

La conjecture de Syracuse

Prenons un entier naturel n . On construit la suite de Syracuse de n en appliquant le procédé suivant:

- si n est pair, on le divise par 2;
- sinon on le multiplie par 3 et on ajoute 1;
- on recommence avec le nombre obtenu.

On arrête le procédé lorsqu'on finit par obtenir 1.

Par exemple, si on choisit 13 comme nombre de départ, on obtient la suite : 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

La [conjecture de Syracuse](#) postule qu'on atteindra toujours 1 quel que soit le nombre n de départ.

Pour chaque suite de Syracuse, de terme de départ n , on définit:

- **le temps de vol**: c'est le nombre d'itérations pour obtenir 1;
- **le temps de vol en altitude**: c'est le nombre d'itérations avant de passer en dessous de n (pour la première fois);
- **l'altitude maximale**: c'est la valeur maximale de la suite.

Par exemple, pour la suite de Syracuse de 13, le temps de vol est 9, le temps de vol en altitude est 3 et l'altitude maximale est 40.

Objectif

L'objectif de ce projet est de construire un *module* contenant différentes fonctions permettant d'étudier ces suites.

En particulier, le module contiendra *a minima* les fonctions suivantes:

- `suivant` : renvoie le successeur d'un nombre entier dans le procédé de Syracuse;
- `syracuse` : renvoie la suite de Syracuse;
- `temps_de_vol`, `altitude_max`, et `temps_vol_altitude` dont les noms sont j'espère assez parlants...
- `affiche_syracuse` : qui doit afficher le graphique de la suite (utiliser pour cela le module `matplotlib.pyplot`).

Partie I

À la main, écrire les suites de Syracuse de 15, 64, et 97. Déterminer pour chacune les temps de vol, temps de vol en altitude et altitude maximale.

Ces exemples serviront plus tard de jeu de tests pour les fonctions à écrire.

Partie II

1. Spécifier chacune des fonctions demandées.
2. Écrire les jeux de tests pour chaque fonction (exceptée `affiche_syracuse`).

Partie III

Écrire les fonctions.