

1. Version récursive de l'élévation à la puissance par multiplications successives.
2. Calcul de  $n!$  (version itérative puis récursive).
3. Version récursive du calcul du pgcd.
4. Calcul du terme  $u_n$  de la suite de Fibonacci, définie par :

$$\begin{aligned}u_0 &= 0 \\u_1 &= 1 \\u_{n+1} &= u_n + u_{n-1}\end{aligned}$$

Donner une version itérative et une version récursive.

5. Soit l'algorithme du tri rapide donné ci-dessous.

```
def partition (t,g,d):
    m = g + 1
    while m <= d:
        if t[m] < t[g]:
            m = m + 1
        else:
            echange(t,m,d)
            d = d - 1
    echange(t,g,d)
    return d

def tri_rapide (t,g,d):
    if g < d:
        m = partition(t,g,d)
        tri_rapide(t,g,m-1)
        tri_rapide(t,m+1,d)

def tri (t):
    tri_rapide(t,0,len(t)-1)
```

Remarque : `partition` est similaire à Dijkstra avec 2 couleurs

- (a) Faire tourner sur un exemple cet algorithme.
- (b) Expliquer son principe.
- (c) Donner un exemple pour lequel cet algorithme n'est pas efficace.
- (d) Donnez la complexité de cet algorithme.