad t_B(n) = \frac{n^4}{10}$$

(a) On calcule n pour 1 semaine (où $\lg(x) = \log_2(x)$) :

$t_A(n)$:

$$\begin{aligned} \frac{2^n}{1000} &= 60 \times 60 \times 24 \times 7 \\ \Leftrightarrow 2^n &= 604\,800\,000 \\ \Leftrightarrow \lg(2^n) &= \lg(604\,800\,000) \\ \Rightarrow n &\approx 29.1719 \end{aligned}$$

$t_B(n)$:

$$\begin{aligned} \frac{n^4}{10} &= 60 \times 60 \times 24 \times 7 \\ \Leftrightarrow n^4 &= 6\,048\,000 \\ \Leftrightarrow n &= \pm \sqrt[4]{6\,048\,000} \\ \Rightarrow n &= \sqrt[4]{6\,048\,000} \\ \Rightarrow n &\approx 49.5910 \end{aligned}$$

(b) Pour un ordinateur 1 000 000 fois plus rapide :

$t_A(n)$:

$$\begin{aligned} \frac{2^n}{1000} &= 60 \times 60 \times 24 \times 7 \times 1\,000\,000 \\ \Leftrightarrow 2^n &= 604\,800\,000\,000\,000 \\ \Leftrightarrow \lg(2^n) &= \lg(604\,800\,000\,000\,000) \\ \Rightarrow n &\approx 49.1035 \end{aligned}$$

$t_B(n)$:

$$\begin{aligned} \frac{n^4}{10} &= 60 \times 60 \times 24 \times 7 \times 1\,000\,000 \\ \Leftrightarrow n^4 &= 6\,048\,000\,000\,000 \\ \Leftrightarrow n &= \pm \sqrt[4]{6\,048\,000\,000\,000} \\ \Rightarrow n &= \sqrt[4]{6\,048\,000\,000\,000} \\ \Rightarrow n &\approx 1568.2054 \end{aligned}$$

(c) **Algorithme le plus rapide :**

On calcule la limite pour connaître l'algorithme le plus rapide à l'infini (RH pour Règle de l'Hôpital) :

$$\begin{aligned}
\lim_{n \rightarrow +\infty} \frac{t_A(n)}{t_B(n)} &= \lim_{n \rightarrow +\infty} \frac{\frac{2^n}{10}}{\frac{n^4}{100}} \\
&= \frac{1}{100} \cdot \lim_{n \rightarrow +\infty} \frac{2^n}{n^4} \\
&\stackrel{RH}{=} \frac{1}{100} \cdot \lim_{n \rightarrow +\infty} \frac{(2^n)'}{(n^4)'} \\
&= \frac{1}{100} \cdot \lim_{n \rightarrow +\infty} \frac{n \cdot 2^{n-1}}{4 \cdot n^3} \stackrel{n>1}{>} \frac{1}{400} \cdot \lim_{n \rightarrow +\infty} \frac{2^{n-1}}{n^3} \\
&\stackrel{RH}{=} \frac{1}{400} \cdot \lim_{n \rightarrow +\infty} \frac{(n-1) \cdot 2^{n-2}}{3 \cdot n^2} \stackrel{n>2}{>} \frac{1}{1200} \cdot \lim_{n \rightarrow +\infty} \frac{2^{n-2}}{n^2} \\
&\stackrel{RH}{=} \frac{1}{1200} \cdot \lim_{n \rightarrow +\infty} \frac{(n-2) \cdot 2^{n-3}}{2 \cdot n} \stackrel{n>3}{>} \frac{1}{2400} \cdot \lim_{n \rightarrow +\infty} \frac{2^{n-3}}{n} \\
&\stackrel{RH}{=} \frac{1}{2400} \cdot \lim_{n \rightarrow +\infty} \frac{(n-3) \cdot 2^{n-3}}{1} \\
&= +\infty
\end{aligned}$$

On calcule les termes des deux fonctions depuis 0 :

$t_A(1) = \frac{1}{500}$	$t_B(1) = \frac{1}{10}$
$t_A(2) = \frac{1}{250}$	$t_B(2) = \frac{8}{5}$
$t_A(3) = \frac{1}{125}$	$t_B(3) = \frac{81}{10}$
...	...
$t_A(10) = \frac{128}{125}$	$t_B(10) = 1000$
...	...
$t_A(25) = 33\,554.4$	$t_B(25) = 39\,062.5$
$t_A(26) = 67\,108.9$	$t_B(26) = 45\,697.6$

L'algorithme B n'est pas toujours plus rapide, mais il le sera systématiquement à partir de $n = 26$.

4

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

$$\begin{aligned} f : \mathbb{N} &\rightarrow \mathbb{R}^+ \\ n &\mapsto (n^3 - 3n^2 + 11)(\log n)^2 \end{aligned}$$

À partir des définitions Pour que $f \notin S$, f ne doit être ni dans $O(n^3 \log n)$ ni dans $\Omega(n^4)$.

La définition de l'ordre va comme suit :

$$O(f) = \{t : \mathbb{N} \rightarrow \mathbb{R}^{\geq 0} \mid (\exists c \in \mathbb{R}^{\geq 0})(\forall n \in \mathbb{N})[t(n) \leq cf(n)]\}$$

donc si $f \in O(n^3 \log n)$ il existe un c tel que pour tout n assez grand :

$$(n^3 - 3n^2 + 11)(\log n)^2 \leq c(n^3 \log n) \quad (3)$$

On sait que la croissance d'un polynôme n'est caractérisée que par son degré le plus élevé, on peut donc réécrire l'équation (3) comme suit quand n est très grand :

$$\begin{aligned} n^3(\log n)^2 &\leq c \cdot n^3 \log n & , n > n_0 \\ \log n &\leq c & , n > n_0 \end{aligned}$$

Or, un tel c ne peut pas exister puisque la fonction \log croît à l'infini.

La définition du oméga va comme suit :

$$\Omega(f) = \{t : \mathbb{N} \rightarrow \mathbb{R}^{\geq 0} \mid (\exists d \in \mathbb{R}^{\geq 0})(\forall n \in \mathbb{N})[t(n) \geq df(n)]\}$$

donc si $f \in \Omega(n^4)$ il existe un d tel que pour tout $n > n_0$:

$$(n^3 - 3n^2 + 11)(\log n)^2 \geq d(n^4) \quad (4)$$

Par la même astuce que tantôt on réécrit l'équation (4) :

$$\begin{aligned} n^3(\log n)^2 &\geq dn^4 & , n > n_0 \\ (\log n)^2 &\geq dn & , n > n_0 \\ \log n &\geq \sqrt{dn} & , n > n_0 \end{aligned}$$

Or un tel d est impossible puisque la fonction racine carrée croît plus vite que le logarithme ; peut importe le d choisi, elle le rattrapera toujours.

Par les limites On sait que $f(n) \in O(g(n))$ si $\lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} \neq +\infty$:

$$\begin{aligned} \lim_{n \rightarrow +\infty} \frac{(n^3 - 3n^2 + 11)(\log n)^2}{(n^3 \log n)} &= \\ \lim_{n \rightarrow +\infty} \frac{\cancel{(n^3 - 3n^2 + 11)} \overset{n^3}{\nearrow} (\log n)^2}{n^3(\log n)} &= \\ \lim_{n \rightarrow +\infty} \frac{\cancel{n^3} (\log n)^{\cancel{2}}}{\cancel{n^3}(\log n)} &= \\ \lim_{n \rightarrow +\infty} \log n &= +\infty \\ \Rightarrow f(n) &\notin O(g(n)) \end{aligned}$$

On sait que $f(n) \in \Omega(g(n))$ si $\lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} = +\infty$:

$$\begin{aligned}
& \lim_{n \rightarrow +\infty} \frac{(n^3 - 3n^2 + 11)(\log n)^2}{n^4} = \\
& \lim_{n \rightarrow +\infty} \frac{\cancel{(n^3 - 3n^2 + 11)}^{\nearrow n^3} (\log n)^2}{n^4} = \\
& \lim_{n \rightarrow +\infty} \frac{n^3 (\log n)^2}{n^4} = \\
& \lim_{n \rightarrow +\infty} \frac{(\log n)^2}{n} = \\
& \lim_{n \rightarrow +\infty} \frac{\log n}{\sqrt{n}} \stackrel{RH}{=} \\
& \lim_{n \rightarrow +\infty} \frac{(\log n)'}{\sqrt{n}'} = \\
& \lim_{n \rightarrow +\infty} \frac{\frac{1}{x}}{\frac{1}{2x}} = \\
& \lim_{n \rightarrow +\infty} 2 = 2 \\
& \Rightarrow f(n) \notin \Omega(g(n))
\end{aligned}$$

On peut dire que l'intersection est vide :

$$S \cap \Theta(f) = \emptyset \quad (5)$$

Par définition, $\Theta(f) = O(f) \cap \Omega(f)$. Si l'intersection n'était pas vide, cela voudrait dire
ou bien que $O(f) \cap \Omega(n^4) \neq \emptyset$ ce qui est impossible puisque $f \notin \Omega(n^4)$,
ou alors que $\Omega(f) \cap O(n^3 \log n) \neq \emptyset$ ce qui est tout aussi impossible puisque $f \notin O(n^3 \log n)$.