

Représentation de chaînes de caractères sous forme de nombres

Christophe Dronne

25 Octobre, 2023

Préliminaires

On définit b le nombre d'éléments de la base \mathbb{B} , définie avec l'ensemble $\{"a", "b", \dots, "z", "A", "B", \dots, "Z"\}$ pour une quantité quelconque de caractères possibles (ici l'alphabet en minuscule et majuscule pour simplifier).

Ainsi, pour donner un exemple plus concret, le nombre $(2)_{10}$ pourrait être représenté sous forme \mathbb{B} tel: $(2)_b$ ou, d'une meilleure façon, $"b"_{\mathbb{B}}$. Pour rendre le tout plus compréhensible, on adoptera cette dernière notation.

Conversement, la chaîne $"bonjour"_{\mathbb{B}}$ pourrait être représentée sous forme décimale, tel: $(50713653423)_{10}$

Le but du tout est de trouver les avantages/désavantages d'une représentation sous forme de nombre de chaînes de caractères en informatique. On prendra particulièrement soin de toujours indiquer la complexité des opérations, à usage exclusif du langage de programmation Python (3.11).

Longueur d'un nombre

On essaie de trouver un moyen efficace d'exprimer la longueur d'un nombre en une base quelconque (la longueur étant la quantité de chiffres dans le nombre pour une base donnée).

On note la longueur de x_b " $\text{len}(x)$ ". Exemple: $\text{len}(1234_{10}) = 4$, car en base 10, le nombre 1234 contient les chiffres $\{1; 2; 3; 4\}$, et $\text{card}(\{1; 2; 3; 4\}) = 4$.

On pose:

$\forall b \in \mathbb{N} - \{0; 1\}$ représentant la base b de symboles l'ensemble \mathbb{B}

$\forall x_b \in \mathbb{N}^*$ représentant le nombre x sous la base b

$\forall n \in \mathbb{N}^*$, tel que $n = \text{len}(x_b) - 1$

$x = \varepsilon \times b^n$

On sait que:

- $1 \leq \varepsilon < b$, car n est l'exposant de l'ordre de grandeur de x_b ,
et $1 \leq \frac{x_b}{b^n} \leq b$
- $1; \varepsilon; b$ sont positifs d'après leur définition.

$$\begin{aligned} 1 &\leq \varepsilon < b \\ \Rightarrow \log_b(1) &\leq \log_b(\varepsilon) < \log_b(b) \\ \Rightarrow 0 &\leq \log_b(\varepsilon) < 1 \\ \Rightarrow E(\log_b(\varepsilon)) &= 0 \end{aligned}$$

Continuant:

$$\begin{aligned} x &= \varepsilon \times b^n \\ \Rightarrow \log_b(x) &= \log_b(\varepsilon \times b^n) \\ \Rightarrow \log_b(x) &= \log_b(\varepsilon) + \log_b(b^n) \\ \Rightarrow \log_b(x) &= \log_b(\varepsilon) + n \\ \Rightarrow E(\log_b(x)) &= E(\log_b(\varepsilon) + n) \\ \text{Or, } n &\text{ est un entier, donc: } n = E(n) \\ \Rightarrow E(\log_b(x)) &= E(\log_b(\varepsilon)) + n \\ \text{Comme démontré auparavant, on a: } E(\log_b(\varepsilon)) &= 0, \text{ donc:} \\ \Rightarrow E(\log_b(x)) &= n \end{aligned}$$

Donc, pour calculer la longueur d'une chaîne de caractères composé d'un nombre représenté en n'importe quelle base, on a:

$$\text{len}(x) = E(\log_b(x)) \tag{1}$$

Indice d'un nombre

On essaie de simuler la fonctionnalité d'indices présents dans les chaînes de caractères en Python.

$\forall b \in \mathbb{N} - \{0; 1\}$ représentant le nombre d'éléments dans \mathbb{B}

$\forall x \in \mathbb{N}^*$ représenté sous forme \mathbb{B}

$\forall n \in \mathbb{N}$, tel que $n <$ la quantité de chiffres dans $x_{\mathbb{B}}$

On note $x_{\mathbb{B}}[n]$ le chiffre dans x sous forme \mathbb{B} d'indice n partant de la droite.

Ainsi, pour $x = \text{"bonjour"}_{\mathbb{B}}$, on a: $x[0] = \text{"r"}_{\mathbb{B}}$ (0 étant la place des unités) et $x[2] = \text{"o"}_{\mathbb{B}}$.

De plus, on note $E(x)$ la fonction de la partie entière, tel que $E(3, 14) = 3$.

On pose:

$$x_{\mathbb{B}}[n] = E\left(\frac{x - E\left(\frac{x}{b^{n+1}}\right) \times b^{n+1}}{b^n}\right) \quad (2)$$

Prenons $x = 4321$ et $b = 10$ (donc le système décimal).

On essaie d'extraire tous les chiffres dans $(x)_b$.

$$x[3] = 4 \Leftrightarrow x[3] = E\left(\frac{4321}{1000}\right) \Leftrightarrow x[3] = E\left(\frac{4321}{10^3}\right) \quad (3)$$

On remarque que $b = 10$ et $n = 3$ pour $x[3]$, donc $10^3 = b^n$. On note:

$$x[3] = E\left(\frac{4321}{b^n}\right) = E\left(\frac{x - 0}{b^n}\right) \quad (4)$$

On continue pour $n = 2$:

$$x[2] = 3 \quad (5)$$

Or, $43 - 40 = 3$, soit $\frac{4321-4000}{100} = 3, 21$, donc:

$$x[2] = E\left(\frac{4321 - 4000}{100}\right) = 3 \quad (6)$$

À nouveau, on constate qu'on a: $n = 2$

$b = 10$

$100 = 10^2 = b^n$

et

$$4000 = E\left(\frac{4321}{1000}\right) \times 1000 = E\left(\frac{x}{b^{n+1}}\right) \times b^{n+1} \quad (7)$$

Finalement, on peut noter $x[2]$, tel:

$$x[2] = E \left(\frac{x - E \left(\frac{x}{b^{n+1}} \right) \times b^{n+1}}{b^n} \right) \quad (8)$$

On retrouve la formule posée, et on remarque que pour $n = 3$ (premier cas, voir haut de la page), on a:

$$E \left(\frac{x}{b^{n+1}} \right) \times b^{n+1} = E \left(\frac{4321}{10000} \right) \times 10000 = 0 \quad (9)$$

Donc la formule peut aussi être utilisée pour $x[3]$.

Indice d'un nombre en Python

Afin de uniformiser la notation d'indice de chaîne de caractères avec celle de l'informatique, on pose:

$\forall x \in \mathbb{N}^*$ en base \mathbb{B}

$i \in \mathbb{N}$, l'indice d'un chiffre dans $x_{\mathbb{B}}$ partant de la gauche (et non la droite comme pour n)

$l \in \mathbb{N}^*$ la quantité de chiffres dans $x_{\mathbb{B}}$.

Ainsi, on a:

$i = l - n$; ou, pour adopter une notation de Python: $i = \text{len}(x) - n$