

## – Algorithme d'Euclide –

1. Écrire une fonction d'entête `euclide(a,b)` calculant de façon itérative le pgcd de deux entiers `a` et `b`.
2. Écrire une version récursive de la fonction précédente. Comparer sa complexité à celle de la fonction précédente.
3. Utiliser les questions précédentes pour écrire une fonction calculant le pgcd de trois nombres.
4. Calculer le pgcd des familles suivantes : (258, 154), (25, 48, 963), (78954, 123, 12) et (94, 155).

## – Une suite récurrente –

On considère la suite récurrente ci dessous :

$$\begin{cases} u_0 = 1 \\ \forall n \geq 0, u_{n+1} = \frac{u_n + \frac{2}{u_n}}{2} \end{cases} .$$

1. Démontrer que la suite  $(u_n)_n$  est bien définie.
2. Écrire une fonction itérative `u_iter(n)` calculant le  $n$ -ième terme de la suite  $(u_n)_n$ . Quelle est sa complexité.
3. On considère la fonction `python` suivante.

```
def u_rec(n):
    if n == 0:
        return 1
    else :
        return (u_rec(n - 1) + 2/ u_rec(n - 1)) / 2
```

- (a) Que renvoie un appel à cette fonction ?
  - (b) Quelle est sa complexité ? Que remarque-t-on ?
  - (c) Proposer une solution au problème remarqué dans la question précédente.
4. Proposer une estimation de la limite de la suite  $(u_n)_n$  à l'aide de l'outil informatique. Comment retrouver mathématiquement ce résultat ?

## – Manipulations de listes et piles –

1. (a) Écrire une fonction récursive d'entête `positifs(L)` renvoyant la liste des termes positifs d'une liste `L`. Quel est l'inconvénient de cette fonction en terme de complexité ?
  - (b) Proposer une solution au problème soulevé dans la question précédente.
2. Reprendre le TP sur les piles. Quelles fonctions et méthodes y étant présentées pourraient être réécrites de façon récursive ? Comparer les complexités.