

La conjecture de Syracuse

On considère un nombre entier strictement positif : u .

- Si ce nombre est pair, on le divise par 2.
- Si ce nombre est impair, on le multiplie par 3 et on ajoute 1.

Puis on recommence le procédé.

Faire quelques essais. Que constate-t-on ?

La conjecture émise par Collatz est la suivante : quel que soit le nombre u choisi, on finit toujours par atteindre le nombre 1.

La conjecture de Syracuse a été énoncée en 1937 par Lothar Collatz, mathématicien allemand. Ce problème fut soumis à l'université de Syracuse (état de New-York, USA) dans les années 1950, mais malgré les recherches, elle n'est toujours pas démontrée à ce jour. Elle a été néanmoins vérifiée pour tous les nombres entiers jusqu'à 3×10^{16} .

Remarque :

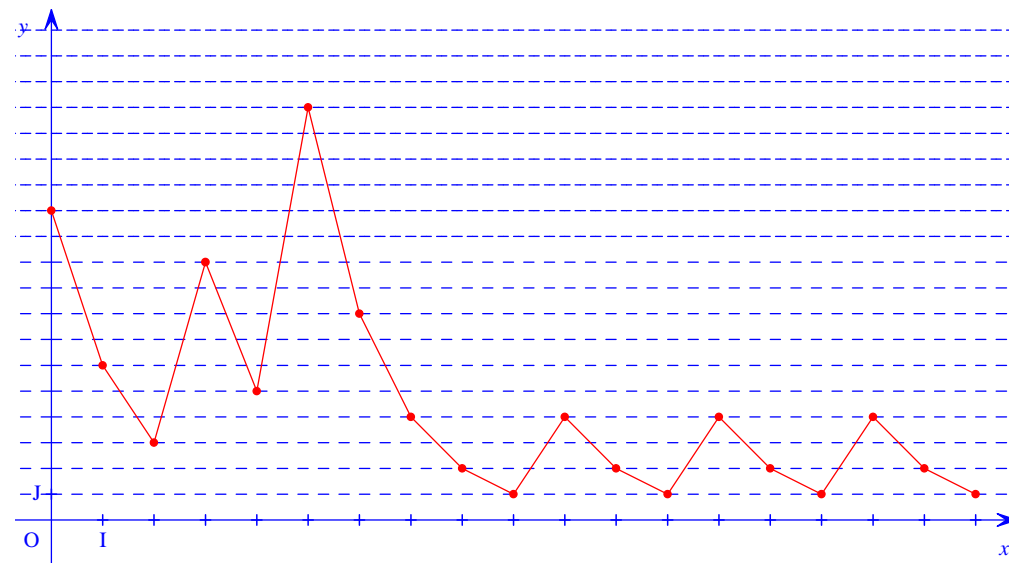
Si on continue après le nombre 1, on obtient une « boucle » : 1, 4, 2, 1, 4, 2, 1...

Représentation graphique :

Lorsqu'on représente graphiquement la suite des nombres obtenus, on peut penser à la trajectoire d'une feuille emportée par le vent, on pourra parler du « vol de la suite », et s'intéresser :

- au nombre d'étapes avant d'arriver pour la première fois au nombre 1 : « le temps de vol » ;
- au plus grand nombre atteint lors de ces étapes : « l'altitude maximale ».

Exemple avec 12 : le temps de vol est 9 et l'altitude maximale est 16.



Questions :

1°) Élaborer un algorithme permettant de déterminer le temps de vol pour une valeur de u donnée. Rédiger l'algorithme en langage naturel dans un cadre sur une seule page (il ne doit pas y avoir de page à tourner).

2°) Réaliser le programme sur la calculatrice ou sur ordinateur.

3°) Modifier le programme pour afficher en plus l'altitude maximale (réécrire l'algorithme correspondant en mettant les changements en couleur). Faire tourner le programme pour les valeurs suivantes de u : 12 ; 15 ; 24 ; 27 ; 86.

4°) Modifier le programme pour faire afficher les valeurs intermédiaires (réécrire l'algorithme correspondant).

Corrigé du DM pour le 16 mai 2012

1°) Algorithme permettant de déterminer le temps de vol pour une valeur de u donnée

L'algorithme fait intervenir une boucle « Tantque » à l'intérieur de laquelle se situe une instruction conditionnelle.

```
Variables :  $u, k$  entiers naturels

Entrée :
Saisir  $u$ 

Initialisations :
 $k$  prend la valeur 0

Traitement :
  Tantque  $u \neq 1$  Faire
    Si  $u$  est pair alors
       $u$  prend la valeur  $\frac{u}{2}$ 
    Sinon  $u$  prend la valeur  $3u + 1$ 
    FinSi
     $k$  prend la valeur  $k + 1$  (déplaçable avant le Si)
  FinTantque

Sortie :
Afficher  $k$ 
```

Cet algorithme peut être rédigé avec une multitude de variante.

Par exemple, on peut noter v la partie décimale de $\frac{u}{2}$.

On peut utiliser une alternative simple ou complète (« Si ... alors ... FinSi » ou « Si ... alors ... Sinon ... FinSi »).

La représentation en organigramme serait malaisée à cause de l'instruction conditionnelle incluse dans la boucle « Tantque ».

Cet algorithme comporte une boucle « Tantque » ; cela suppose qu'il y a un rang à partir duquel la boucle s'arrête – sinon on aurait une boucle infinie. Cela s'appuie sur la conjecture de Collatz, non démontrée à ce jour, ce qui peut être un obstacle pour un esprit très rigoureux.

La conjecture de Collatz n'a toujours pas été résolue. Paul Erdős, mathématicien d'origine tchèque du XXe siècle, pensait que les mathématiciens n'avaient pas encore les outils pour démontrer cette conjecture.

2°) Programme

a) Deux commentaires pour commencer

- On doit traduire la condition « u est pair » ;
- On peut faire afficher du texte « Le temps de vol est ».

b) À propos de la condition « u pair »

Rappelons que « u pair » signifie que « u est divisible par 2 » ou encore que « u est un multiple de 2 ».

On peut aussi dire (u pair) \Leftrightarrow ($\frac{u}{2}$ est un entier)

Dans le programme, la condition « u pair » peut se traduire de diverses manières suivant l'outil de programmation utilisé.

« u pair » peut se traduire de l'une des manières équivalentes suivantes :

- La partie entière de $\frac{u}{2}$ est égale à $\frac{u}{2}$
- La partie fractionnaire (ou décimale) de $\frac{u}{2}$ est égale à 0.
- Le reste de la division euclidienne de u par 2 est égal à 0.
- Le PGCD de u et de 2 est égal à 2.

Les deux premières manières sont ce qu'il y a de plus simple.

3°) **Algorithme modifié**

Variables : u, k, A entiers naturels

Entrée :
Saisir u

Initialisations :
 k prend la valeur 0
A prend la valeur 0

Traitement :

Tantque $u \neq 1$ **Faire**

Si u est pair alors

u prend la valeur $\frac{u}{2}$

Sinon u prend la valeur $3u + 1$

FinSi

k prend la valeur $k + 1$

Si $u > A$

alors **A prend la valeur** u

FinSi

FinTantque

Sortie :
Afficher k **et** A

u	12	15	24	27	86
Altitude maximale	16	160	16	9232	196
Temps de vol	9	17	10	111	30

On peut vérifier les résultats en faisant tourner l'algorithme à la main (un peu long... mais faisable) sauf pour 27 (car on trouve 111).

On pouvait aussi utiliser la notion de liste.

Remarque :

Lorsque $u = 31$, l'altitude maximale est de 9232 et le temps de vol est 106.
 Lorsque $u = 125$, l'altitude maximale est de 9232 et le temps de vol est 106.
 Lorsque $u = 62$, l'altitude maximale est de 9232 et le temps de vol est 107.
 Lorsque $u = 41$, l'altitude maximale est de 9232 et le temps de vol est 109.
 Lorsque $u = 82$, l'altitude maximale est de 9232 et le temps de vol est 110.
 Lorsque $u = 27$, l'altitude maximale est de 9232 et le temps de vol est 111.
 Lorsque $u = 165$, l'altitude maximale est de 9232 et le temps de vol est 111.
 Lorsque $u = 54$, l'altitude maximale est de 9232 et le temps de vol est 112.
 Lorsque $u = 55$, l'altitude maximale est de 9232 et le temps de vol est 112.
 Lorsque $u = 109$, l'altitude maximale est de 9232 et le temps de vol est 113.
 Lorsque $u = 110$, l'altitude maximale est de 9232 et le temps de vol est 113.
 Lorsque $u = 146$, l'altitude maximale est de 9232 et le temps de vol est 146.

On constate que l'altitude de 9232 revient pour de nombreux choix de la valeur de u saisie en entrée. De plus, le temps de vol augmente ici de 1 en 1, entre 106 et 116.

Critères de notations :

Respect de la forme dans la rédaction d'un algorithme en langage naturel (mise en évidence des étapes, indentations, visualisation des différents blocs d'instructions).

