

Bases de la programmation impérative

Ensimag 1A

1. Syracuse

On se propose de ré-implémenter le calcul du nombre d'étapes pour que la suite de Syracuse atteigne 1. Pour rappel : si u_n est pair, $u_{n+1} = u_n/2$ sinon $u_{n+1} = 3 * u_n + 1$.

1.1. Implémentez une fonction récursive `etapes(val_init)`.

1.2. Exécutez pas à pas `etapes(10)`.

2. Fibonacci

On considère le code suivant :

```
1 def fibonacci(rang):
2     """
3     renvoie le terme de rang donne de la suite de Fibonacci.
4     """
5     print("rang", rang)
6     if rang <= 1:
7         return rang
8     else:
9         return fibonacci(rang - 1) + fibonacci(rang - 2)
```

Qu'affiche `fibonacci(4)` ?

3. Exponentiation rapide

On souhaite calculer x^y où x est d'un type quelconque supportant la multiplication et y un entier > 0 . En remarquant que pour y pair on a $x^y = (x^{\frac{y}{2}})^2$, proposez une fonction récursive calculant très rapidement x^y .

4. Dichotomie

4.1. À l'aide des tranches de tableaux, implémentez une recherche dichotomique dans un tableau trié.

4.2. Que penser des performances ?