

Représentation des nombres et conséquences

Yves Josse

Lycée Chateaubriand – PCSI 3

Année 2024-2025

Représentation des entiers naturels

Représentation binaire des entiers naturels

On appelle écriture ou représentation binaire d'un entier naturel, son écriture en base 2. Par exemple $(11)_{10} = (1011)_2$.

Algorithme d'écriture binaire d'un entier naturel

Données : n

$i \leftarrow 0$

tant que $n \neq 0$ **faire**

$a_i \leftarrow$ reste de la division euclidienne de n par 2

$n \leftarrow$ quotient de la division euclidienne de n par 2

$i \leftarrow i + 1$

Résultat : la représentation est $a_i \dots a_1 a_0$

Exemples

- Convertir en base 2, 63 et 145.
- Convertir 101101 en décimal.

Remarques

- Tous les nombres entiers naturels admettent une écriture binaire finie.
- python : conversion en binaire `bin(11)`, écriture en binaire : `0b1011`.

Cas des entiers relatifs

Codage d'un entier relatif en base 2

- Première méthode : un bit de signe suivie de la représentation de la valeur absolue. Par exemple $(0110)_2 = 6$ et $(1110)_2 = -6$. Problème le zéro est codé deux fois nécessitant deux tests pour la comparaison à 0.
- Complément à 2 : on représente sur N bits tous les entiers allant de : -2^{N-1} à $2^{N-1} - 1$ (Complément à 1+1).
 - un entier positif x dans $\llbracket 0, 2^{N-1} - 1 \rrbracket$ est représenté par son écriture binaire sur N bits. Exemple : $5 = (0101)_2$
 - un entier négatif x dans $\llbracket -2^{N-1}, -1 \rrbracket$ est représentée par l'écriture binaire de $x + 2^N$. Exemple : $-6 = (1010)_2$

Représentation des entiers en mémoire

- Sous python, un entier relatif est codé en complément à 2 sur 32 ou 64 bits (de -2^{63} à $2^{63} - 1$) selon le type d'ordinateur (32 ou 64 bits).
- Pour un nombre plus grand, le typage dynamique permet à python de modifier le type et de passer à un format long (limité par la capacité de stockage de l'ordinateur).

Principe de la représentation des nombres réels en mémoire

norme IEEE 754 : $sm2^n$, nombres à virgule flottante (cas 64 bits)

- s est le signe : (0 correspond à +) (1bit)
- l'exposant n est un entier relatif codé sur 11 bits (compris entre -1022 et 1023) représenté par $n + 1023$ (0 et 2047 pour coder $\pm\infty$ ou NaN)
- La mantisse m codé sur 52 bits est un nombre binaire compris entre 1 inclus et 2 exclu.

Si on note s le bit de signe, $e_1 \dots e_{11}$ les bits d'exposant et $m_1 \dots m_{52}$ les bits de la mantisse, le nombre s'écrit :

$$(-1)^s \times 2^{e_1 \dots e_{11} - 1023} \times \left(1 + \sum_{i=1}^{52} m_i \frac{1}{2^i} \right)$$

Exemples sur 8 bits (s : 1bit, n : 2bits, m : 5bits)

(01010100) : signe +, exposant : $2-1=1$, mantisse $1/2^1 + 1/2^3$: 1.625 soit 3.25
 $2,75 = 2 \times 1.375$: signe 0, exposant 1, mantisse $1.375=1+0.25+0.125$ soit
0 10 01100

Conséquences de la représentation limitée des nombres réels

Conséquences

- dépassement de capacités : 64 bits peuvent ne pas suffire ($2^{1023} = 9 \times 10^{307}$)
- erreurs d'arrondis : un nombre réel peut ne pas être représenté de façon exacte (pas de test de comparaison possible). Précision : $2^{-52} = 2 \times 10^{-16}$ (16 chiffres significatifs en décimal)
- propagation des erreurs d'arrondis (multiplication de deux nombres...)
- comparaison à zéro $x == 0$ à remplacer par $x \leq \varepsilon$ où ε est l'erreur d'arrondi (risque de boucle infini).

Bugs célèbres liés à des erreurs d'arrondis

- Missile Patriot (intercepteur de missiles) (février 1991) : accumulation d'arrondis sur un temps très long : 28 soldats tués.
- Processeur d'intel (1994) : erreur lors de divisions...
- Explosion d'Ariane 5 (juin 1996 : 500 millions de \$) : Conversion d'un nombre réel en un entier relatif codé sur 16 bits (overflow...).

Écriture binaire

On souhaite convertir un nombre entier en une chaîne de caractères qui correspond à son écriture en binaire.

- 1 Écrire une fonction `binaire(n)` qui prend en entrée un entier positif n et qui retourne une chaîne de caractères qui représente l'écriture binaire de n .
- 2 Écrire une fonction `decimal(c)` qui fait l'opération inverse ; elle prend en paramètre une chaîne de caractères représentant l'écriture binaire d'un entier positif, et retourne l'entier représenté.
- 3 Adapter les mêmes fonctions pour l'écriture binaire signé d'un entier relatif sur N bits.

Nombre de uns

Écrire une fonction `NombreDeUns` qui prend en argument un entier naturel n et qui renvoie le nombre de 1 dans la décomposition en base 2 de n . Par exemple `NombreDeUns(23)` renvoie 4.