Didactique de l'informatique

M1 MEEF NSI

Séance 1 : notions générales de didactique

Emmanuel Beffara Benjamin Wack

Institut national supérieur du professorat et de l'éducation
Académie de Grenoble

2023 – 2024

Un exemple pour commencer

Vous devez démarrer le cours de représentation binaire des nombres en classe de 1ère :

- Quelles sont les questions que vous vous posez ?
- Comment vous y prenez-vous ?
- Comment commence votre cours ?

(Réfléchissez-y 5 min, puis chacun donne un élément de réponse.)

Trois documents

Que pensez-vous de ces façons de présenter la numération binaire ?

(Réfléchissez-y 5 min, puis chacun donne un élément de réponse.)

Manuel Spécialité ISN en Tle S, Eyrolles, 2012

2. Comment peut-on décrire les motifs suivants, répétés indéfiniment : 0000100000 et 0101010000 ?
3. Comment faut-il procéder pour qu'un motif donné ne s'affiche que toutes les dix secondes ?
4. Dans une situation où il y aurait deux types de panne en même temps, comment pourrait-on procéder pour afficher les motifs correspondant aux deux messages d'erreurs différents ?

La représentation des entiers naturels

Depuis le Moyen Âge, on écrit les nombres entiers naturels en *notation décimale à position*. Cela signifie que, pour écrire le nombre entier naturel n, on commence par imaginer n objets, que l'on groupe par paquets de dix, puis on groupe ces paquets de dix objets en paquets de dix paquets, etc. À la fin, il reste entre zéro et neuf objets isolés, entre zéro et neuf paquets isolés de dix objets, entre zéro et neuf paquets isolés de cent, etc. Et on écrit cet entier naturel en notant, de droite à gauche, le nombre d'objets isolés, le nombre de paquets de dix, le nombre de paquets de cent, le nombre de paquets de mille, etc. Chacun de ces nombres étant compris entre zéro et neuf, seuls dix chiffres sont nécessaires : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9. Par exemple, l'écriture 2359 exprime un entier naturel formé de 9 unités, 5 dizaines, 3 centaines et 2 milliers.

Le choix de faire des paquets de dix est peut-être dû au fait que l'on a dix doigts, mais on aurait pu tout aussi bien décider de faire des paquets de deux, de cinq, de douze, de vingt, de soixante, etc. On écrirait alors les nombres entiers naturels en notation à position en base deux, cinq, douze, vingt ou soixante. La notation décimale à position s'appelle donc aussi la notation à position en base dix.

En *notation binaire*, c'est-à-dire en notation à position en base deux, le nombre treize s'écrit $\underline{1101}$: de droite à gauche, 1 unité, 0 deuzaine, 1 quatraine et 1 huitaine. L'écriture d'un entier naturel en binaire est en moyenne 3,2 fois plus longue que son écriture en base dix, mais elle ne demande d'utiliser que deux chiffres : 0 et 1.

Indiquer la base

Dans ce livre, quand une suite de chiffres exprime un nombre dans une base différente de dix, on indique la base entre parenthèses, par exemple : 1101 (en base deux). On souligne aussi parfois un mot pour indiquer qu'il exprime un nombre en base deux : $\underline{1101}$. Enfin, on rassemble parfois les bits par groupe de quatre ou de huit dans les mots très longs pour qu'ils soient plus faciles à lire : $\underline{1111111101}$ est écrit $\underline{11\ 1111\ 1101}$. Comme en base dix, ces groupes sont formés de droite à gauche.

Exercice 7.5

Un horloger excentrique a eu l'idée de fabriquer une montre sur laquelle l'heure est indiquée par 10 diodes électroluminescentes appelées 1 h, 2 h, 4 h, 8 h, 1 min, 2 min, 4 min, 8 min, 16 min et 32 min. Pour connaître l'heure, il suffit d'ajouter la valeur de toutes les diodes allumées.

Quelle heure est-il quand sont allumées les diodes 1 h, 2 h, 4 h, 1 min, 2 min, 8 min, 16 min et 32 min ? Quelles sont les diodes allumées à 5 h 55 ? Est-il possible de représenter toutes les heures ? Toutes les configurations sont-elles la représentation d'une heure ?

Comme la mémoire des ordinateurs est constituée de circuits qui ne peuvent être chacun que dans deux états, on peut utiliser chaque circuit de la mémoire pour représenter un chiffre binaire, en identifiant l'un de ces états avec le chiffre binaire 0 et l'autre avec le chiffre binaire 1 – on comprend *a posteriori* pourquoi on a choisi d'appeler ces états eux-mêmes 0 et 1. Le nombre 13 = $\underline{1101}$ est donc représenté dans la mémoire d'un ordinateur par le mot 1101, c'est-à-dire par quatre circuits respectivement dans les états 1, 0, 1 et 1.

La base cinq

Pour comprendre comment transformer un entier naturel, écrit en base dix, dans une autre base, on commence par la base cinq, moins particulière que la base deux, sur laquelle on reviendra plus tard.

SAVOIR-FAIRE Trouver la représentation en base cinq d'un entier naturel donné en base dix

Pour écrire les entiers naturels en base cinq, on a besoin de cinq chiffres : 0, 1, 2, 3, 4. Quand on a n objets, on les groupe par paquets de cinq, qu'on groupe ensuite en paquets de cinq paquets, etc. Autrement dit, on fait une succession de divisions par 5, jusqu'à obtenir un quotient égal à 0.

Exercice 7.6 (avec corrigé)

Trouver la représentation en base cinq de 47.

47 objets se regroupent en 9 paquets et 2 unités, puis les 9 paquets se regroupent en 1 paquet de paquets et 4 paquets.

$47 = 9 \times 5 + 2 = (1 \times 5 + 4) \times 5 + 2 = (1 \times 5^2) + (4 \times 5^1) + (2 \times 5^0)$

Donc 47 = 142 (en base cinq).

Manuel NSI 1ère Prépabac, Hatier, 2019

COURS EXERCICES & SUJETS CORRIGÉS

1 Représentations des entiers naturels en binaire

En bref *Nous sommes habitués à utiliser l'écriture en base 10 des entiers. Mais plus généralement, tout entier peut s'écrire dans une autre base, comme la base 2 ou la base 16, couramment utilisées en informatique car plus adaptées à la mise en mémoire des nombres.*

I Écriture d'un entier en base β

- **Définition :** Une **base** d'un **système de numération positionnel** est un entier naturel β supérieur ou égal à deux.

Dire qu'un nombre s'écrit $a_n a_{n-1} a_{n-2} \dots a_1 a_{0\beta}$ en base β signifie qu'il est égal à :

$$a_n \times \beta^n + a_{n-1} \times \beta^{n-1} + a_{n-2} \times \beta^{n-2} + \dots + a_1 \times \beta + a_0$$

où les entiers naturels a_i sont strictement inférieurs à β.

- *Exemple* : 237_{10} est égal à $2 \times 10^2 + 3 \times 10 + 7$ et $237_8 = 2 \times 8^2 + 3 \times 8 + 7 = 159_{10}$.

À NOTER
Dans l'écriture 237_{10}, le 10 en indice indique que le nombre est écrit en base 10.

II Les bases privilégiées en informatique

En informatique, on travaille en base 2 (les bits), en base 16 (adresses mémoire, couleurs HTML) et parfois en base 8 (droits des fichiers UNIX).

1 Écriture en base 16

Pour écrire un nombre en base 16, il faut disposer d'un caractère pour chacun des entiers de 0 à 15. Or, on ne dispose pas d'assez de chiffres pour écrire les 16 chiffres de la base 16. On complète donc les chiffres de 0 à 9 par les six premières lettres de l'alphabet : A, B, C, D, E et F.

Base 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Base 16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11

2 Écriture en base 2

En base 2, tous les nombres sont représentés avec les deux symboles 0 et 1.

Base 10	0	1	2	3	4	5	6	7	8	9	10
Base 2	0	1	10	11	100	101	110	111	1000	1001	1010

- *Exemple* : 1101_2 est égal à $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$.

III Conversions de la base 10 vers la base 2

1 Algorithme de soustraction

Pour convertir un nombre de la base 10 vers la base 2, on retranche du nombre la plus grande puissance de 2 possible.

Exemple :

$58_{10} - 1 \times 2^5 = 58_{10} - 32_{10} = 26_{10}$ 1
$26_{10} - 1 \times 2^4 = 10_{10}$ 1
$10_{10} - 1 \times 2^3 = 2_{10}$ 1
$2_{10} - 0 \times 2^2 = 2_{10}$ 0
$2_{10} - 1 \times 2^1 = 0$ 1
$0_{10} - 0 \times 2^0 = 0$ 0

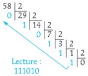
Donc $58_{10} = 111010_2$.

À NOTER
En Python, on utilise des préfixes pour indiquer dans quelle base les nombres sont donnés. `0x` pour la base 16, `0b` pour la base 2 et `0o` pour la base 8.

2 Algorithme de divisions

Pour convertir un nombre de la base 10 vers la base 2, on effectue des divisions successives de ce nombre par 2. En lisant les restes de bas en haut on obtient le résultat.

Exemple : $58_{10} = 111010_2$.



À NOTER
On utilise ce même algorithme de la division pour la conversion de la base 10 en n'importe quelle autre base.

IV Autres conversions

- De la base 2 vers la base 16, on groupe les bits par paquets de 4, quitte à rajouter des 0 à gauche.

Exemple : $10101000011_2 = \underbrace{0101}_{5}\,\underbrace{0100}_{4}\,\underbrace{0011}_{3} = 543_{16}$.

- De la base 16 vers la base 2, on transforme chaque caractère par un groupe de 4 bits.

Exemple : $A3C_{16} = \underbrace{1010}_{A_{16}=10_{10}}\,\underbrace{0011}_{3}\,\underbrace{1100}_{C_{16}=12_{10}} = 101000111100_2$.

- De la base 16 à la base 10, on écrit les caractères en base 10 et on utilise la définition avec les puissances de la base de départ, ici 16.

Exemple :

$$\begin{aligned} A3C_{16} &= 10_{10} \times 16^2 + 3_{10} \times 16^1 + 12_{10} \times 16^0 \\ &= 10_{10} \times 256_{10} + 3_{10} \times 16_{10} + 12_{10} \\ &= 2\,620_{10}. \end{aligned}$$

À NOTER
On obtient par le même algorithme la conversion de n'importe quelle base vers la base 10.

Trouvé sur Internet (Éduscol)

Représentation	Signes utilisés	Un même entier
Binaire	0, 1	1111011
Ternaire	0, 1, 2	11120
Octale	0, 1, 2, 3, 4, 5, 6, 7	173
Décimale	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	123
Hexadécimale	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	7B

2.1. Représentation décimale

Tout nombre entier naturel décimal peut s'écrire comme combinaison linéaire de puissance de 10. Par exemple :

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

De manière générale, pour tout entier naturel non nul [2] n comportant p chiffres, il existe un ensemble de chiffres $\{d_0, d_1, \ldots, d_{p-1}\}$, où chaque $d_i \in \{0, 1, \ldots, 9\}$ pour $i \in \{0, \ldots, p-1\}$ et $d_{p-1} \neq 0$, tels que :

$$n = \sum_{i=0}^{p-1} d_i \times 10^i$$

Cette écriture constitue la *décomposition décimale* de n. La suite $(d_{p-1}, d_{p-2}, \ldots, d_0)$ est sa *représentation décimale*. Dans le reste de l'exposé, cette représentation est notée plus traditionnellement sous la forme $d_{p-1}d_{p-2}\ldots d_0$.[3]

Pour obtenir les chiffres d_i à partir de l'entier naturel n, deux opérations sont utiles.

- ▷ la division entière, // en *Python*, qui à deux entiers a et b associe leur quotient q ;
- ▷ le reste de la division entière, % en *Python*, qui à deux entiers a et b associe le reste r de la division euclidienne : $r = a - b \times q$.

À l'aide de ces opérations, il est possible d'obtenir les chiffres d'un entier naturel à l'aide de l'algorithme suivant.

```
Choisir un entier naturel n.
Initialiser r à n.
Tant que r n'est pas nul, répéter les instructions suivantes.
  Calculer r % 10 et stocker le résultat.
  Remplacer r par r // 10
Renvoyer les chiffres stockés
```

Les chiffres sont les entiers stockés. En *Python*, cet algorithme peut être traduit par le code suivant.

```python
# entier à décomposer
n = 123
# initialisation
r = n
chiffres = []
# détermination des chiffres
while r != 0:
  chiffres.append(r % 10)
    r = r // 10
chiffres.reverse()
```

Dans ce code, les résultats sont stockés dans un tableau initialement vide. À chaque nouveau calcul du reste, le résultat est ajouté en queue de tableau à l'aide de la méthode `append`. Pour avoir les chiffres dans le bon ordre, il faut alors retourner le tableau à l'aide de `reverse`.
Partant d'une liste de chiffres représentée sous forme d'un tableau, un entier est peut être re-construit. Noter l'absence d'utilisation de l'opération d'exponentiation dans le code suivant.

2. Le nombre 0 a une écriture qui diffère un peu, puisqu'elle comporte un unique `0`.
3. La représentation décimale de 0 devrait être une suite de longueur 0. À la place, on a comme convention d'écrire `0`.


```python
# tableau de chiffres
chiffres = [3, 1, 4, 1, 5, 9, 2]
# initialisation
n = 0
# calcul de l'entier
for c in chiffres:
  n = 10 * n + c
```

2.2. Représentation binaire

Tout nombre entier naturel peut aussi être représenté en base 2 à l'aide des seuls signes `0` et `1`. Par exemple, $(\texttt{101})_2$, $(\texttt{111001})_2$, $(\texttt{1111111})_2$. Pour distinguer ces représentations binaires de représentations décimales, un indice 2 est ajouté. Il convient également de ne pas lire ces nombres comme on lirait des nombres décimaux. Ainsi, $(\texttt{101})_2$ n'est pas cent un mais un zéro un.

De manière générale, la *représentation binaire* de tout entier naturel non nul n à m chiffres `0` ou `1` peut se mettre sous la forme $b_{m-1}b_{m-2}\ldots b_0$ où $b_i \in \{\texttt{0}, \texttt{1}\}$ pour $i \in \{0, \ldots, m-1\}$ et $b_{m-1} \neq \texttt{0}$ (en binaire, cette dernière condition impose nécessairement $b_{m-1} = \texttt{1}$). Ainsi, on peut écrire :

$$n = (\texttt{1}b_{m-2}\ldots b_0)_2$$

Convertir un entier du système binaire au système décimal et réciproquement peut aisément se faire.

$$(\texttt{1111011})_2 \quad \leftrightarrow \quad 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \quad = \quad 123$$

Un paragraphe présente plus loin un code réalisant ces opérations.

2.3. Représentation hexadécimale

Comme les représentations décimale et binaire, la *représentation hexadécimale* est un système de numération positionnel en base 16. Tout entier naturel non nul n à p chiffres dans ce système peut s'écrire sous la forme :

$$n = (h_{p-1}\ldots h_0)_{16}$$

avec, pour tout entier $i \in \{0, \ldots, p-1\}$, $h_i \in \{\texttt{0}, \texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}, \texttt{7}, \texttt{8}, \texttt{9}, \texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}, \texttt{E}, \texttt{F}\}$ et $h_{p-1} \neq \texttt{0}$.
Le tableau suivant présente les conversions des premiers entiers naturels entre les trois représentations précédentes.

Décimal	Binaire	Hexadécimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

Décimal	Binaire	Hexadécimal
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

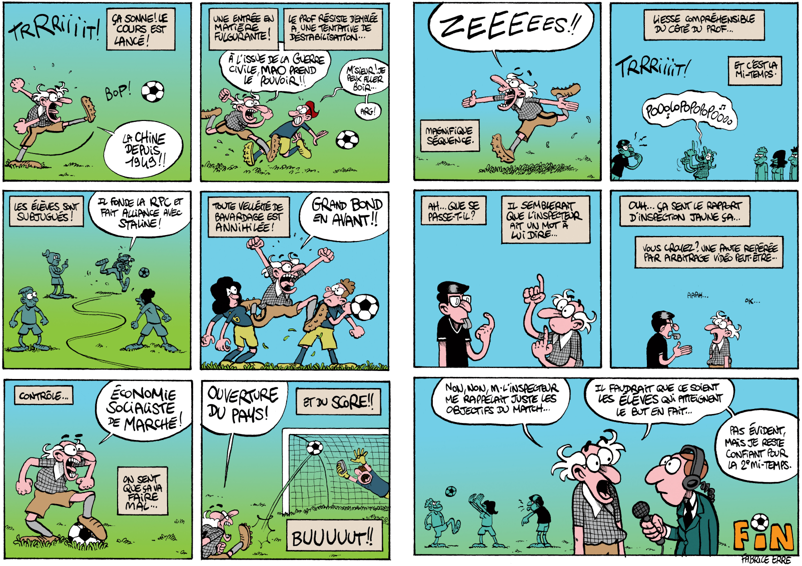
Là encore, convertir un entier du système hexadécimal au système décimal et réciproquement peut aisément se faire.

$$(\texttt{7B})_{16} \quad \leftrightarrow \quad 7 \times 16^1 + 11 \times 16^0 \quad = \quad 123$$

La conversion hexadécimal ↔ binaire peut également se faire rapidement : chaque chiffre hexadécimal équivaut à exactement quatre chiffres binaires.

- ▷ Reprenant l'exemple du nombre `7B`, le tableau précédent indique que `7` peut être converti en `0111` et `B` en `1011`. Ce qui donne la représentation binaire `01111011` ou `1111011` si on omet le premier zéro.
- ▷ À la représentation binaire `1101110` peut être associée deux blocs de quatre chiffres : `0110` et `1110`. Le premier bloc se convertit en `6` et le second en `E` de sorte que la représentation hexadécimale de l'entier est `6E`.

Particulièrement pratique pour représenter de grands entiers sous forme compacte, la représentation hexadécimale est fréquemment utilisée en électronique numérique et dans le monde des réseaux pour décrire des adresses.



Fabrice Erre, *Une année au lycée*

Une autre façon de s'y prendre : tour de magie binaire

(cf par exemple IREM de Clermont-Ferrand)

1	3	5	7	9	11	13	15
17	19	21	23	25	27	29	31

2	3	6	7	10	11	14	15
18	19	22	23	26	27	30	31

4	5	6	7	12	13	14	15
20	21	22	23	28	29	30	31

8	9	10	11	12	13	14	15
24	25	26	27	28	29	30	31

16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

Pensez à un nombre entre 0 et 31 et dites-moi dans quelles tables il apparaît.

Déroulement en classe (ici de 6ème)

« Ça marche à chaque fois ! »

- Donc il y a un truc, pouvez-vous le trouver ?

« Le professeur cherche à toute vitesse le nombre commun »
« Il mémorise tous les nombres inscrits dans les tables »

- Est-ce que ça vous semble plausible ?
- Réalisation du tour les yeux bandés

« Le professeur fait un calcul »

- Oui mais lequel ?
- Peux-tu reproduire le tour toi-même ?
- *(une fois que quelqu'un réussit)* Peux-tu expliquer au reste de la classe comment tu as fait ?

À la fin : mise en commun et prise de notes de cours

Quelques constats

- Connaître $\neq$ savoir utiliser
  (Ex : comptine des nombres VS dénombrement)
- Apprendre demande d'associer du sens aux notions abordées :
  - pour les reconnaître dans une autre situation
  - pour trouver un intérêt à se les approprier

La démarche didactique

L'idée est d'étudier les *conditions* rendant possible l'apprentissage des notions et des techniques.
Précisons un peu :

- L'apprentissage en soi est du ressort de l'élève, le rôle de l'enseignant est de créer les conditions qui lui permettent de le faire.
- Chaque savoir (notion, technique...) apporte une amélioration de la compréhension d'un objet donné.
- On donne du *sens* à une notion en la mettant en œuvre, ce qui nécessite d'en comprendre le contexte et les enjeux.

Ces principes généraux sont à décliner en fonction du domaine à enseigner. La didactique est spécifique à la discipline.

Hypothèses de travail (Brousseau)

- Un élève n'arrive pas vierge de connaissances et de conceptions.
- Il n'apprend que s'il modifie lui-même sa connaissance.

Une situation d'apprentissage « fondamentale » :

- doit pouvoir être communiquée à un apprenant qui n'en connaît pas la méthode de résolution
  (donc en se basant sur ses connaissances préalables)
- mais il doit pouvoir **apprendre** à la résoudre
  (en particulier sa réponse initiale ne doit pas être satisfaisante)
- sans intervention didactique de son professeur (c'est à dire portant sur la connaissance).
  (l'élève doit percevoir de lui-même l'inefficacité de sa stratégie initiale)

Construire une situation

La situation dans laquelle l'apprentissage se fait est construite par l'enseignant, elle fournit à l'élève un *milieu* dans lequel expérimenter :

- un contexte disciplinaire (ce qu'on suppose connu et maîtrisé)
- des règles du jeu établies : ce qu'on a le droit d'utiliser pour les besoins de la situation (*contrat didactique*)
- une possibilité de *validation* : lorsque l'élève essaie des choses, il peut constater par lui-même si ce qu'il fait fonctionne ou pas.

L'intention didactique se trouve idéalement dans la construction de la situation et *pas* dans le comportement de l'enseignant (une hypothèse est validée par interaction avec l'objet d'étude, pas par un argument d'autorité).

Retour sur l'exemple du jour

Appliquons ensemble cette approche dans le cas du cours sur la représentation binaire des nombres :

- analyser les documents précédents,
- analyser le tour de magie binaire.

Tour de magie : limites de cette activité

- Quels écueils peuvent se présenter dans le déroulement ?
- Quelles conceptions erronées les élèves risquent d'en retenir ?
- Quels aspects de l'écriture binaire cette activité ne couvre-t-elle pas ?

Aller plus loin

- Comment sont construites les tables ?
- Par quel nombre débuterait la table suivante ?
- Quel nombre maximal permettrait-elle de deviner ?
- Combien de cases sur chaque table ?
- Comment les construire ?