

## Complexité des algorithmes

Calculer la complexité des algorithmes suivants :

```
1.
Pour i de 2 à n faire
  k ← i-1;
  x ← T[i];
  Tant que (T[k] > x et k > 0) faire
    T[k+1] ← T[k];
    k ← k-1;
  Fin TQ;
  T[k+1] ← x;
Fin Pour;
```

```
4.
Fonction Fib( n : entier) : entier;
Début
  Si (n < 2) Alors
    Fib ← 1;
  Sinon
    Fib ← Fib(n - 1)+Fib(n - 2);
  Fin Si;
Fin;
```

```
2.
i ← n;
S ← 0;
Tant que (i > 0) faire
  j ← 2*i;
  Tant que (j > 1) faire
    S ← S+(j-i)* (S+1);
    j ← j-1;
  Fin TQ;
  i ← i div 2;
Fin TQ;
```

```
3.
i ← 1;
j ← 0;
Pour k de 1 à n faire
  j ← i+j;
  i ← j-i;
Fin Pour;
```

Que fait cet algorithme sachant que le résultat est dans j ?

```
5.
P ← 1;
Pour I de 1 à n faire
  J ← 1;
  K ← 1;
  Tant que (K ≤ n) faire
    P ← P * (K + J);
    K ← K + 1;
  Si (K > n) Alors
    J ← J + 1;
    Si (J ≤ n) Alors
      K ← 1;
    Fin Si;
  Fin Si;
Fin TQ;
Fin Pour;
```

```

6.
i ← 1;
Tant que (i < n) faire
    | j ← 1;
    | Tant que (j < 2*n) faire
    |     | j ← j*2;
    |     Fin TQ;
    | i ← i+1;
Fin TQ;

```

```

8.
i ← 1;
j ← 1;
Tant que (i < n) faire
    | Si (j < n) Alors
    |     | j ← j * 2;
    |     Sinon
    |         | j ← 1;
    |         Fin Si;
    | i ← i + 1;
Fin TQ;

```

```

7.
Var x : entier ;
Fonction f( i, j, k : entier) : entier;
Début
    | Si (k+j = i) Alors
    |     | f ← ((i-j) div k) + 1;
    |     Sinon
    |         | x ← f(i, j+1, k-2);
    |         | f ← f(i+1, j+x, k-2);
    |         Fin Si;
Fin;

```

```

9.
Procédure F( x, y, z : réel);
Début
    | y ← 2 * z;
    | Si (x > x/(y - z)) Alors
    |     | x ← x - 2;
    |     | y ← y/4;
    |     | z ← z/5;
    |     | F(x, y, z);
    | Fin Si;
Fin;

```

\*\*\* Bonne chance \*\*\*