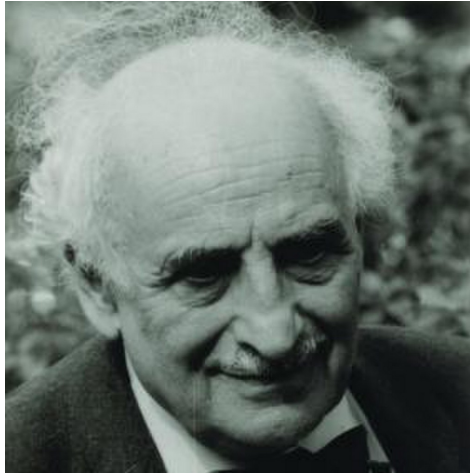


L'énigme de Freudenthal



Énoncé

On choisit deux nombres différents, strictement supérieurs à 1, dont la somme est inférieure à 100. On indique à Sofiane la somme de ces deux nombres et à Priscillia le produit. On assiste alors à la conversation suivante :

Priscillia : Je ne connais pas ces deux nombres.

Sofiane : Je savais que tu dirais ça !

Priscillia : Ah, alors j'ai trouvé !

Sofiane : Ah, alors moi aussi !

Questions :

1. 🖥️ Quels sont ces deux nombres ?
2. 🍌🖥️ Quel est le plus grand entier, tel que si l'on remplace dans l'énoncé précédent 100 par cet entier, il soit toujours possible de trouver les deux inconnues, après avoir assisté à la conversation ci-dessus ?

Solution

1. La solution est (4,13). On peut utiliser le code Python suivant pour le montrer.

```
python 📄 Télécharger le code  
  
import numpy as np  
  
def Denniston(M):  
    prod_to_sums, sum_to_prods = {}, {}  
    for x in range(2, M-2):  
        for y in range(2, x):  
            s = x + y  
            if s > M:  
                break  
            p = x * y  
            prod_to_sums.setdefault(p, []).append(s)  
            sum_to_prods.setdefault(s, []).append(p)  
    prods = sorted(prod_to_sums)  
    sums = sorted(sum_to_prods)  
    prod_index = {p: i for i, p in enumerate(prods)}  
    sum_index = {s: j for j, s in enumerate(sums)}  
    A = np.full((len(prods), len(sums)), -1, dtype=int)
```

```

for p, s_list in prod_to_sums.items():
    i = prod_index[p]
    for s in s_list:
        j = sum_index[s]
        A[i, j] = 0

# 1
for p, s_list in prod_to_sums.items():
    if len(s_list) == 1:
        j = sum_index[s_list[0]]
        A[prod_index[p], j] = 1

# 2
for s, p_list in sum_to_prods.items():
    j = sum_index[s]
    if any(A[prod_index[p], j] == 1 for p in p_list):
        for p in p_list:
            if A[prod_index[p], j] == 0:
                A[prod_index[p], j] = 2

# 3
for p, s_list in prod_to_sums.items():
    zeros = [s for s in s_list if A[prod_index[p], sum_index[s]] == 0]
    if len(zeros) > 1:
        for s in zeros:
            A[prod_index[p], sum_index[s]] = 3

# 4
for s, p_list in sum_to_prods.items():
    zeros = [p for p in p_list if A[prod_index[p], sum_index[s]] == 0]
    if len(zeros) > 1:
        for p in zeros:
            A[prod_index[p], sum_index[s]] = 4

# Solutions
solutions = []
for (i, j) in zip(*np.where(A == 0)):
    p = prods[i]
    s = sums[j]
    y = int((s + np.sqrt(s*s - 4*p)) / 2)
    x = s - y
    solutions.append((x, y))
return solutions

if __name__ == "__main__":
    print(Denniston(100))

```

2. On étudie le problème pour deux nombres entiers distincts $x, y \geq 2$, avec $x + y \leq M$. Pour ce problème avec un M fixé, avec l'algorithme de la question précédente, on pose :

$$P_1(M) \triangleq \{p : 0 \in A[p, :] \text{ après l'étape 1}\},$$

$$S_2(M) \triangleq \{s : 0 \in A[:, s] \text{ après l'étape 2}\},$$

$$P_3(M) \triangleq \{p : 0 \in A[p, :] \text{ après l'étape 3}\},$$

$$S_4(M) \triangleq \{s : 0 \in A[:, s] \text{ après l'étape 4}\}.$$

Pour un produit p on note $M_1(p)$ (resp. $M_3(p)$) l'ensemble des M telles que $p \in P_1(M)$ (resp. $p \in P_3(M)$). De façon analogue on définit $M_2(s)$ et $M_4(s)$ pour les sommes. Pour un produit p (resp. une somme s), on appelle *somme compatible* de p (resp. *produit compatible* de s) toute valeur $x + \frac{p}{x}$ (resp. $x(s - x)$), où $x \geq 2$ parcourt les diviseurs stricts de p (resp. les entiers

$x \leq s - 2$). L'ensemble des sommes (resp. produits) compatibles de p (resp. s) est notée $SC(p)$ (resp. $PC(s)$).

Théorème (Propriétés topologiques des ensembles M_i). *Pour tout produit p et toute somme s les ensembles $M_1(p), M_2(s), M_3(p), M_4(s)$ satisfont :*

- (a) $M_1(p)$ est soit vide soit une demi-droite $[L, \infty)$.
- (b) $M_2(s)$ est soit vide soit une demi-droite.
- (c) $M_3(p)$ est soit vide, soit une demi-droite, soit un intervalle fini $[L, R]$.
- (d) $M_4(s)$ est soit vide, soit une demi-droite, soit un intervalle fini.

Démonstration. On détaille chacun des points ci-dessous.

(a) On a

$$M_1(p) = \{M > 0, \text{ il y a au moins 2 factorisations } p = xy \text{ avec } x > y \geq 2, x + y \leq M\}.$$

Quand M augmente, d'autres factorisations peuvent apparaître mais jamais disparaissent ; il s'ensuit que la propriété devient vraie à partir d'un certain seuil et reste vraie au-delà.

(b) On a que

$$\begin{aligned} M_2(s) &= \{M > 0, \text{ tous les produits compatibles de } s \text{ sont dans } P_1(M)\} \\ &= \cap_{\pi \in PC(s)} M_1(\pi) \end{aligned}$$

est une demi-droite (ou est vide), comme une intersection de demi-droites qui ont des bornes à gauche.

(c) Avec $s(p), s'(p)$ tels que

$$M_2(s(p)) = \cup_{\sigma \in SC(p)} M_2(\sigma), \quad M_2(s'(p)) = \cup_{\sigma \in SC(p), \sigma \neq s(p)} M_2(\sigma)$$

(si la deuxième union est vide, on peut prendre $s'(p) = 5$ car $M_2(5) = \emptyset$ car $6 \in PC(5)$ et $M_1(6) = \emptyset$), on a que

$$\begin{aligned} M_3(p) &= \{M > 0, \text{ parmi les sommes compatibles de } p \text{ exactement une est dans } S_2(M)\} \\ &= M_2(s(p)) \setminus M_2(s'(p)) \end{aligned}$$

est soit vide, soit une demi-droite, soit un intervalle fini.

(d) Soit s un somme fixée. Soit $\pi \in PC(s)$. On a $s \in SC(\pi)$, donc si $s \neq s(\pi)$, alors $M_2(s) \subset M_2(s'(\pi))$ (par définition de s') et $M_2(s) \cap M_2(s(\pi)) \setminus M_2(s'(\pi)) = \emptyset$. Ainsi, pour $A \subset PC(s)$,

$$\begin{aligned} \cup_{\pi \in A} M_2(s) \cap M_3(\pi) &= \cup_{\pi \in A} M_2(s) \cap M_2(s(\pi)) \setminus M_2(s'(\pi)) \\ &= \cup_{\pi \in A, s=s(\pi)} M_2(s) \setminus M_2(s'(\pi)). \end{aligned}$$

Dans cette dernière union, la borne gauche de l'ensemble des termes est la même que celle de $M_2(s)$. Comme tous ces intervalles ou demi-droites partagent la même borne gauche, ils sont nécessairement imbriqués : l'union totale coïncide donc avec l'un d'entre eux. On peut donc poser $p(s), p'(s)$ tels que

$$\begin{aligned} M_2(s) \cap M_3(p(s)) &= \cup_{\pi \in PC(s)} M_2(s) \cap M_3(\pi), \\ M_2(s) \cap M_3(p'(s)) &= \cup_{\pi \in PC(s), \pi \neq p(s)} M_2(s) \cap M_3(\pi) \end{aligned}$$

(si la deuxième union est vide, on peut prendre $p'(s) = 6$ car $M_3(6) = \emptyset$ car $M_1(6) = \emptyset$). On a donc

$$\begin{aligned} M_4(s) &= \{M > 0, \text{ parmi les produits compatibles de } s \text{ exactement un est dans } S_3(M)\} \\ &= M_2(s) \cap M_4(s) \\ &= M_2(s) \cap M_3(p(s)) \setminus (M_2(s) \cap M_3(p'(s))). \end{aligned}$$

Il s'ensuit que $M_4(s)$ est lui-même la différence de deux intervalles (ou demi-droites, ou ensembles vides), et est donc encore un intervalle, une demi-droite, ou l'ensemble vide. \square

Définition. Pour une paire (x, y) on définit $M(x, y)$ l'ensemble des M pour lesquels l'exécution de l'algorithme laisse une entrée 0 correspondant à $(xy, x+y)$, i.e., $x+y \in S_4(M)$ et $xy \in P_3(M)$. En d'autres termes, $M(x, y) = M_3(xy) \cap M_4(x+y)$, qui est une demi-droite ou un intervalle par le théorème précédent. On dit que (x, y) est une solution stable si $M(x, y)$ est une demi-droite. On dit que (x, y) est une solution fantôme si $M(x, y)$ est un intervalle fini.

Théorème (Stabilité de $(4, 13)$). La paire $(4, 13)$ est une solution stable et $M(4, 13) = [65, \infty)$.

Démonstration. On montre en deux étapes :

(i) Pour tout $M \geq 65$, $(4, 13)$ est solution. On vérifie d'abord que pour $M \geq 65$ les sommes

11, 17, 23, 27, 35, 37

appartiennent à $S_2(M)$. Ceci se voit en listant pour chacune des sommes les produits compatibles et en constatant que chacun de ces produits a au moins deux factorisations admissibles, i.e. appartient à $P_1(M)$, une fois $M \geq 65$; la justification exhaustive peut se faire numériquement. Nous listons ci-dessous, pour chaque produit compatible de chaque somme, deux factorisations admissibles avec facteurs ≤ 65 .

Somme $s = 11$.

(u, v)	uv	seconde factorisation	(u, v)	uv	seconde factorisation
2 + 9	18	3 · 6	4 + 7	28	2 · 14
3 + 8	24	2 · 12	5 + 6	30	3 · 10

Somme $s = 17$.

(u, v)	uv	seconde factorisation	(u, v)	uv	seconde factorisation
2 + 15	30	3 · 10	5 + 12	60	4 · 15
3 + 14	42	6 · 7	6 + 11	66	3 · 22
4 + 13	52	2 · 26	7 + 10	70	5 · 14
			8 + 9	72	3 · 24

Somme $s = 23$.

(u, v)	uv	seconde factorisation	(u, v)	uv	seconde factorisation
2 + 21	42	6 · 7	7 + 16	112	8 · 14
3 + 20	60	4 · 15	8 + 15	120	6 · 20
4 + 19	76	2 · 38	9 + 14	126	7 · 18
5 + 18	90	9 · 10	10 + 13	130	5 · 26
6 + 17	102	3 · 34	11 + 12	132	6 · 22

Somme $s = 27$.

(u, v)	uv	seconde factorisation	(u, v)	uv	seconde factorisation
2 + 25	50	5 · 10	8 + 19	152	4 · 38
3 + 24	72	8 · 9	9 + 18	162	6 · 27
4 + 23	92	2 · 46	10 + 17	170	34 · 5
5 + 22	110	10 · 11	11 + 16	176	8 · 22
6 + 21	126	7 · 18	12 + 15	180	9 · 20
7 + 20	140	10 · 14	13 + 14	182	26 · 7

Somme $s = 35$.

(u, v)	uv	seconde factorisation	(u, v)	uv	seconde factorisation
2 + 33	66	6 · 11	10 + 25	250	5 · 50
3 + 32	96	8 · 12	11 + 24	264	8 · 33
4 + 31	124	2 · 62	12 + 23	276	6 · 46
5 + 30	150	10 · 15	13 + 22	286	26 · 11
6 + 29	174	3 · 58	14 + 21	294	6 · 49
7 + 28	196	4 · 49	15 + 20	300	10 · 30
8 + 27	216	12 · 18	16 + 19	304	8 · 38
9 + 26	234	6 · 39	17 + 18	306	6 · 51

Somme $s = 37$.

(u, v)	uv	seconde factorisation	(u, v)	uv	seconde factorisation
$2 + 35$	70	$5 \cdot 14$	$10 + 27$	270	$9 \cdot 30$
$3 + 34$	102	$6 \cdot 17$	$11 + 26$	286	$13 \cdot 22$
$4 + 33$	132	$6 \cdot 22$	$12 + 25$	300	$15 \cdot 20$
$5 + 32$	160	$8 \cdot 20$	$13 + 24$	312	$12 \cdot 26$
$6 + 31$	186	$3 \cdot 62$	$14 + 23$	322	$7 \cdot 46$
$7 + 30$	210	$10 \cdot 21$	$15 + 22$	330	$10 \cdot 33$
$8 + 29$	232	$4 \cdot 58$	$16 + 21$	336	$12 \cdot 28$
$9 + 28$	252	$12 \cdot 21$	$17 + 20$	340	$10 \cdot 34$
			$18 + 19$	342	$6 \cdot 57$

Par le Théorème; point (b), ces 6 sommes restent dans $S_2(M)$ pour tout $M \geq 65$. Par conséquent, l'ensemble $P_3(M)$ ne contient aucun des six produits suivants :

$$\begin{aligned}
 30 &= 5 \cdot 6 = 2 \cdot 15, \\
 42 &= 2 \cdot 21 = 3 \cdot 14, \\
 60 &= 3 \cdot 20 = 4 \cdot 15, \\
 66 &= 2 \cdot 33 = 6 \cdot 11, \\
 70 &= 2 \cdot 35 = 7 \cdot 10, \\
 72 &= 3 \cdot 24 = 8 \cdot 9.
 \end{aligned}$$

En revanche, le produit

$$52 = 4 \cdot 13 = 2 \cdot 26$$

appartient à $P_3(M)$, puisque $17 \in S_2(M)$ et $28 \notin S_2(M)$ ($28 = 5 + 23$ mais $5 \cdot 23$ n'a aucune autre factorisation). Par la configuration des produits compatibles avec la somme 17 (qui sont 30, 42, 52, 60, 66, 70, 72), exactement un de ces produits est dans $P_3(M)$ (c'est 52). Ainsi $17 \in S_4(M)$, i.e., la paire (4, 13) est solution.

(ii) Pour $M \leq 64$, (4, 13) n'est pas solution. Nous affirmons que ni 19 ni 37 n'appartiennent à $S_2(M)$:

- Le produit $2 \cdot 17$ n'étant pas dans $P_1(M)$, cela implique $19 = 2 + 17 \notin S_2(M)$.
 - Le produit 186 n'appartient pas à $P_1(M)$, car seule la factorisation $186 = 6 \cdot 31$ est légale (la factorisation $186 = 3 \cdot 62$ n'est pas admissible puisque $3 + 62 > M$). Donc $6 + 31 = 37 \notin S_2(M)$.
- Supposons par contradiction que la paire (4, 13) soit solution pour $M \leq 64$. Alors $17 \in S_2(M)$ et $52 \in P_3(M)$. Or, les factorisations de 70 sont $2 \cdot 35$, $5 \cdot 14$ et $7 \cdot 10$, et comme exactement une des sommes correspondantes 37, 19, 17 est dans $S_2(M)$, on obtient $70 \in P_3(M)$.

Comme $P_3(M)$ contient deux produits $52 = 4 \cdot 13$ et $70 = 7 \cdot 10$ compatibles avec la somme 17, cela implique $17 \notin S_4(M)$.

Par conséquent, la paire (4, 13) ne peut pas être une solution pour $M \leq 64$. \square

En faisant tourner l'algorithme de la question précédente avec le code suivant (ce qui prend environ 20 minutes), on trouve que la paire (4, 13) est en réalité l'unique solution du problème pour $65 \leq M \leq 1684$. Pour $M \leq 64$, il n'y a pas de solution, et pour $M = 1685$, la paire (4, 61) forme une seconde solution. Il suffit donc de montrer que (4, 61) est stable pour conclure que le plus grand entier que l'on cherche est 1684.

python


 Télécharger le code

```

if __name__ == "__main__":
    import time
    t0, M = time.time(), 1
    while True:
        M += 1
        solutions = Denniston(M)
        print(f"[M={M}] solutions : ", solutions)
        if len(solutions) > 1:
            break
    print(f'computation time: {time.time()-t0}')
```

Théorème (Stabilité de $(4, 61)$). $(4, 61)$ est une solution stable et $M(4, 61) = [1685, \infty)$.

Démonstration. On utilise la même méthode que pour la solution $(4, 13)$. Voici le code python correspondant :

```
python  Télécharger le code

import math, numpy as np

def compatible_sums(M, p):
    out = []
    for u in range(2, int(math.isqrt(p)) + 1):
        if p % u == 0:
            v = p // u
            s = u + v
            if v >= 2 and s <= M and u != v:
                out.append(s)
    return list(set(out))

def compatible_products(M, s):
    assert s <= M
    out = []
    for u in range(2, int(s/2) + 1):
        if u != s - u:
            out.append(u * (s - u))
    return list(set(out))

def is_in_P1(M, p):
    if len(compatible_sums(M, p)) < 2:
        return False
    return True

def is_in_S2(M, s):
    for p in compatible_products(M, s):
        if not is_in_P1(M, p):
            return False
    return True

def proof_stable(M, sol):
    s_star = np.sum(sol)
    p_star = np.prod(sol)
    assert is_in_S2(M, s_star)

    sums_in_S2 = [s_star]
    prods_not_in_P3 = []
    for p in compatible_products(M, s_star):
        if p == p_star:
            continue
        new_sums = [s for s in compatible_sums(M, p) if s != s_star and
                    is_in_S2(M, s)]
        assert len(new_sums) > 0
        sums_in_S2.append(new_sums[0])
        prods_not_in_P3.append(p)

    print(f"{sorted(set(sums_in_S2))} sont dans S_2({M}), et donc dans S_2(M) pour tout M >= {M}.")
    print(f"P3(M) ne contient donc pas {sorted(set(prods_not_in_P3))}, mais contient {p_star}.")
    print(f"Donc, exactement un des produits compatibles de {s_star} est dans P3(M) (c'est {p_star}), donc {s_star} est dans S4(M)")

if __name__ == "__main__":
    proof_stable(1685, (4, 61))
```

Ce code vérifie que, pour tout $M \geq 1685$, exactement un produit compatible avec 65 appartient à $P_3(M)$; il en résulte que $65 \in S_4(M)$. \square

Notes et références

L'*énigme de la somme et du produit*, également appelée *énigme impossible* car elle semble manquer d'informations pour être résolue, est une énigme de connaissance. Elle remonte à 1969, lorsqu'Hans Freudenthal la proposa dans le *Nieuw Archief voor Wiskunde* [1, 2]. Le nom *énigme impossible* a été proposé par Martin Gardner [3]. Il existe de nombreuses variantes et déclinaisons [4, 5, 6], dont le point commun est la manière de tirer les bonnes conclusions d'une conversation étrange composée principalement de déclarations du type « Je ne sais pas » et « Maintenant je sais ».

Aujourd'hui, l'énigme de Freudenthal est célèbre dans le monde entier. Elle apparaît régulièrement dans les rubriques de casse-têtes mathématiques, et l'analyse du flux d'information dans la conversation sous-jacente est devenue un exercice classique pour les étudiants en informatique.

Sources

- [1] H Freudenthal. Problem no. 223 (formulation of the sum and product problem). *Nieuw Archief voor Wiskunde*, 3(17):152, 1969.
- [2] H Freudenthal. Solution to problem no. 223. *Nieuw Archief voor Wiskunde*, 3(18):102–106, 1970.
- [3] Martin Gardner. Mathematical games : A pride of problems, including one that is virtually impossible. *Scientific american*, 241(6):22–32, 1970.
- [4] Axel Born, Cor AJ Hurkens, and Gerhard J Woeginger. The freudenthal problem and its ramifications (part i). *Bull. EATCS*, 90:175–191, 2006.
- [5] Axel Born, Cor AJ Hurkens, and Gerhard J Woeginger. The freudenthal problem and its ramifications (part ii). *Bull. EATCS*, 91:189–204, 2007.
- [6] Axel Born, Cor AJ Hurkens, and Gerhard J Woeginger. The freudenthal problem and its ramifications (part iii). *Bull. EATCS*, 95:201–219, 2008.