La programmation dynamique

Table des matières

I) Définition

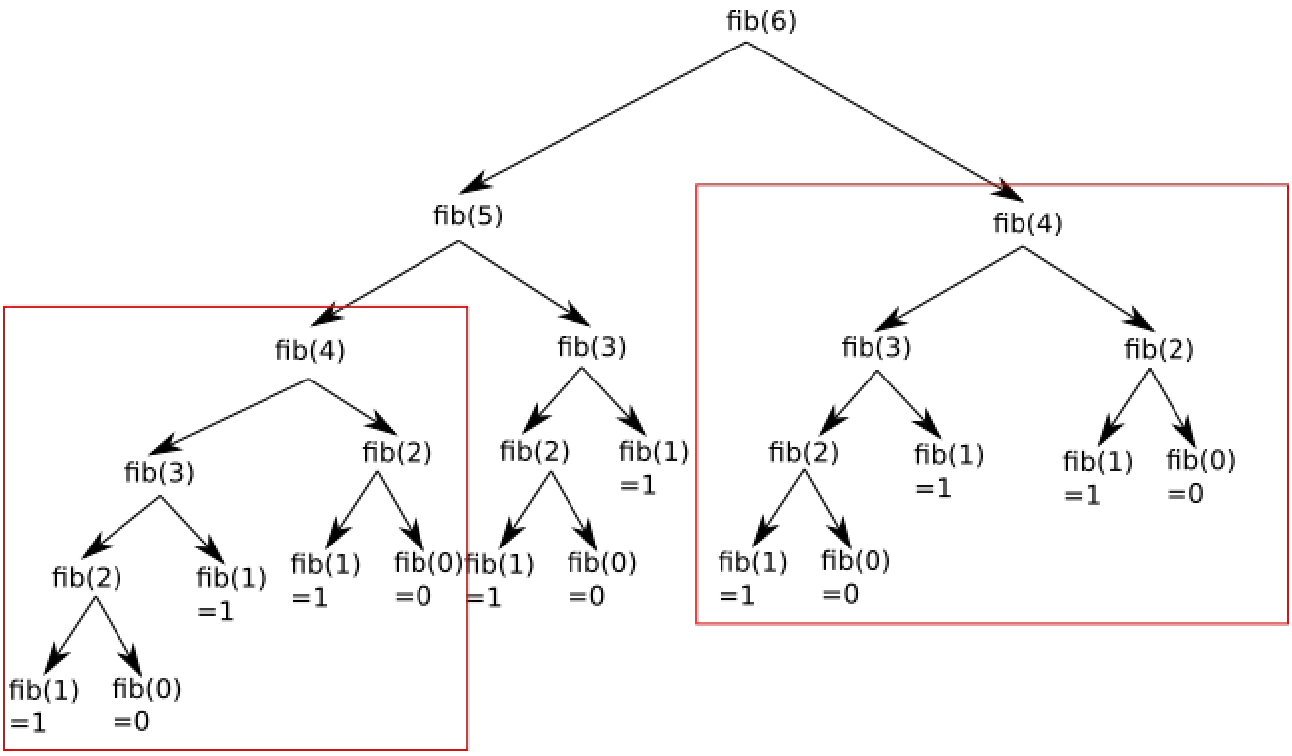
La programmation dynamique est un paradigme de programmation qui vise, comme la méthode diviser pour régner, à résoudre des problèmes en combinant des solutions de sous-problèmes. Mais à la différence de la méthode diviser pour régner, la programmation dynamique s'applique quand les sous-problèmes se recoupent. Un algorithme de programmation dynamique résout donc chaque sous-problème une seule fois et mémorise sa réponse dans un tableau, évitant ainsi le recalcul de la solution chaque fois qu'il résout chaque sous-problème

II) Exemple d'application

Un exemple simple d'utilisation de la programmation dynamique est la suite de Fibonacci. Pour rappel, la suite de Fibonacci se définit de la manière suivante :

$$F_0=0 \quad F_1=1 \qquad\qquad F_{n+2}=F_{n+1}+F_n$$

Or, on voit sur le schéma suivant que lors du calcul de F_6 , On calcule 2 fois F_4 , 3 fois F_3 et 5 fois F_2 :



On peut donc optimiser ce calcul en stockant les valeurs de F_4, F_3 et F_2. Sur F_6 le temps de calcul ne sera pas trop impacté, mais sur F_{543251}, cela fait une grosse différence.

Voici l'algorithme de calcul de la suite de Fibonacci classique :

```
def fibo(n):
  if n <= 1:
    return n
  return fibo(n-1) + fibo(n-2)
```

Et le voici en programmation dynamique :

```
def fibo_dynamique(n):
  def fibo(n,m):
  if n==0 or n==1:
    m[n]=n
    return n
  elif m[n]>0:
    return m[n]
  else:
    m[n]=fib_mem_c(n-1,m) + fib_mem_c(n-2,m)
    return m[n]

  mem = [0]*(n+1)
  return fibo(n,mem)
```