

DERNIÈRE SÉANCE

Vendredi 5 Juin

Option Informatique
Ecole Alsacienne



PLAN

1. Correction du devoir
2. Bilan de l'année
3. Quelques énigmes
4. Quelques liens pour les vacances

CORRECTION DU DEVOIR

QUESTION DE COURS – QUESTION 1



- Une boucle `while` dépend avant tout d'une **condition**, c'est-à-dire une question fermée (à laquelle on répond par oui ou par non).
- Au début de chaque passage dans la boucle, on vérifie que cette condition est vérifiée :
 - Si c'est le cas, on exécute le corps de boucle et on revient tester la condition
 - Sinon on passe à la suite du programme
- Syntaxe

```
while (cond):  
    instr1  
    instr2  
    ...  
    instrN
```
-  En Python, l'indentation, c'est très important ! 

QUESTION DE COURS – QUESTION 2

- Question 2. Que renvoie le code suivant ?

```
if (3 > 2) :  
    print ("A")  
else:  
    print ("B")  
print ("B2")
```

- Rien !
- Par contre, il affiche
A
B2
-  En Python, l'indentation, c'est très important ! 

QUESTION DE COURS – QUESTIONS 3 ET 4

- **Question 3.** En Python, il n'y a pas de différence entre tableaux et listes : dans les deux cas, on utilise le type `list`.
- **Question 4.** Que renvoie le code suivant ?

```
v = [ 2 , 4 , 6 ]  
print(v[3])
```

Traceback (most recent call last):

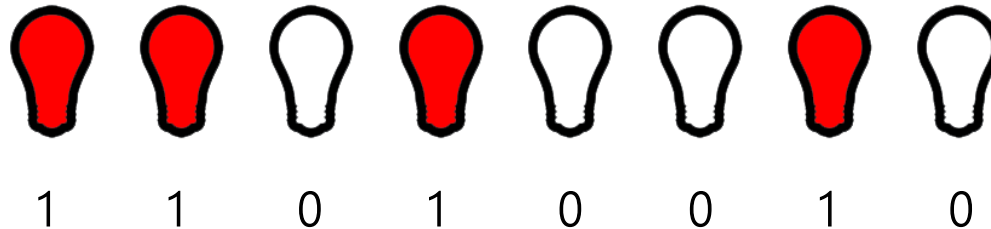
```
File "C:\...\test.py", line 2, in <module>
```

```
    print(v[3])
```

IndexError: list index out of range

QUESTION DE COURS – QUESTION 5

- *Comment représenter un nombre entier positif ou nul avec des 0 et des 1?*
- On utilise son écriture en base 2.
- Exemple : $210 = \underline{11010010}_2$



- Avec 8-bits, le plus grand nombre est donc $\underline{11111111}_2 = 255$

QUESTION DE COURS – QUESTION 6

- **ASCII** : American Standard Code for Information Interchange

- 128 caractères :

- 26 majuscules
- 26 minuscules
- 10 chiffres
- 32 symboles ! « # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ ` { | } ~
- 1 signe d'espace
- 33 symboles de mise en page ou de contrôle

- 8 bits par caractère

- 7 auraient suffi
- Le bit de point fort vaut toujours 0

Correction du devoir

PDF: fr en v · d · m	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
002	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
003	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
006	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

QUESTION DE COURS – QUESTION 7

- Un algorithme de compression a pour objectif de réduire la taille d'un message. On parle de **compression sans perte** si l'intégralité de l'information est conservée.
- Même avec perte, les différences sont parfois invisibles à l'œil nu



36 Ko



13 Ko



3 Ko

- Les seuils varient en fonction des utilisations
 - Streaming vidéo : tolérance élevée
 - Enregistrement d'une image : tolérance faible
 - Sécurité d'un avion : tolérance zéro

QUESTION DE COURS – QUESTION 8

- Un algorithme de détection des erreurs permet de déterminer automatiquement si un message a été altéré (usure, distance, problème réseau, etc.)
- La méthode du bit de contrôle consiste à ajouter un "bit de contrôle" pour chaque paquet de X bits. Ce bit de contrôle indique la parité du nombre de 1 dans le paquet :
 - S'il y avait un nombre pair de 1, le bit de contrôle vaut 0
 - S'il y avait un nombre impair de 1, le bit de contrôle vaut 1

01001000 01**1**100101 0110**1**1100 011011**0**00 01101111**0**

- Autre méthode possible : ajout de redondance

000111000**1**00111000000000

QUESTION DE COURS – QUESTION 9

- Le **chiffre de César** est basé sur un principe assez simple :
*Chaque lettre est remplacée par celle située
trois lettres plus loin dans l'alphabet*

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	X	Y	Z
D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	X	Y	Z	A	B	C

- Nom officiel : **chiffrement par décalage**
- DOHD MDFWD HVW → ALEA JACTA EST
 - "Le sort en est jeté" ou "Les dés sont jetés"
 - Jules César, 10 janvier 49 av. JC, avant de franchir le Rubicon

QUESTION DE COURS – QUESTION 10

- **Problème** : Alice veut écrire à Bob, mais il ne leur est pas possible de se mettre d'accord sur une clé secrète



- Principe :
 1. Bob envoie un cadenas ouvert à Alice, mais il garde la clé
 2. Alice met son message dans une boîte, et ferme la boîte avec le cadenas
 3. Alice envoie la boîte cadenassée à Bob
 4. Comme Bob est le seul à avoir la clé du cadenas, il est le seul qui peut ouvrir la boîte

Clé publique



Clé privée



QUESTION DE COURS – QUESTION 11

- La **complexité en temps** d'un programme est
 - Une estimation du temps qu'il met à s'exécuter
 - En fonction de la taille des arguments passés en entrée
 - A une constante près
- La **complexité en mémoire** d'un programme est
 - Une estimation de l'espace dont il a besoin pour s'exécuter
 - En fonction de la taille des arguments passés en entrée
 - A une constante près
- **Remarque** : On se concentrera dans un premier temps sur la complexité en temps.

EXERCICES – QUESTIONS 12 ET 13

- Question 12.

```
def afficherCarresBoucle(n):  
    for i in range(n, 0, -1):  
        print(i*i)
```

- Question 13.

```
def afficherCarresRecurusif(n):  
    if(n > 0):  
        print(n*n)  
        afficherCarresRecurusif(n-1)
```

EXERCICES – QUESTION 14

- Question 14.

```
def nombreImpairs(t):  
    compteur = 0  
    for i in range(len(t)):  
        if((t[i] % 2) == 1):  
            compteur = compteur + 1  
    return compteur
```

EXERCICES – QUESTION 15

- Question 15.

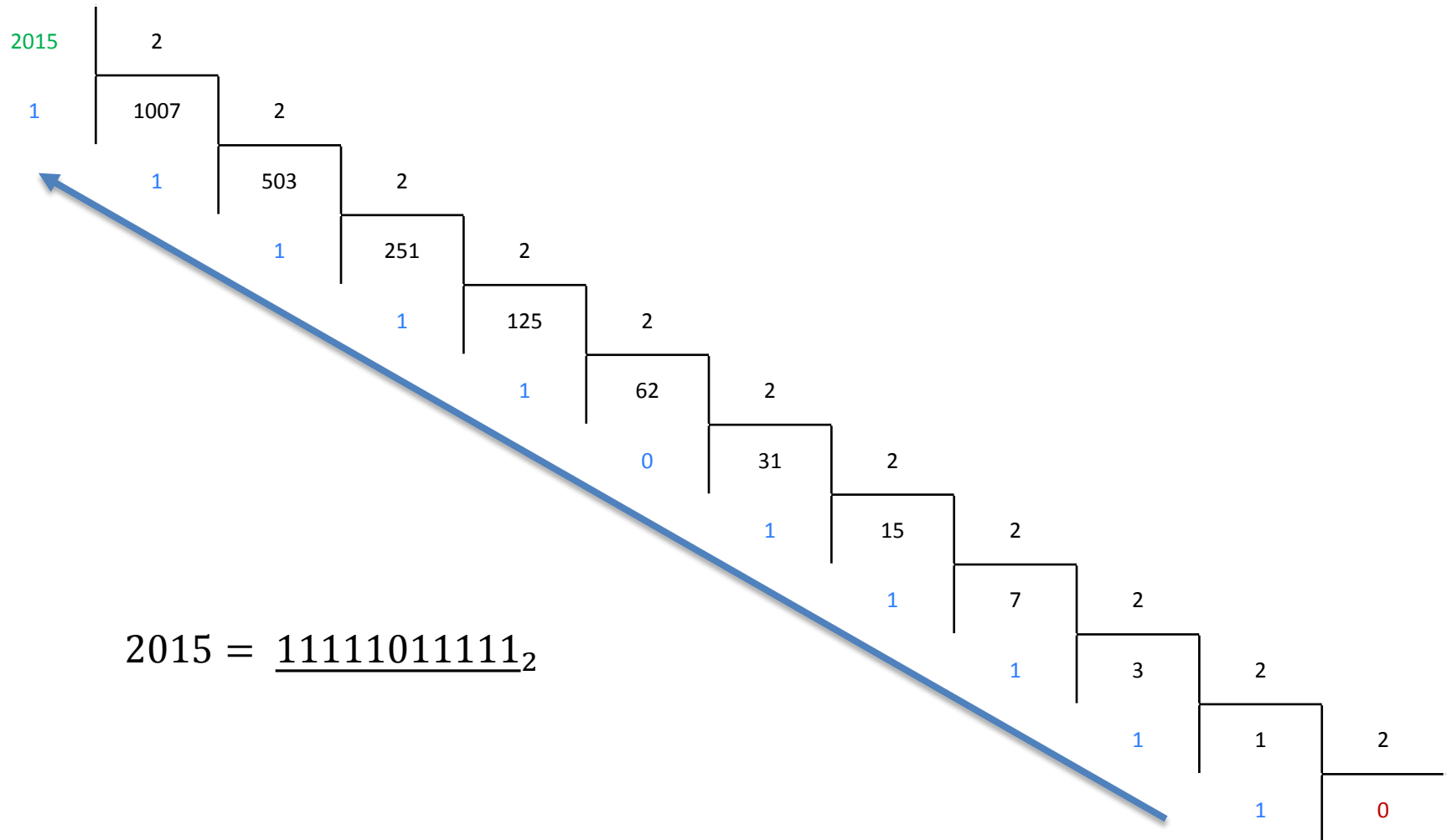
$$\underline{11011111101}_2 = 1 \times 2^{10} + 1 \times 2^9 + 0 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$\underline{11011111101}_2 = 1024 + 512 + 128 + 64 + 32 + 16 + 8 + 4 + 1$$

$$\underline{11011111101}_2 = 1789$$

EXERCICES – QUESTION 16

- Question 16.



EXERCICES – QUESTION 17

- Question 17.

```
def racineCarreeEntiere(n):  
    nombre = 0
```



```
    while((nombre*nombre) < n):  
        nombre = nombre + 1  
    return nombre
```

```
def racineCarreeEntiere(n):  
    nombre = 0
```



```
    while((nombre*nombre) < n):  
        nombre = nombre + 1  
    return nombre - 1
```

```
def racineCarreeEntiere(n):  
    nombre = 0
```



```
    while(((nombre+1)*(nombre+1)) <= n):  
        nombre = nombre + 1  
    return nombre
```

EXERCICES – QUESTIONS 18, 19 ET 20

- Question 18.

```
def estDiviseur(a,b):  
    return ((b%a) == 0)
```

- Question 19.

```
def listeDiviseurs(a):  
    diviseurs = []  
    for i in range(1, a+1):  
        if(estDiviseur(i,a)):  
            diviseurs.append(i)  
    return diviseurs
```

- Question 20.

```
def estPremier(a):  
    return (len(listeDiviseurs(a))==2)
```

EXERCICES – QUESTION 21

- Question 21.

```
def valeurMediane(t):  
  
    for i in range(len(t)):  
  
        inferieurs = 0  
        superieurs = 0  
  
        for j in range(len(t)):  
            if (t[j]>t[i]):  
                superieurs = superieurs + 1  
            else:  
                inferieurs = inferieurs + 1  
  
        if(inferieurs == superieurs):  
            return t[i]  
  
    raise Exception("Mediane non trouvee !")
```

EXERCICES – QUESTION 22

- Question 22.

```
def trouverIndiceDuMaximum(t, iMax):  
    pos = 0  
    val = t[0]  
    for i in range(1, iMax+1):  
        if(t[i] > val):  
            pos = i  
            val = t[i]  
    return pos
```

```
def echanger(t, i, j):  
    x = t[i]  
    t[i] = t[j]  
    t[j] = x
```

```
def triParSelection(t):  
    tCopie = list(t)  
    for i in range(len(tCopie)-1, -1, -1):  
        pos = trouverIndiceDuMaximum(tCopie, i)  
        echanger(tCopie, pos, i)  
    return tCopie
```

EXERCICES – QUESTION 23

- Question 23.

```
def estCarreMagique(g):  
  
    nombreMagique = 0  
    for colonne in range(len(g)):  
        nombreMagique = nombreMagique + g[0][colonne]  
  
    for ligne in range(1, len(g)):  
        sommeLigne = 0  
        for colonne in range(len(g)):  
            sommeLigne = sommeLigne + g[ligne][colonne]  
        if(sommeLigne != nombreMagique):  
            return False  
  
    for colonne in range(len(g)):  
        sommeColonne = 0  
        for ligne in range(len(g)):  
            sommeColonne = sommeColonne + g[ligne][colonne]  
        if(sommeColonne != nombreMagique):  
            return False  
  
    sommeDiagonale1 = 0  
    sommeDiagonale2 = 0  
    for i in range(len(g)):  
        sommeDiagonale1 = sommeDiagonale1 + g[i][i]  
        sommeDiagonale2 = sommeDiagonale2 + g[i][len(g)-1-i]  
    if((sommeDiagonale1 != nombreMagique) or (sommeDiagonale2 != nombreMagique)):  
        return False  
  
    return True
```

EXERCICES – REMARQUES GÉNÉRALES

- Inutile de tester la valeur d'un booléen pour le renvoyer :

```
if ((b % a) == 0) :  
    return True  
else  
    return False
```

s'écrit plus simplement

```
return ((b % a) == 0)
```

EXERCICES – REMARQUES GÉNÉRALES

- De la même façon, si une fonction renvoie un booléen, inutile de vérifier si son résultat est `True` :

```
if (estDiviseur(a,b) == True) :
```

s'écrit plus simplement

```
if (estDiviseur(a,b)) :
```

EXERCICES – REMARQUES GÉNÉRALES

- Ne vous compliquez pas la vie

```
def afficherCarresRecurusif(n):  
    acc = n  
    if acc == 1:  
        print(1)  
    else:  
        print(acc*acc)  
        afficherCarresRecurusif(n)
```

EXERCICES – REMARQUES GÉNÉRALES

- Attention aux divisions par 0
 - Il est très rare de faire explicitement une division par 0 :

```
if (3 % 0) == 0:  
    [...]
```

- Par contre, il arrive souvent qu'on se ramène dans ce cas-là sans s'en rendre compte :

```
for i in range(a+1):  
    if (a % i) == 0:  
        [...]
```

RÉSULTATS

- Barème :
 - 1 points par question
 - Note de base sur 23, ramenée sur 20
- Résultats :
 - Moyenne : 13,6
 - Les notes vont de 5 à 18,7
- Ajustements pour la moyenne trimestrielle :
 - TD rendus
 - Attitude et sérieux

BILAN DE L'ANNÉE

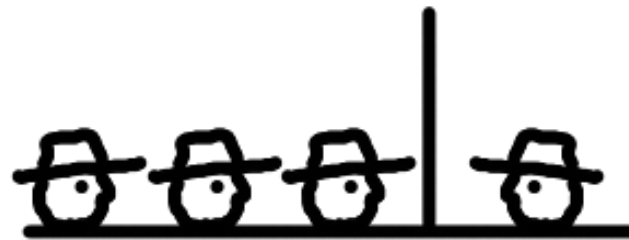
BILAN DE L'ANNÉE

- Merci
- Bravo
- N'hésitez pas à me contacter pendant les vacances
- Votre avis compte

QUELQUES ENIGMES

LES 4 PRISONNIERS ET LEURS CHAPEAUX

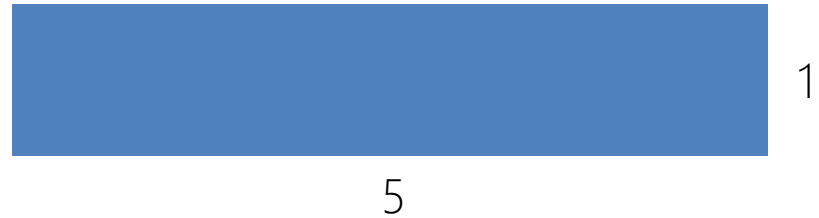
- Une situation inconfortable
 - Quatre prisonniers sont enterrés jusqu'au cou
 - Chacun a sur la tête un chapeau dont il ne peut voir la couleur
 - Il y a 2 chapeaux noirs et 2 chapeaux blancs
 - Ils ne peuvent pas parler entre eux
 - Ils ne peuvent pas tourner la tête
 - Ils sont situés de part et d'autre d'un mur opaque :



- Dilemme
 - Un des prisonniers doit prendre la parole et annoncer la couleur du chapeau qu'il porte
 - S'il dit la vérité, ils sont tous libérés
 - S'il se trompe, ils sont tous exécutés (ils n'ont droit qu'à un seul essai)

COUPE AU CARRÉ

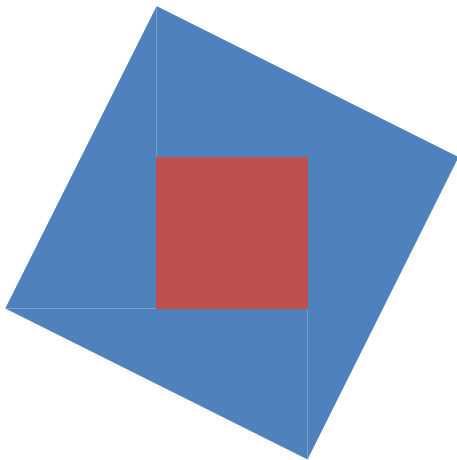
Peut-on découper le rectangle suivant et recoller les morceaux pour obtenir un carré ?



- Indice 1 :



- Indice 2 :



LA PESÉE UNIQUE

- Vous avez devant vous 10 sacs remplis de pièces :
 - 9 sacs sont remplis de vraies pièces, qui pèsent 1 g chacune
 - 1 sac est rempli de fausses pièces, qui pèsent 2g chacune
- Question : Comment déterminer quel est le sac qui contient les fausses pièces ?
- Contrainte :

Vous disposez d'une balance, et vous pouvez prendre autant de pièces que vous le souhaitez dans chaque sac, mais vous n'avez le droit qu'à une seule pesée



LE CARRÉ ET L'ANNEAU

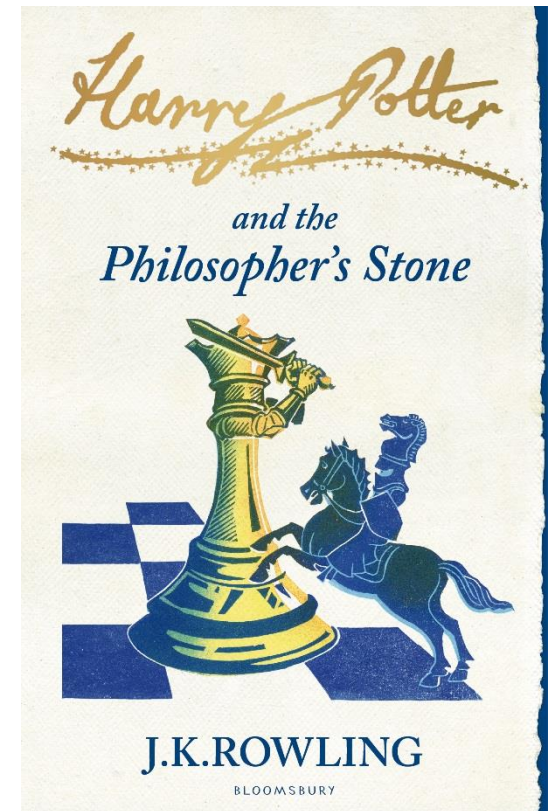
- Peut-on découper le carré suivant en 4 morceaux identiques capables de passer dans l'anneau ci-dessous ?



- Indice 1 : $\sqrt{2} \approx 1,41$
- Indice 2 : $\sqrt{2} \times 3 \approx 4,24$

UN PETIT PEU DE LECTURE

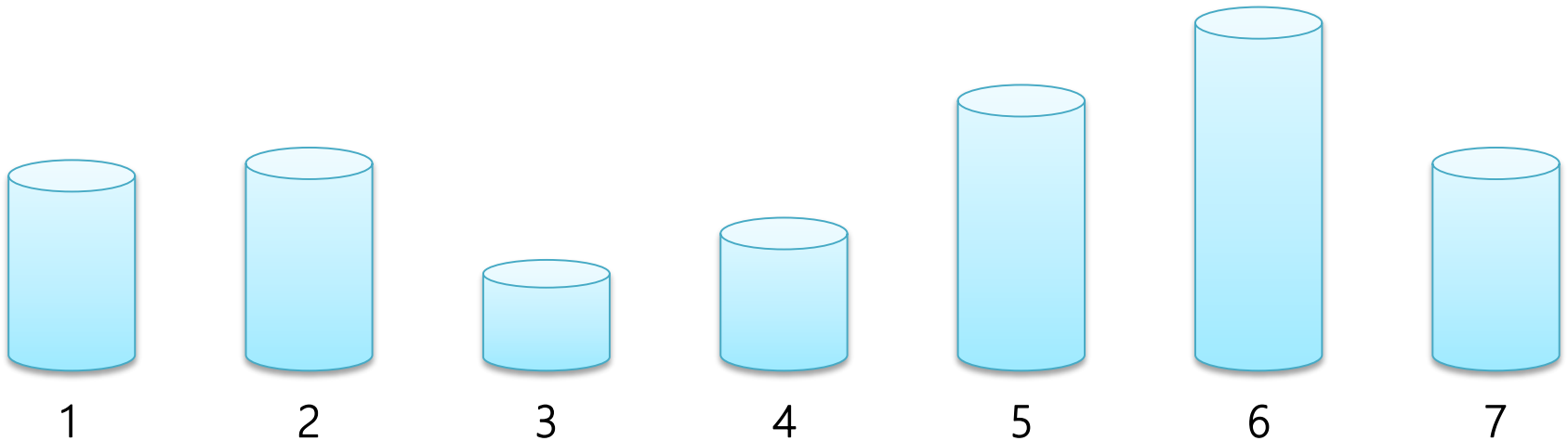
*Devant est le danger, le salut est derrière.
Deux sauront parmi nous conduire à la lumière,
L'une d'entre les sept en avant te protège
Et une autre en arrière abolira le piège,
Deux ne pourront t'offrir que simple vin d'ortie
Trois sont mortels poisons, promesse d'agonie,
Choisis, si tu veux fuir un éternel supplice,
Pour t'aider dans ce choix, tu auras quatre indices.
Le premier : si rusée que soit leur perfidie,
Les poisons sont à gauche des deux vins d'ortie.
Le second : différente à chaque extrémité,
Si tu vas de l'avant, nulle n'est ton alliée.
Le troisième : elles sont de tailles inégales,
Ni naine ni géante en son sein n'est fatale.
Quatre enfin : les deuxièmes, à gauche comme à droite,
Sont jumelles de goût, mais d'aspect disparates.*



J.K. Rowling, *Harry Potter à l'école des sorciers*

L'ÉNIGME DES POTIONS

- Situation :
 - Des flammes noires empêchent d'avancer
 - Des flammes violettes empêchent de reculer
 - Sept potions sur une table



- But : Identifier les potions qui permettent de quitter la pièce

L'ÉNIGME DES POTIONS

- Sept potions

L'une d'entre les sept en avant te protège

Et une autre en arrière abolira le piège,

Deux ne pourront t'offrir que simple vin d'ortie

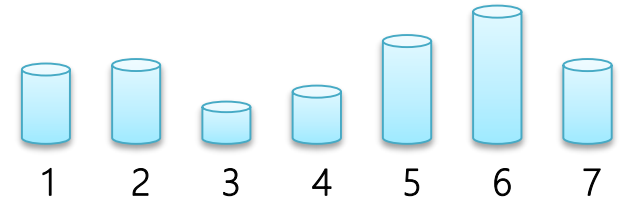
Trois sont mortels poisons, promesse d'agonie,

- Quatre types de potions

- Un potion qui permet d'aller vers l'avant
- Un potion qui permet de retourner en arrière
- Deux vins d'orties (inoffensifs mais inutiles)
- Trois poisons mortels

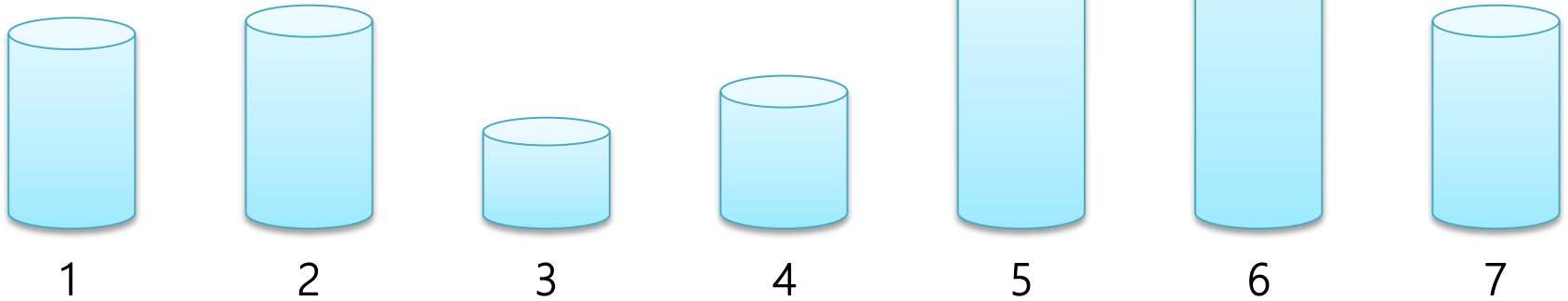
L'ÉNIGME DES POTIONS

- Quatre indices :
 - Le premier : si rusée que soit leur perfidie,
Les poisons sont à gauche des deux vins d'ortie.*
 - Le second : différente à chaque extrémité,
Si tu vas de l'avant, nulle n'est ton alliée.*
 - Le troisième : elles sont de tailles inégales,
Ni naine ni géante en son sein n'est fatale.*
 - Quatre enfin : les deuxièmes, à gauche comme à droite,
Sont jumelles de goût, mais d'aspect disparates.*

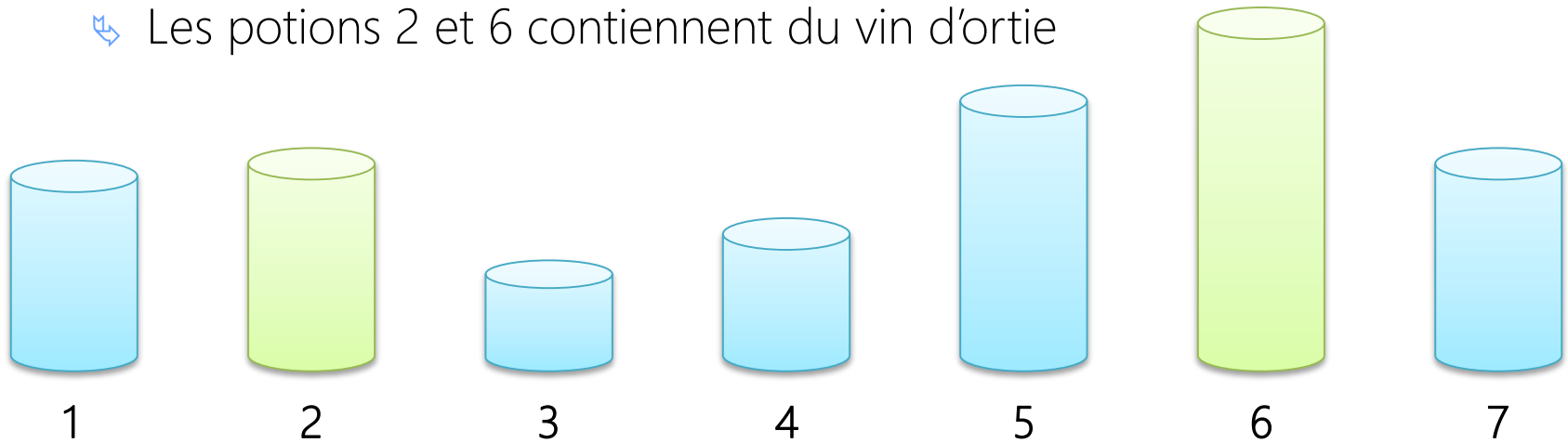


- Version moins alambiquée
 1. A gauche de chaque vin d'ortie se trouve un poison
 2. La potion 1 et la potion 7 sont différentes et ne permettent d'avancer
 3. La plus grande (6) et la plus petite potion (3) ne sont pas des poisons
 4. Les potions 2 et 6 n'ont pas la même taille mais ont le même contenu

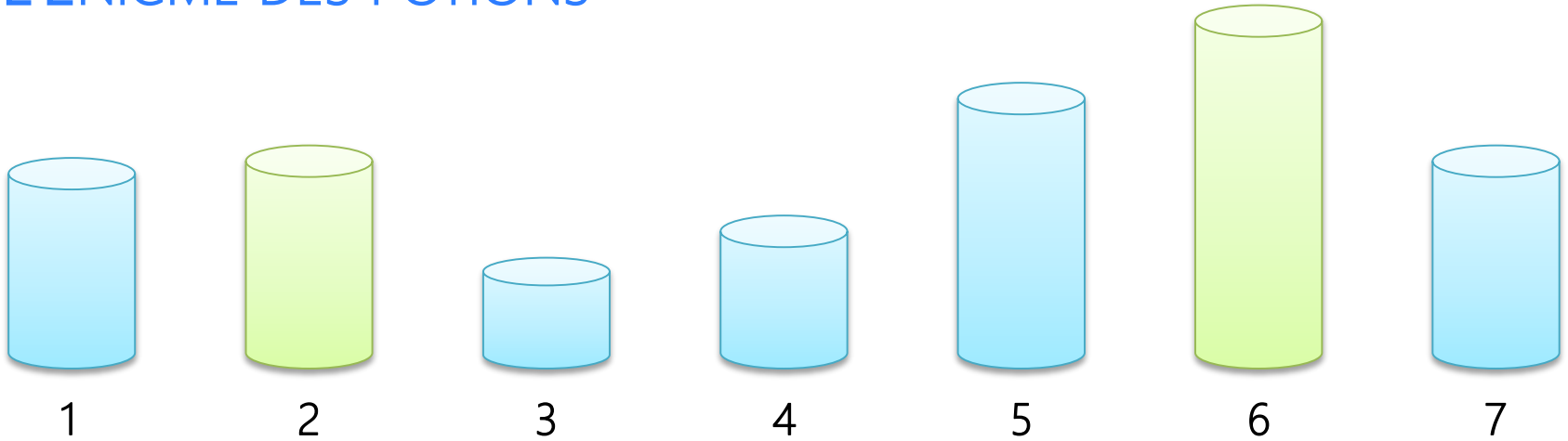
L'ÉNIGME DES POTIONS



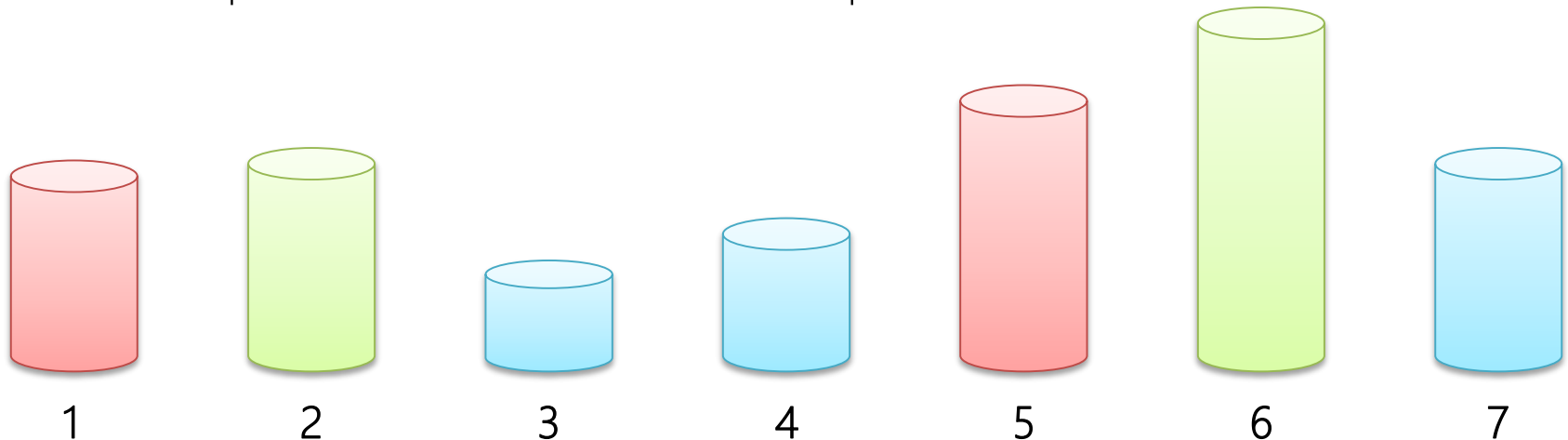
- Indice 4 : Les potions 2 et 6 ont le même contenu
 - ↳ Elles contiennent soit du poison, soit du vin d'ortie
- Indice 2 : La plus grande potion (6) n'est pas un poison
 - ↳ Les potions 2 et 6 contiennent du vin d'ortie



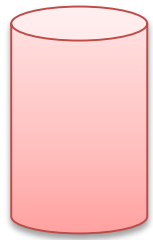
L'ÉNIGME DES POTIONS



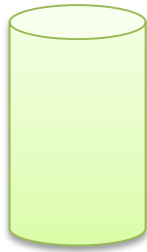
- Indice 1 : A gauche de chaque vin d'ortie se trouve un poison
↳ Les potions 1 et 5 contiennent du poison



L'ÉNIGME DES POTIONS



1



2



3



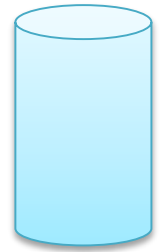
4



5

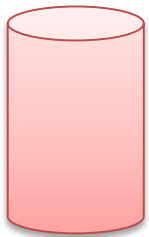


6

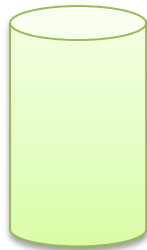


7

- Indice 2 : Les potions 1 et 7 sont différentes et ne permettent pas d'avancer
 - ↳ La potion 7 contient la potion qui permet de reculer



1



2



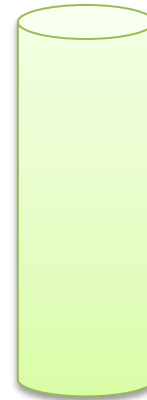
3



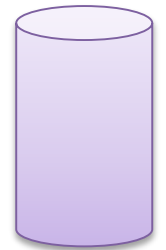
4



5

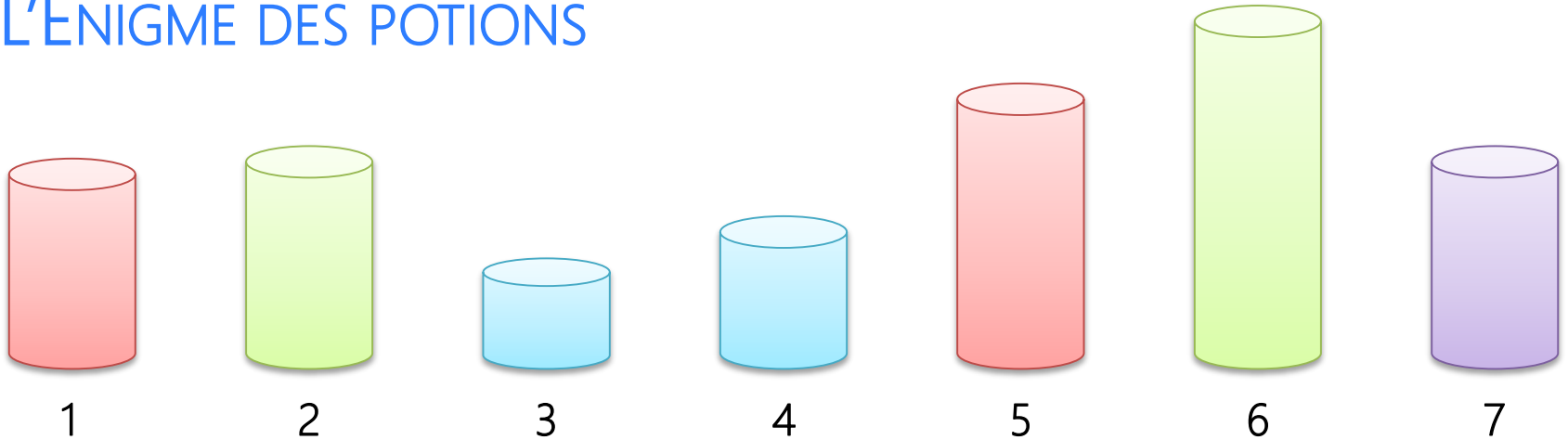


6

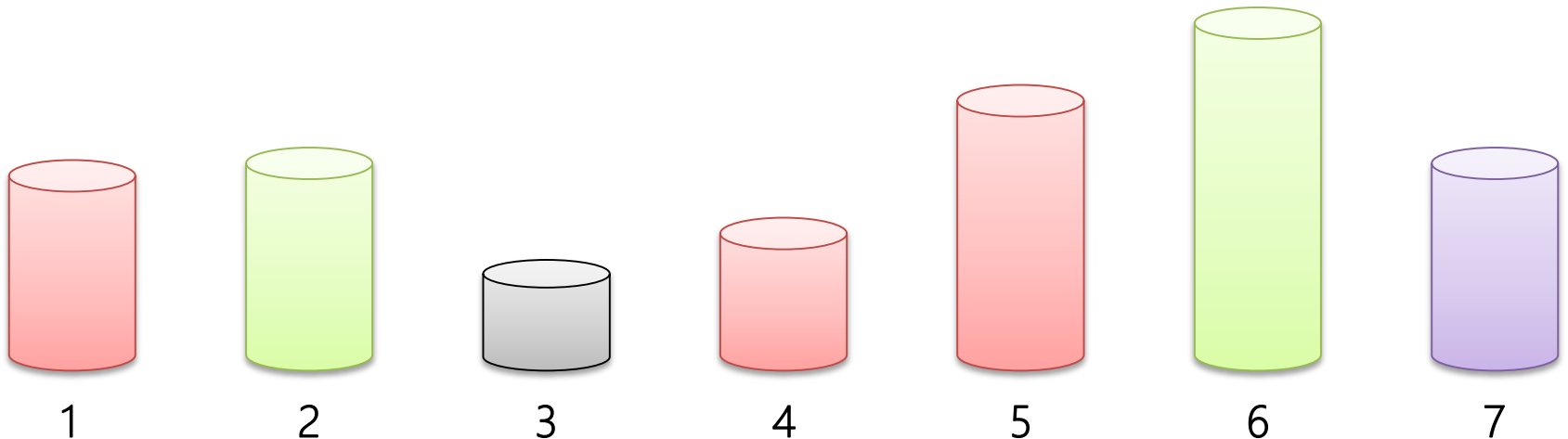


7

L'ENIGME DES POTIONS

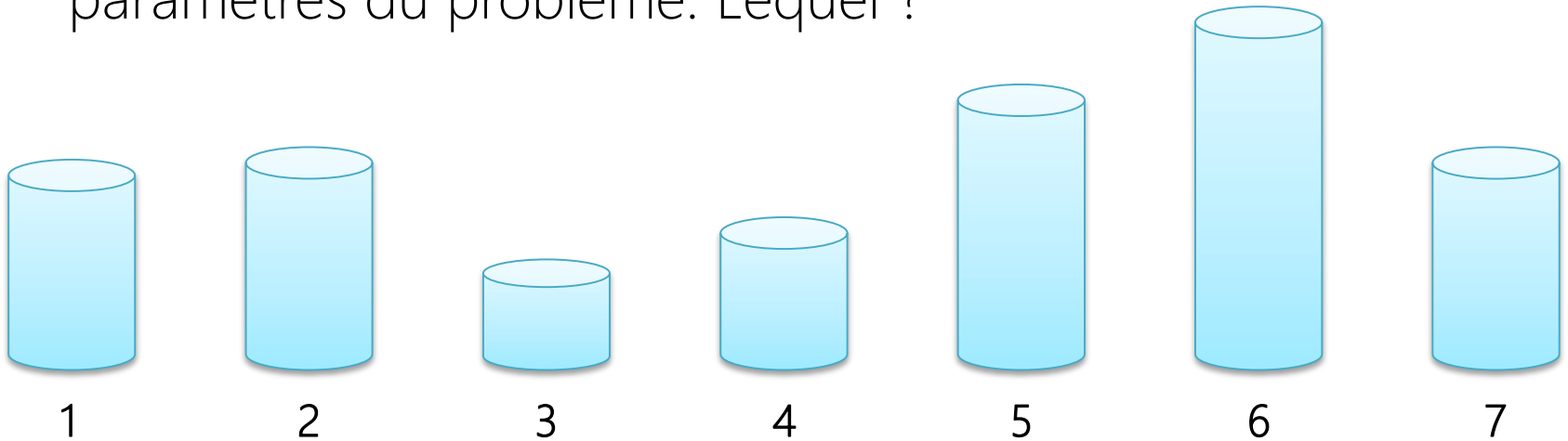


- Indice 3 : La plus petite potion (3) n'est pas un poison
 - ↪ La potion 3 contient la potion qui permet d'avancer
 - ↪ La potion 4 contient le dernier poison



LA VÉRITÉ SUR L'ÉNIGME DES POTIONS

- Question : Nous avons arbitrairement fixé l'un des paramètres du problème. Lequel ?



Les tailles des potions !

- Question : Peut-on trouver une configuration où l'énigme est insoluble, car il n'est pas possible de trancher ?

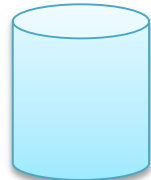
UN CAS INSOLUBLE



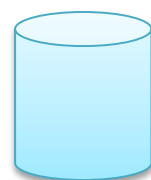
1



2



3



4



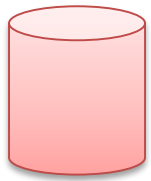
5



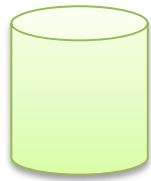
6



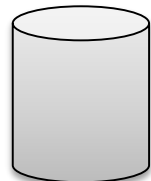
7



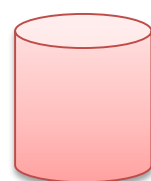
1



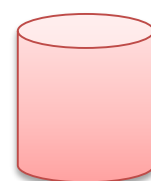
2



3



4



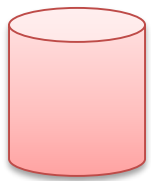
5



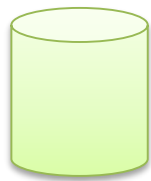
6



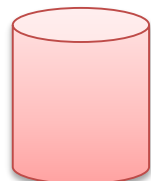
7



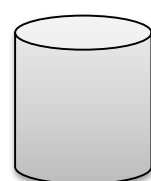
1



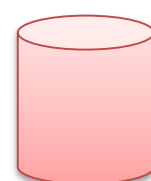
2



3



4



5



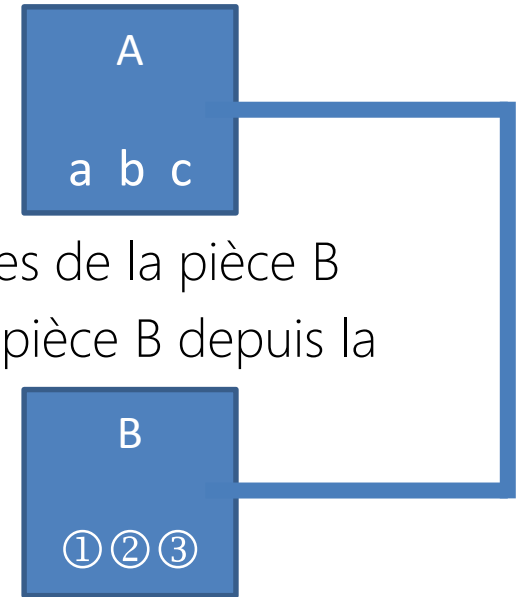
6



7

LES 3 AMPOULES

- Pièce A (position de départ)
 - La pièce A contient 3 interrupteurs : a, b, c
 - Chaque interrupteur contrôle l'une des ampoules de la pièce B
 - Il est impossible de voir ce qui se passe dans la pièce B depuis la pièce A
- Pièce B
 - La pièce B contient 3 ampoules initialement éteintes (1, 2 et 3)
 - Une fois entré dans la pièce B, il n'est possible d'en ressortir qu'en répondant correctement à l'énigme
- Enigme : Comment déterminer quel interrupteur commande quelle ampoule ?



LES MOINES MALADES



- Un étrange monastère
 - 40 moines vivent dans un monastère
 - Ils ont fait vœu de ne jamais communiquer entre eux (en aucune manière)
 - Ils ont fait vœu de ne jamais regarder leur propre image ou reflet
 - Ils se ne croisent qu'une fois par jour, pour déjeuner tous ensemble à midi
- Un midi :
 - Les moines sont prévenus qu'une terrible maladie s'est déclarée
 - Il y a au moins un malade parmi eux
 - Les premiers symptômes sont simples : un point rouge au milieu du front
 - Tout moine convaincu qu'il est malade doit quitter le monastère
 - Ce midi-là, personne ne quitte le monastère
- Jours suivants
 - Le lendemain, personne ne quitte le monastère
 - Le surlendemain, plusieurs moines se lèvent et quittent le monastère
- Question : Combien sont-ils ?

LES MOINES MALADES

- Supposition 1 : Il n'y a qu'un seul malade.
 - Le premier midi, le moine malade ne voit aucun moine malade
 - Comme il y a au moins un malade, c'est donc lui
 - Comme il n'y a pas de départ le premier jour, la supposition 1 est fausse
- Supposition 2 : Il y a deux malades.
 - Le premier midi, les deux moines malades ne voient chacun qu'un moine malade
 - Comme personne n'est parti le premier jour, chacun comprend qu'il est malade aussi et tous deux quittent le monastère
 - Comme il n'y a pas de départ le deuxième jour, la supposition 2 est fausse
- Supposition 3 : Il y a trois malades.
 - Le premier midi, les trois moines malades voient chacun deux moines malades
 - Comme personne n'est parti le deuxième jour, chacun comprend qu'il est malade aussi et tous trois quittent le monastère
- Solution générale : si un départ a lieu le n -ième jour, c'est qu'il y a n malades, et tous partent ce jour-là

LE TRÉSOR DES PIRATES

- Cinq pirates doivent se partager un trésor de 100 pièces d'or
 - Chacun doit prendre la parole, du plus jeune au plus vieux
 - Chacun propose un partage du trésor (ex : « Tout pour moi »)
 - Si la stricte majorité des pirates accepte, chacun part avec sa part
 - Sinon, le plus jeune est exécuté et on passe au nouveau plus jeune

- Les pirates sont tous intelligents et sanguinaires.

Chacun veut donc (dans cet ordre) :

- Rester en vie
 - Optimiser ses gains
 - Tuer un maximum de pirates
- Questions :
 - Le plus jeune a-t-il une chance de s'en sortir ?
 - Le cas échéant, combien de pièces d'or peut-il récupérer ?



LE TRÉSOR DES PIRATES

- Nommons A, B, C, D et E nos pirates, du plus jeune au plus vieux.
- Si E propose un partage,
 - C'est qu'il est le seul survivant
 - E part avec les 100 pièces d'or
- Si D propose un partage
 - C'est qu'il ne reste plus que lui et E.
 - Il ne peut rien proposer d'intéressant à E
 - D ne veut absolument pas en arriver là
- Si C propose un partage
 - C'est qu'il ne reste plus que lui, D et E.
 - D sera d'accord avec n'importe quelle offre (sinon il meurt)
 - C part avec les 100 pièces d'or



LE TRÉSOR DES PIRATES

- Si B propose un partage,
 - C'est qu'il ne reste plus que lui, C, D et E.
 - D et E seront d'accord avec lui s'ils reçoivent au moins une pièce d'or
 - B peut donc proposer le partage suivant :
 - 98 pièces pour B
 - 1 pièce pour D
 - 1 pièce pour E
- Finalement, lorsque A prend la parole,
 - Il lui suffit de deux voix en plus de la sienne
 - C se contentera d'une pièce, et D (ou E) de deux
 - A peut donc proposer le partage suivant :
 - 97 pièces pour A
 - 1 pièce pour C
 - 2 pièces pour E



LE PARADOXE DES ANNIVERSAIRES

- Question : Combien de personnes faut-il réunir dans une pièce pour avoir une chance sur deux que deux soient nées le même jour de l'année ?
- Réponse : 23
- Question : Combien de personnes faut-il réunir dans une pièce pour avoir 99% de chances que deux soient nées le même jour ?
- Réponse : 57

LE PARADOXE DES ANNIVERSAIRES

- Remarque : pour simplifier, on suppose toutes les années sont non bissextiles

- La probabilité qu'un évènement E arrive peut s'écrire :

$$p(E) = \frac{\text{Nombre de cas qui vérifient } E}{\text{Nombre total de cas possibles}}$$

- S'il y a n personnes dans la pièce, le nombre total de cas possibles est 365^n (365 dates possibles pour chaque personne)
- Notre évènement E correspond aux cas de figures où il y a au moins deux personnes dans la pièce qui sont nées le même jour de l'année
- On va s'intéresser à l'évènement contraire \bar{E} , c'est-à-dire les cas de figures où toutes les personnes de la pièce sont nées des jours de l'année différents.
- Si on note \bar{E} l'évènement contraire de l'évènement E , on a

$$p(E) = 1 - p(\bar{E})$$

LE PARADOXE DES ANNIVERSAIRES

- On a donc

$$p(\bar{E}) = \frac{\text{Nombre de cas qui vérifient } \bar{E}}{\text{Nombre total de cas possibles}} = 1 - p(E)$$

- Estimons le nombre de cas qui vérifient \bar{E} s'il y a n personnes dans la pièce :
 - Il y a 365 dates possibles pour le premier invité
 - Il reste 364 dates pour le second invité
 - Il reste 363 dates pour le troisième invité
 - ...
 - Il reste $365 - n + 1$ dates pour le n -ième invité
 - Soit finalement un nombre de cas qui vérifient \bar{E} qui vaut

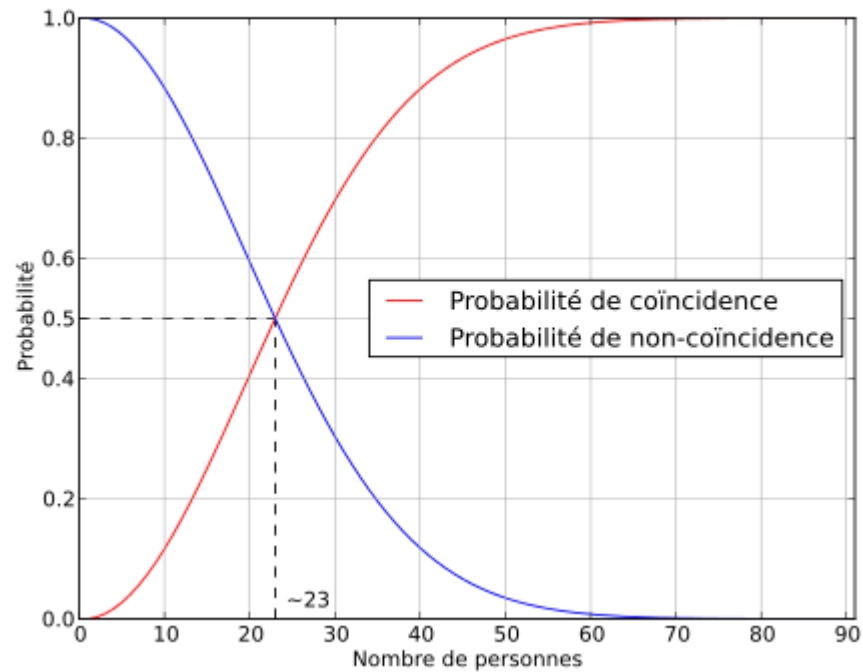
$$365 \times 364 \times 363 \times \dots \times (365 - n + 1) = \frac{365!}{(365 - n)!}$$

- D'où finalement

$$p(\bar{E}) = \frac{\left(\frac{365!}{(365 - n)!}\right)}{365^n} \quad \text{et} \quad p(E) = 1 - \frac{\left(\frac{365!}{(365 - n)!}\right)}{365^n}$$

LE PARADOXE DES ANNIVERSAIRES

n	5	10	20	23	30	50	57	100	366
$p(E)$	2,7 %	11,7 %	41,1 %	50,7 %	70,6 %	97 %	99 %	99,9997 %	100 %



LE PROBLÈME DE MONTY HALL



- Un plateau de jeu télévisé :
 - Un candidat est face à 3 portes
 - Derrière l'une des portes se trouve une voiture
 - Derrière les deux autres portes se trouve une chèvre
 - Le présentateur sait ce qu'il y a derrière chaque porte
- Déroulement du jeu :
 - Le joueur choisit une des portes (sans l'ouvrir)
 - Le présentateur ouvre l'une des portes restantes derrière laquelle se trouve une chèvre
 - Le présentateur offre le choix suivant au candidat :
 - Rester sur son choix initial
 - Changer d'avis pour choisir la porte restante
- Question : Quel est le meilleur choix pour le candidat ?

LE PROBLÈME DE MONTY HALL

- Réponse : Le candidat a deux fois plus de chances de gagner la voiture s'il change d'avis !
- Démonstration informelle :
 - Prenons deux candidats jouant à ce jeu plusieurs fois de suite.
 - Le candidat A s'en tient toujours à son choix initial : il a donc toujours 1 chance sur 3 de gagner la voiture.
 - Le candidat B change toujours d'avis pour prendre la porte restante. Deux cas de figure sont possibles :
 - Si son premier choix était le bon (1 chance sur 3), il perd
 - Si son premier choix n'était pas le bon (2 chances sur 3), il gagne.
 - En moyenne, le candidat A gagnera donc 1 fois sur 3, alors que le candidat B gagnera 2 fois sur 3.

LE PARADOXE DE SIMPSON

- Alice et Bob préparent leurs TPE et font des expériences
- Au premier trimestre,
 - Alice réussit 60 % de ses expériences
 - Bob réussit 90 % de ses expériences
- Au deuxième trimestre,
 - Alice réussit 10 % de ses expériences
 - Bob réussit 30 % de ses expériences
- Pourtant, au cours de ces deux trimestres, Alice a réussi davantage d'expériences que Bob, et tous deux ont fait le même nombre d'expériences
 - Comment est-ce possible ?

LE PARADOXE DE SIMPSON

- Explication :

	Trimestre 1	Trimestre 2	Total
Alice	60 / 100 60 %	1 / 10 10 %	61 / 110 55,45 %
Bob	9 / 10 90 %	30 / 100 30 %	39 / 110 35,45%

- Verdict : Attention lorsqu'on manipule des pourcentages !
- Nom officiel : Paradoxe de Simpson

QUELQUES LIENS POUR LES VACANCES

OPENCLASSROOM



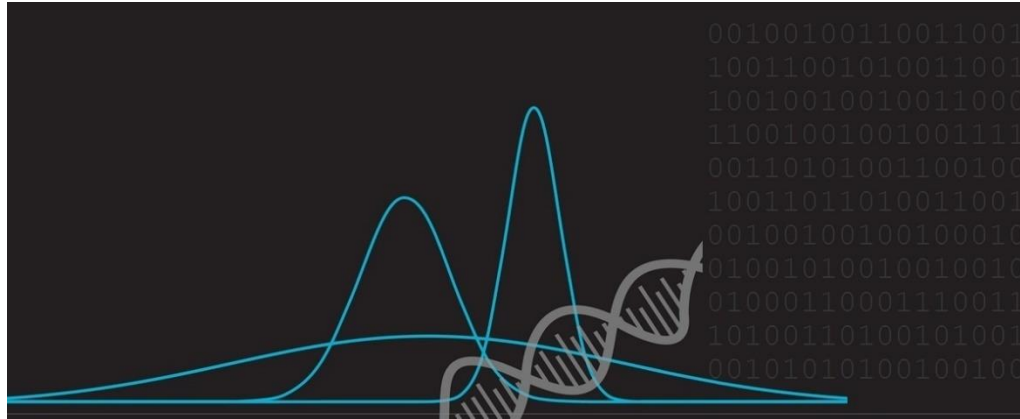
- Anciennement nommé le Site du Zéro
- Un vaste choix de tutoriels sur différents sujets
 - Langages de programmation
 - Sites web (HTML, CSS, JS, etc.)
 - Infographie 2D et 3D
- Site Internet : <http://fr.openclassrooms.com/>

WHAT IF ?



- Chaque semaine, une question physico-ludique est traité avec un certain sérieux :
 - [Un train à grande vitesse pourrait-il prendre un looping ?](#)
 - [Quels seraient les apports journaliers recommandés pour un T-rex lâché dans New-York ?](#)
 - [Peut-on se servir du recul des armes à feu pour s'envoler ?](#)
 - [Quelle est la puissance \(en Watts\) déployée par Maître Yoda ?](#)
- Site géré par Randall Munroe, ancien consultant sur les robots à la NASA et auteur du blog xkcd
- Site internet : <https://what-if.xkcd.com/>

DATA GENETICS



- Blog autour des statistiques, des mathématiques et de l'informatique
- Des aspects ludiques et des démonstrations très poussées
 - Echapper à un monstre
 - Jouer aux dés ou aux cartes
 - etc.
- Site internet : <http://www.datagenetics.com/blog.html>

.... Quelques liens pour les vacances

FRANCE IOI



- De nombreux cours et exercices
- Des préparations pour les Olympiades Internationales d'Informatiques
- Site internet : <http://www.france-ioi.org/>

.... Quelques liens pour les vacances

BONNES ÉPREUVES,
ET
BONNES VACANCES !