

## 22. Complexité en mémoire d'un algorithme

**Exercice 1.** On considère les algorithmes suivants :

```

Fonction: fib( $n$ )
Début
   $l \leftarrow [1, 1]$ 
  Pour  $k$  allant de 2 à  $n$  faire
     $l \leftarrow l + [l[k - 2] + l[k - 1]]$ 
  FinPour
  Renvoyer  $l[n]$ 
Fin
    
```

```

Fonction: smartfib( $n$ )
Début
  Si  $n < 2$  alors
    Renvoyer 1
  sinon
     $l \leftarrow [1, 1, 2]$ 
    Pour  $k$  allant de 3 à  $n$  faire
       $l[0] \leftarrow l[1]$ 
       $l[1] \leftarrow l[2]$ 
       $l[2] \leftarrow l[0] + l[1]$ 
    FinPour
    Renvoyer  $l[2]$ 
  FinSi
Fin
    
```

1. Déterminer la sortie de chacun des algorithmes.
2. Déterminer la longueur des listes dans chacun des algorithmes.

**Exercice 2.** On considère l'algorithme suivant :

```

Fonction: f( $n$ )
Début
   $l \leftarrow [1, 1]$ 
  Pour  $i$  allant de 0 à  $n - 2$  faire
    Pour  $j$  allant de 0 à  $i$  faire
       $l[j] \leftarrow l[j] + l[j + 1]$ 
    FinPour
     $l \leftarrow [1] + l$ 
  FinPour
  Renvoyer  $l$ 
Fin
    
```

1. Déterminer la sortie de l'algorithme.
2. Déterminer le nombre d'additions effectuées.
3. Déterminer la taille de la liste de sortie.

**Exercice 3.** On représente un polynôme  $P$  de degré  $n$  par la liste de ses coefficients.

1. Écrire un algorithme permettant de calculer la dérivée  $k$ -ième de  $P$ .
2. Déterminer le nombre d'opérations effectuées ainsi que la taille de la liste de sortie.