

## Feuille de TD n° 4bis : complexité

**Exercice 1.** L'algorithme suivant prend en entrée deux entiers. Que fait-il ? Quelle est sa complexité (compter le nombre d'opérations élémentaires) ? Minorer également la le nombre d'opérations de sorte à minorer la complexité.

```
1 def f(m, n):
2     r = 0
3     while n > 0:
4         if n%2 == 1:
5             r += m
6         m *= 2
7         n //= 2
8     return r
```

**Exercice 2.** Que fait la fonction maxtab suivante ? Quelle est sa complexité ? Comment pourrait-on améliorer le code ?

```
1 def maxdic(tab, begin, end):
2     if begin + 1 < end:
3         mid = (begin + end) // 2
4         maxl = maxdic(tab, begin, mid)
5         maxr = maxdic(tab, mid, end)
6         if maxl > maxr:
7             return maxl
8         else:
9             return maxr
10    else:
11        return tab[begin]
12
13 def maxtab(tab):
14    return maxdic(tab, 0, len(tab))
```

**Exercice 3.** Soit  $f$  une fonction prenant une chaîne de caractères composée de 0 et de 1, et retournant un entier. On suppose que sa complexité est en  $O(1)$ . On définit la fonction  $g$  comme ci-dessous. Que fait  $g$  ? Comment pourrait-on réécrire le code ?

```
1 def h(f, v, end_str, n):
2     if n == 0:
3         if f(end_str) == v:
4             return end_str
5         else:
6             return False
7     else:
8         tab0 = h(f, v, '0' + end_str, n - 1)
9         tab1 = h(f, v, '1' + end_str, n - 1)
10        if tab0 != False:
11            return tab0
12        else:
13            return tab1
14
15 def g(f, v, n):
16    return h(f, v, '', n)
```