**Analyse des algorithmes récursifs**

## Les principes de la récursivité :

Fct (n) -> Fct-n-1) : Formule de récurrence avec une valeur d’arrêt.

Ex : Fct (**0**) -> val

Ex : f (n) = 3 f (n-1) en sachant que f (0) = 1 (On peut calculer f de n’importe quelle valeur car on connait une des valeurs de la fonction, on dit qu’on connait au moins une condition terminale.

On commence par faire un test pour savoir quand on s’arrête (ex : quand n = 0 ou n = 1) sinon, on renvoie le résultat de la fonction(n).

def factorielle(n) :

          if (n==0) or (n==1) :

                    return 1

          else :

                    return n\*factorielle(n-1)

print(factorielle(7))

**Fonctionnement de la récursivité :**

Le programme va empiler des appels pour calculer n\*factorielle(n-1) :

5\*factorielle (4) (il ne connait pas factorielle de 4 donc il doit la calculer en rappelant la fonction)

4\*factorielle (3)

3\*factorielle (2)

2\*factorielle (1)

factorielle (1)

Programme permettant de déplacer n éléments de T1 à T2 en passant par T3 :

T1 = [3,2,1]

T2 = []

T3 = []

def Hanoi(n, TA, TB, TC, NomA, NomB, NomC):

    if n==1:

        TB.append(TA[-1])

        del TA[-1]

        print("Déplacer {} de {} vers {}".format(TB[-1], NomA, NomB))

    else:

        Hanoi(n-1, TA, TC, TB, NomA, NomC, NomB)

        Hanoi(1, TA, TB, TC, NomA, NomB, NomC)

        Hanoi(n-1, TC, TB, TA, NomC, NomB, NomA)

Hanoi(len(T1), T1, T2, T3, "T1", "T2", "T3")