

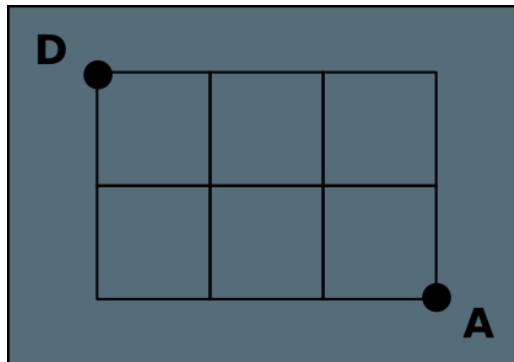
Principes de la programmation dynamique

1. Un premier exmple débranché

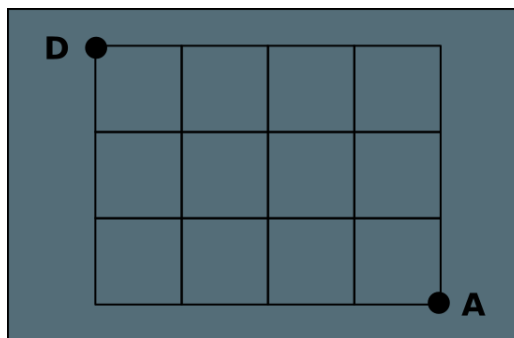
? Parcours sur une grille

Enoncé

1. Combien y-a-t'il de chemin menant du point D au point A sur le graphique suivant, en ne se déplaçant à chaque pas que vers la droite ou vers le bas ?



2. Combien y-a-t'il de chemin menant du point D au point A sur le graphique suivant, en ne se déplaçant à chaque pas que vers la droite ou vers le bas ?



Solution

A venir !

2. La suite de Fibonacci

La **suite de Fibonacci** est une suite définie par une récurrence d'ordre 2 de la manière suivante, :

$$\left\{ \begin{array}{l} F_0 = 0 \\ F_1 = 1 \\ F_{n+2} = F_{n+1} + F_n \quad \forall n \in \mathbb{N} \end{array} \right.$$

? Calculer

Enoncé

Calculer les 10 premiers termes de la suite de Fibonacci.

Solution

Les 10 premiers termes sont : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

i Notation

On notera $F(n)$ le nombre de la suite de Fibonacci de rang n . Par exemple $F(0) = 0$ et $F(6) = 13$.

Algorithmiquement parlant, la suite de Fibonacci étant une suite définie par récurrence, nous serions tentés de créer une fonction récursive pour calculer les termes $F(n)$ de la suite. Pour ce faire, nous pourrions utiliser la fonction suivante :

```
1 def fibo(n) :  
2     if n == 0 :  
3         return 0  
4     elif n == 1 :  
5         return 1  
6     else :  
7         return fibo(n-1) + fibo(n-2)
```

Est-ce un choix judicieux ?

? Tester et voir les limites

Enoncé

1. Tester la fonction `fibonacci` avec le code suivant :

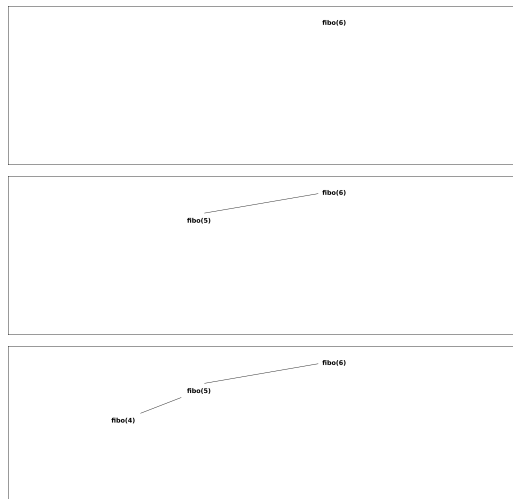
```
import time
for n in range(40) :
    start = time.perf_counter()
    print(f"fibonacci({n}) = {fibonacci(n)}", end=" ")
    end = time.perf_counter()
    print(f" Temps : {end - start}")
```

Que constate-t-on ?

2. Réaliser un schéma de la pile d'appels récursif effectués lors de l'exécution de `fibonacci(6)` .

Solutions

1. Le temps d'exécution croît de manière exponentielle.
2. On a la construction suivante :



[Previous](#) [Next](#)