La conjecture de Syracuse

Objectifs : Découvrir un célèbre algorithme d'arithmétique et étudier une fonction de la variable entière.

Question étudiée : On applique l'algorithme suivant à un nombre entier strictement positif : s'il est pair, on le divise par 2 ; s'il est impair, on le multiplie par 3 et on ajoute 1 au résultat. On réitère ce même processus sur le nombre obtenu.

A Expériences

1. **Tous ensemble...** Faisons tourner l'algorithme en distribuant les rôles à des groupes de trois élèves : Tiphaine teste la parité de l'entier qu'on lui donne : s'il est pair elle le transmet à Pierre, s'il est impair à Ilyesse. Quand on donne un entier pair à Pierre, il le divise par 2, le remplace par le résultat obtenu et rend le papier à Tiphaine qui recommence (si l'entier est pair ...). Quand on donne un entier impair à Ilyesse, il le multiplie par 3 et ajoute 1 au résultat le remplace par le résultat obtenu et rend le papier à Tiphaine qui recommence (si l'entier est pair ...). Le professeur écrit un entier strictement positif sur un papier et le donne à Tiphaine. Observons ...

2. **...ou chacun pour soi.** Recopier et remplir le tableau suivant :

Nombre de départ	Itérations successives														
3															
32															
votre choix															

Que semble-t-il se passer ? Faire une conjecture.

Cette conjecture dont l'origine, vers 1950 reste confuse porte une grande variété de noms provenant de mathématiciens l'ayant étudiée ou fait connaître : problème de Collatz, problème de Kakutani, problème de l'algorithme de Hasse, problème d'Ulam. Le nom de conjecture de Syracuse est lié à l'université de Syracuse, aux Etats-Unis, où le problème fut étudié.(...) S. Kakutani fit circuler le problème et raconte : "Pendant un mois, tout le monde à l'Université de Yale travailla dessus, sans résultat. Un phénomène semblable se produisit à l'Université de Chicago. Cette énigme, pensaient certains, avait été avancée par le KGB pour ralentir la recherche mathématique aux Etats-Unis." (D'après J-P Delahaye)

Cette conjecture a été vérifiée jusqu'au nombre $3{,}2 \times 10^{16}$. Vous pouvez essayer de la démontrer mais sachez qu'un bon nombre de grands mathématiciens ont déjà essayé ... sans succès !

B Etude "à la main" de la fonction de base

Nous allons étudier la fonction f définie par le processus suivant :
L'entrée est un entier strictement positif.
S'il est pair, le diviser par 2.
S'il est impair, le multiplier par 3 et ajouter 1 au résultat.
Le résultat final est donné en sortie.

1. Quel est l'ensemble de définition de f ?
2. Remplir le tableau de valeurs donnant $f(n)$ pour les entiers n de 1 à 20.
3. Peut-on entrer *une* formule dans la calculatrice pour calculer $f(n)$?
4. Tracer à la main dans un repère la représentation graphique de f pour les entiers allant de 1 à 20.
5. En utilisant le tableau de valeurs, dire si 1 a des antécédents par la fonction f.
6. En utilisant le graphique, dire si 4 a des antécédents par la fonction f.

Pour aller plus loin : Trouver par le calcul *tous* les antécédents de 58 par la fonction f. Même question pour les antécédents de 13.

C Programmation de la fonction de base

1. **Comment tester la parité d'un nombre**

a. **Avec la fonction "partie entière" E.** Par exemple E(2,75)=2 ; E(7,25)=7 etc...
A quelle condition sur le nombre x a-t-on $\mathrm{E}(x) = x$?
A quelle condition sur le nombre entier naturel n a-t-on $\mathrm{E}(n/2) = (n/2)$?

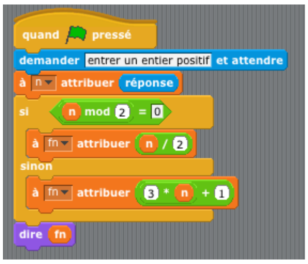
b. **Avec la fonction mod** qui donne le reste dans une division euclidienne.
Par exemple mod(50,7)=1 car le reste de la division de 50 par 7 est 1.
Si n est un entier naturel quelles valeurs peut prendre mod(n,2) ? Préciser dans quel cas.

2. Compléter l'algorithme suivant écrit en langage algorithmique.

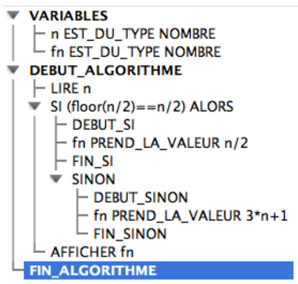
Algorithme :
Variables :
n, fn : entiers
Entrées :
Saisir n
Traitement :
Si ..... **alors** $\longleftarrow$ *Utiliser 1a ou 1b*
fn prend la valeur .....
Sinon
fn prend la valeur .....
FinSi
Sorties :
Afficher fn

3. Programmons cet algorithme.

Script Scratch



Programme Algobox



Remarquer que dans ces deux programmes, le test de la parité d'un entier est adapté en fonction des fonctionnalités du langage.

D Itération du processus

Nous devons réaliser une boucle avec comme condition d'arrêt l'obtention de 1.

1. Compléter l'algorithme suivant en utilisant celui de la partie B :

Algorithme :
Variables :
n, fn : entiers
Entrées :
Saisir n
Traitement et sorties :
Répéter
⋮
Afficher fn
n prend la valeur fn
Jusqu'à n=1

Programme Algobox

```
VARIABLES
  n EST_DU_TYPE NOMBRE
  fn EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  LIRE n
  TANT_QUE (n!=1) FAIRE
    DEBUT_TANT_QUE
    SI (floor(n/2)==n/2) ALORS
      DEBUT_SI
      fn PREND_LA_VALEUR n/2
      FIN_SI
      SINON
        DEBUT_SINON
        fn PREND_LA_VALEUR 3*n+1
        FIN_SINON
    AFFICHER fn
    n PREND_LA_VALEUR fn
    FIN_TANT_QUE
FIN_ALGORITHME
```

2. **Pour aller plus loin :**
Mesures de quelques *vols*.
On appelle *vol* la suite obtenue à partir d'un entier avant "d'atterrir" à 1.
On appelle *durée du vol* le nombre d'étapes du vol.
On appelle enfin *altitude du vol* le plus grand entier obtenu dans la liste.
Dans le script Scratch ci-contre, identifier parmi les variables p et q laquelle est un compteur qui détermine la durée du vol et laquelle mémorise l'altitude.
Compléter alors les messages de sorties.

Programmer ce script et compléter alors le tableau suivant.

Entier n	2	3	4	5	9	10	14	16	18	24	27
Durée du vol		7			19						111
Altitude du vol		16			52						9232