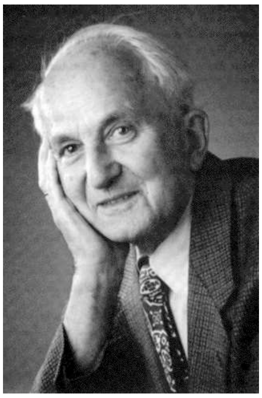
La conjecture de Syracuse



Lothar COLLATZ
06/07/1910-26/09/1990

Information

Dès 1928, Lothar Collatz s'intéressait aux itérations dans les nombres entiers. Il inventa alors le problème $3x+1$, et le présentait souvent ensuite dans ses séminaires. En 1952, lors d'une visite à Hambourg, Collatz expliqua son problème à Helmut Hasse. Ce dernier le diffusa en Amérique à l'université de Syracuse : le succès fut immédiat et la suite de Collatz prit alors le nom de suite de Syracuse. Entre temps, le mathématicien polonais Stanislas Ulam le répand dans le Laboratoire national de Los Alamos. Dans les années 1960, le problème est repris par le mathématicien Shizuo Kakutani qui le diffuse dans les universités de Yale et Chicago.

Ce problème mobilisa tant les mathématiciens durant les années 1960, en pleine guerre froide, qu'une blague courut selon laquelle ce problème faisait partie d'un complot soviétique visant à ralentir la recherche américaine. [a]

[a] D'après Wikipedia

Définition

Étant donnés un entier naturel n, on construit la suite de Syracuse du nombre n en appliquant l'algorithme suivant :

```
Variables : n (nombre entier)
début
    Répéter
        Si n pair alors
            n←n/2;
        Sinon
            n←3n+1;
fin
```

Algorithme 1 : Algorithme de Collatz

Exercice 1

Construire à la main la suite de Syracuse du nombre 17, puis celle du nombre 20. Que constate-t-on ?

Exercice 2

Écrire un algorithme permettant de donner les 100 premiers termes de la suite de Syracuse d'un nombre quelconque entré par l'utilisateur. Programmer cet algorithme sur AlgoBox[1], puis sur votre calculatrice[2].

Exercice 3

Taper le programme ci-contre sur Xcas, puis observer le résultat lorsque l'on tape :

```
syracuse(17)
```

```
syracuse(n):={
 local a,s;
 a:=n;
 s:=a;
 while (a>1)
 {if (irem(a,2)==0)
  {a:=a/2;}
  else
  {a:=3*a+1;}
  s:=s,a;}
 return(s); }
```

Exercice 4

Utiliser ce programme(en le modifiant éventuellement) pour vérifier sur Xcas votre conjecture pour tous les entiers de 1 à 1 000.

[1] $n\,\%2$ vaut 0 lorsque n est pair

[2] partDec (n) vaut 0 lorsque n est pair sur une TI et Frac n également sur une Casio.