Choisissez les bons poids

Année 2015- 2016

Ugo PIOLA (élève de 2de), Roxana MIHAI (X-ième), Eduard OLTEANU (X-ième) et Mihail TANASESCU (X-ième),

Encadrés par Marie DIUMENGE lycée François Arago de Perpignan, France, Melania NICOLAE Collège National "Bogdan Petriceicu Hasdeu" de Buzau, Roumanie

Établissements : Lycée François Arago (Perpignan), Collège National "Bogdan Petriceicu Hasdeu" de Buzau, Roumanie

Chercheurs : Robert BROUZET, Bogdan ENESCU

Présentation du sujet:

On a une balance à plateaux de pesée et on souhaite utiliser le moins de poids d'équilibrage (chacun pèse un nombre entier de grammes) de sorte que l'on puisse peser tout objet dont le poids est toujours un nombre entier et ne dépasse pas 500 grammes. Par exemple, une solution connue est celle où l'ensemble des poids est {1, 2, 4, 8, 16, 32, 64, 128, 256}. Pouvez-vous trouver d'autres solutions ? Combien sont possibles ?

Les solutions trouvées:

- Avec des puissances de 2, nous avons trouvé 75 solutions
- Avec des puissances de 3, nous avons trouvé 39183 solutions
- Au total, 39258 solutions trouvées

I. Preuve que c'est impossible avec seulement 8 poids :

Au début, nous avons essayé de prouver qu'avec le système proposé dans la déclaration (1), nous pourrions peser tous les poids jusqu'à 500 avec une feuille de calcul, mais nous n'avons pas trouvé de formule générale et nous avons donc dû vérifier les 500 poids un par un. Ensuite, nous avons abandonné, mais nous étions convaincus que nous pouvions peser tous les poids avec ce système, et nous avons cherché un système de poids.

Par récurrence forte :
On prouve que pour toute série de n poids permettant de tout peser jusqu'à 500g , le k-ième poids $p_k \leq 2^{k-1}$
La notation p_k se réfère à chaque poids de la série comme suit: $p_1=1, p_2=2, p_3=4, p_4=8\ldots$ (2)

Initialisation : $p_1=1\leq 2^0$, parce que le premier nombre de la série est 1.

Hérédité :
On suppose que pour tout $i \leq k$, $p_i \leq 2^{i-1}$
On veut montrer que $p_{k+1} \leq 2^k$
$p_1+p_2+\ldots+p_k \leq 2^0+2^1+\ldots+2^{k-1}=\frac{1-2^k}{1-2}=2^k-1$

Démonstration :

$2^0+2^1+\ldots+2^{k-1}=S$
$2^1+2^2+\ldots+2^k=2\cdot S$
Nous les soustrayons: $2\cdot S-S = 2^1+2^2+\ldots+2^k-2^0-2^1-\ldots-2^{k-1} \Rightarrow S=2^k-2^0 \Rightarrow S=2^k-1$

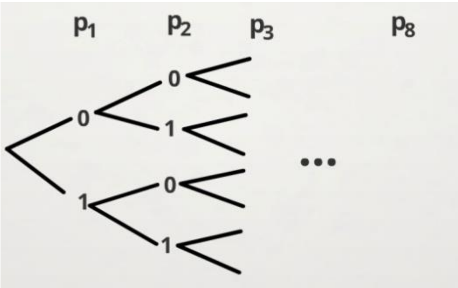
Donc si le système est complet (3), le poids suivant : $p_{k+1} \leq 2^k$

Donc avec 8 poids :
$S \leq 2^8 - 1 = 255$ (insuffisant pour peser des poids jusqu'à 500 !), où S est la somme de tous les poids.

Avec 9 poids :
$S \leq 2^9-1 = 511$ (suffisant)

II. Une autre méthode: avec 8 poids $\{p_1, p_2, p_3, \ldots, p_8\}$

Nous pouvons utiliser le poids ou non, donc :



$2\cdot 2\cdot\ldots\cdot 2=2^8=256$ pesées de poids au maximum

On a donc prouvé que lorsqu'on utilise 8 poids de pesée, c'est impossible (4) car l'on aura que 256 poids totaux maximum, alors qu'on cherche à en peser 500.

On a donc prouvé avec deux méthodes différentes, qu'il était impossible de peser tous les poids entiers de 1 à 500 avec des séries de 8 poids.

III. Preuve que c'est possible avec 9 poids :

La modification (5) d'un nombre naturel « n » du système décimal en un nouveau système de numérotation se fait en divisant le nombre donné par la nouvelle base « b »; les restes successifs et le dernier quotient (différent de 0) sont les chiffres du nombre donné, considérés dans l'ordre inverse.

Si a_k est le dernier quotient (différent de 0), et a_{k-1}, ..., a_1, a_0 sont les restes de la division de n au nombre b, écrit en ordre inverse, alors nous écrivons n comme $\overline{\mathbf{a_k a_{k-1} \dots a_0}}$ et on dit que n est écrit dans la base b, où $b \geq 20$ et $\leq a_k, a_{k-1}, \dots, a_1, a_0 < b$.

Tout nombre naturel n peut être écrit comme $n=a_k \cdot \boldsymbol{b^k} + a_{k-1} \cdot \boldsymbol{b^{k-1}} + \dots + a_1 \cdot b + a_0$, où $a_0, a_1, a_2, \dots, a_k, b, k \in \mathbb{N}$, $b \geq 2$, $0 \leq a_i < b$, pour $i = 0,\dots,k$ et $a_k \neq 0$.

Alors, tout entier n (dans ce cas, au plus égale à 500) s'écrit en base 2 comme ceci:

$\boxed{n=a_k \cdot 2^k + a_{k-1} \cdot 2^{k-1} + a_{k-2} \cdot 2^{k-2} + \dots + a_1 \cdot 2^1 + a_0}$

où k est un entier positif, $a_k = 1$ et $a_i \in \{0,1\}$, pour tous les $i = 0,\dots,k$.

Si $a_i = 0$, nous n'avons pas besoin de ce poids afin de peser, et si $a_i = 1$, nous en avons besoin. Donc, avec 1g, 2g, 4g, 8g, 16g, 32g, 64g, 128g, 256g (9 au total et leur somme est 511) nous pouvons peser tous les poids de 1 à 500 (6), comme nous l'avons prouvé dans le premier paragraphe (7).

Le nombre minimum de poids sera 9.

IV. Autres systèmes possibles:

Nous soustrayons 1 de chaque nombre (de la solution donnée dans l'énonciation du problème: 1, 2, 4…256), du final au début, tant que la somme (S) est ≥ 500 (afin qu'il puisse contenir tous les numéros $n \in \{1,\dots, 500\}$). Les p_k ($k \in \{0,\dots, 8\}$) sont les termes de la série contenant les pouvoirs de 2 (8), qui constituent la solution initiale ($p_0=1$, $p_1=2$, $p_2=4\dots$).

Donc : (9)

Les nombres non modifiés	Leur somme	Les nombres modifiés				La somme finale
de p_0 à p_7	255				$p_8=\overline{245,256}$	$S=\overline{500,511}$
de p_0 à p_6	127			$p_7=127$	$p_8=\overline{246,255}$	$S=\overline{500,509}$
				$p_7=126$	$p_8=\overline{247,254}$	$S=\overline{500,507}$
				$p_7=125$	$p_8=\overline{248,253}$	$S=\overline{500,505}$
				$p_7=124$	$p_8=\overline{249,252}$	$S=\overline{500,503}$
				$p_7=123$	$p_8=\overline{250,251}$	$S=\overline{500,501}$
de p_0 à p_5	63		$p_6=63$	$p_7=127$	$p_8=\overline{247,254}$	$S=\overline{500,507}$
				$p_7=126$	$p_8=\overline{248,253}$	$S=\overline{500,505}$
				$p_7=125$	$p_8=\overline{249,252}$	$S=\overline{500,503}$
				$p_7=124$	$p_8=\overline{250,251}$	$S=\overline{500,501}$
			$p_6=62$	$p_7=126$	$p_8=\overline{249,252}$	$S=\overline{500,503}$
				$p_7=125$	$p_8=\overline{250,251}$	$S=\overline{500,501}$
de p_0 à p_4	31	$p_5=31$	$p_6=63$	$p_7=126$	$p_8=\overline{249,252}$	$S=\overline{500,503}$
				$p_7=125$	$p_8=\overline{250,251}$	$S=\overline{500,501}$

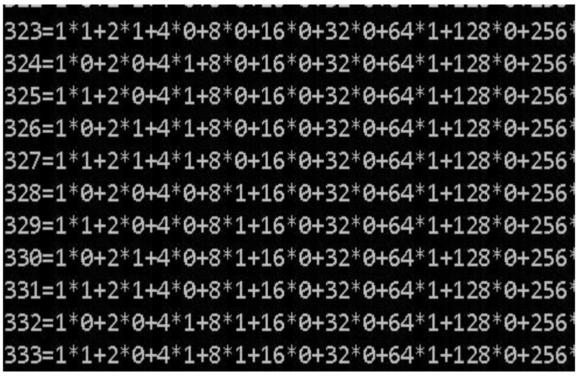
Si, par exemple, p_8=244 (sur la deuxième ligne de la table), la somme finale sera 499, ce qui signifie que nous ne pourrons pas peser un objet de 500 grammes.

=> 75 solutions obtenues à partir de la solution donnée dans l'énoncé.

V. Algorithme:

Nous avons également fait un code en C++ qui décompose un nombre avec la solution donnée par l'énoncé:

```
int main()
{int s=0;
for(n=1;n<=500;n++)
for(int a0=0; a0<2; a0++)
for(int a1=0; a1<2; a1++)
for(int a2=0; a2<2; a2++)
for(int a3=0; a3<2; a3++)
for(int a4=0; a4<2; a4++)
for(int a5=0; a5<2; a5++)
for(int a6=0; a6<2; a6++)
for(int a7=0; a7<2; a7++)
for(int a8=0; a8<2; a8++)
if(n==1*a0+2*a1+4*a2+8*a3+16*a4+32*a5+64*a6
+128*a7+256*a8)
{cout<<x<<"=1·"<<a0<<"+2·"<<a1<<"+4·"<<
a2<<"+8·"<<     a3<<"+16·"<<     a4<<"+32·"<<
a5<<"+64·"<<    a6<<"+128·"<<    a7<<"+256·"<<
a8<<endl;
s+=1;}}cout<<endl<<s; return 0;}
```


```
323=1*1+2*1+4*0+8*0+16*0+32*0+64*1+128*0+256*
324=1*0+2*0+4*1+8*0+16*0+32*0+64*1+128*0+256*
325=1*1+2*0+4*1+8*0+16*0+32*0+64*1+128*0+256*
326=1*0+2*1+4*1+8*0+16*0+32*0+64*1+128*0+256*
327=1*1+2*1+4*1+8*0+16*0+32*0+64*1+128*0+256*
328=1*0+2*0+4*0+8*1+16*0+32*0+64*1+128*0+256*
329=1*1+2*0+4*0+8*1+16*0+32*0+64*1+128*0+256*
330=1*0+2*1+4*0+8*1+16*0+32*0+64*1+128*0+256*
331=1*1+2*1+4*0+8*1+16*0+32*0+64*1+128*0+256*
332=1*0+2*0+4*1+8*1+16*0+32*0+64*1+128*0+256*
333=1*1+2*0+4*1+8*1+16*0+32*0+64*1+128*0+256*
```

Le code fait ce qui suit: pour chaque numéro n, il recherche les coefficients qui décident si une puissance de 2 (1,2,4,8 ...) est utilisée dans la formation du nombre n. Le coefficient varie entre 0 et 1. Par conséquent, s'il est 0, la puissance de 2 n'est pas nécessaire pour former le nombre n. Sinon, si le coefficient est 1, la puissance de 2 sera ajoutée à la somme qui représente le nombre n à atteindre. Pour chaque coefficient trouvé pour la première puissance de 2, le code générera et vérifiera les autres possibilités de coefficients pour les prochaines puissances. Après avoir trouvé une seule possibilité d'écrire le nombre n en tant que somme de puissance de 2, il sera affiché sur l'écran, puis le code vérifiera les autres possibilités et les affichera aussi si elles sont correctes.

VI. Recherche d'une autre solution en utilisant les deux plateaux de pesée:

Il y a aussi une autre solution très intéressante: si nous mettons des poids sur chaque plateau de pesée, nous pouvons utiliser juste 7 poids.

En utilisant la modification d'un nombre naturel de la base 10 à la base 3 (expliqué au paragraphe III), nous écrivons $n = a_k\cdot 3^k+a_{k-1}\cdot 3^{k-1}+\ldots+a_1\cdot 3^1+a_0\cdot 3^0$, où $k \geq 0$, $a_k \neq 0$, $a_i \in \{0,1,2\}$, pour $i = 0,\ldots k$.

Si l'un de $a_k, a_{k-1}, \ldots, a_0$ est 2, on va le remplacer par (3-1), on annule les crochets sur la distribution de la multiplication vers l'addition (10) et on obtient $c_{k+1}\cdot 3^{k+1}+c_k\cdot 3^k + \ldots + c_1\cdot 3^1 + c_0\cdot 3^0$, où $k \in \mathbb{N}$, $c_{k+1} \neq 0$, $c_i \in \{-1,0,1\}$ pour $i = 0,\ldots k+1$.

Comme $n \leq 500$, alors $k = 6$ car

$3^0 + 3^1 + 3^2 + 3^3 + 3^4 + 3^5 = 364 < 500$ (insuffisant)

$3^0 + 3^1 + 3^2 + 3^3 + 3^4 + 3^5 + 3^6 = 1093 > 500$ (suffisant)

En effet, nous pouvons écrire tous les nombres $n \leq 500$ avec la formule:

$N = 3^0 \cdot c_0 + 3^1 \cdot c_1 + 3^2 \cdot c_2 + 3^3 \cdot c_3 + 3^4 \cdot c_4 + 3^5 \cdot c_5 + 3^6 \cdot c_6$

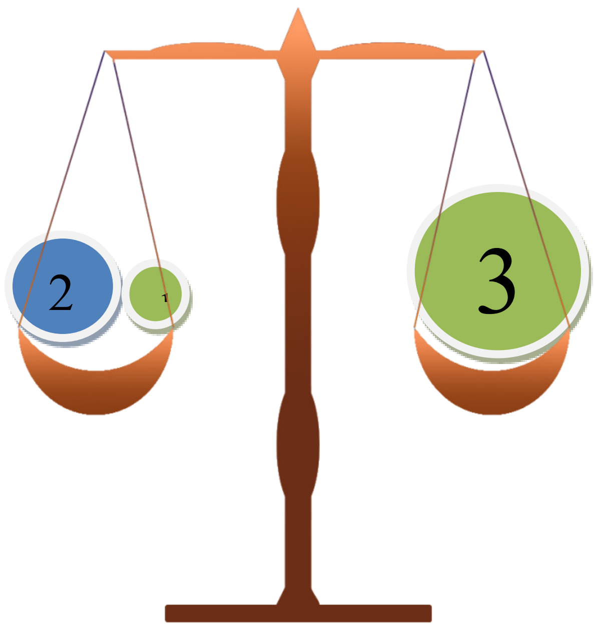
où $c_i \in \{-1,0,1\}$, pour tous les $i = 0,\ldots 6$.

Par exemple :

$2 = -1+3$
$4 = 1+3$
$5 = -1-3+9$
$6 = -3+9$
....
$500=-1-3-9+27-243+729$

Avec le **code** on a trouvé les combinaisons de signes pour chaque nombre de 1 à 500:

```
int main()
{int s=0;
for(n=1;n<=500;n++)
for(int c0=−1; c0<2; c0++)
for(int c1=−1; c1<2; c1++)
for(int c2=−1; c2<2; c2++)
for(int c3=−1; c3<2; c3++)
for(int c4=−1; c4<2; c4++)
for(int c5=−1; c5<2; c5++)
for(int c6=−1; c6<2; c6++)
if(n==1*c0+3*c1+9*c2+27*c3+81*c4+243*c5+729*c6)
{cout<<n<<"=1·("<<c0<<")+3·("<<c1<<")+9·("<<c2<
<")+27·("<<c3<<")+81·("<<c4<<")+243·("<<c5<<")+7
29· ("<<c6<<")"<<endl;
s+=1;}
 return 0;}
```



Nous avons adapté l'algorithme pour cette approche différente: pour chaque numéro n, il recherche les coefficients qui décident si une puissance de 3 (1,3,9,27 ...) est utilisée dans la formation du nombre n. Le coefficient varie entre −1 et 1. Par conséquent, s'il est −1, nous devons mettre le poids (la puissance de 3) sur le même côté que celle que nous devons peser (n). Si c'est 0, le poids n'est pas nécessaire pour former le numéro n. Sinon, si le coefficient est 1, le poids sera placé sur l'autre côté, donc il sera ajouté à la somme qui représente le nombre n à atteindre. Pour chaque coefficient trouvé pour la première puissance de 3, le code générera et vérifiera les autres possibilités de coefficients pour les prochaines puissances. Après avoir trouvé une seule possibilité d'écrire le numéro n en tant que somme et / ou différence de pouvoirs de 3, il sera affiché à l'écran, puis le code vérifiera les autres possibilités et les affichera également si elles sont correctes.

Tout comme dans la partie avec les puissances de 2, nous soustrayons 1 plusieurs fois de chaque poids (1,3,9,27,81,243,729) pour savoir combien de série de 7 poids sont solutions.

```
494=1*(-1)+3*(0)+9*(1)+27*(0)+81*(0)+243*(-1)+729*(1)
495=1*(0)+3*(0)+9*(1)+27*(0)+81*(0)+243*(-1)+729*(1)
496=1*(1)+3*(0)+9*(1)+27*(0)+81*(0)+243*(-1)+729*(1)
497=1*(-1)+3*(1)+9*(1)+27*(0)+81*(0)+243*(-1)+729*(1)
498=1*(0)+3*(1)+9*(1)+27*(0)+81*(0)+243*(-1)+729*(1)
499=1*(1)+3*(1)+9*(1)+27*(0)+81*(0)+243*(-1)+729*(1)
500=1*(-1)+3*(-1)+9*(-1)+27*(1)+81*(0)+243*(-1)+729*(1)
```

Nous avons utilisé le code pour soustraire pratiquement à la main (parce que nous n'avons pas trouvé un algorithme qui pourrait nous aider) de chaque puissance de 3, de sorte que la somme finale ne dépasse pas 500 et chaque fois que nous avons obtenu un certain nombre de solutions (dont le code a compté), que nous avons résumé et obtenu 39183 (11) solutions au total à partir de la solution principale (1,3,9,27,81,243,729).

Donc:

- Si on pose les poids de pesée des 2 côtés de la balance, la plus petite série est de 7 poids et il existe 39183 systèmes de 7 poids solutions.
- Si on pose les poids de pesée d'un seul côté de la balance, la plus petite série est de 9 poids et il existe 75 série possible de 9 poids.

Notes d'édition

(1) Il faut lire : « présentation du sujet ».

(2) Les valeurs $p_1=1, p_2=2, p_3=4, p_4=8\ldots$ sont données à titre d'exemple. Elles peuvent être différentes, mais elles doivent être rangées en ordre croissant $p_1 \leq p_2 \leq p_3 \leq p_4\ldots$

(3) Un système complet de poids est un système de poids permettant de peser tous les objets dont les poids (en gramme) sont des nombres entiers compris entre 1 et 500.

(4) Ce qui est impossible : de peser tous les objets d'un poids ≤ 500g avec un système de 8 poids.

(5) « Modification » ici veut dire qu'on passe de l'écriture en base 10 (décimale) d'un nombre entier à son écriture en base 2 (ou 3 plus loin).

(6) Avec k=8 (donc 9 poids), on peut donc peser tous les objets dont les poids (en gramme) sont des nombres entiers compris entre 1 et 500. Les 9 poids du système sont les poids 1, 2, 4, 8, 16, 32, 64, 128, 256.

(7) Dans le premier paragraphe on a prouvé qu'une condition nécessaire pour un système complet de poids est qu'il comporte au moins 9 poids. Ici on a prouvé que cette condition est suffisante.

(8) Il s'agit ici des « puissances de 2 » (clin d'œil à nos amis roumains)

(9) Il faut avouer que ce tableau n'est pas très clair. Il semble s'agir de faire la liste des poids maximum que l'on peut peser suivant les valeurs modifiées de certains p_k en calculant leur somme. Mais cela ne donne qu'une condition nécessaire pour avoir un système complet.

(10) On peut procéder comme cela, en partant de la droite et pour chaque 2 remplacé par 3-1 reporter une retenue de 1 sur le coefficient précédent puis en continuant avec ce coefficient modifié ; mais il ne faudra pas que le coefficient de gauche soit égal à 2 ou le devienne après les retenues. Cela veut dire que l'écriture ne doit pas commencer par une suite de 1 suivie d'un 2, autrement dit que $n \leq 3^k+3^{k-1}+\ldots+3+1$, qui est la somme de tous les poids du système.

(11) Ce nombre est bien mystérieux.