

Programación Dinámica

Miguel Raggi

Algoritmos

Centro de Ciencias Matemáticas
UNAM

9 de abril de 2019

Índice:

1 Programación Dinámica

- Motivación: Fibonacci
- Introducción

Índice:

1 Programación Dinámica

- Motivación: Fibonacci
- Introducción

Recursión en Fibonacci

- El código **recursivo** para encontrar Fibonacci es:

```
def Fibonacci(n):  
    if (n == 0):  
        return 0  
    if (n == 1):  
        return 1  
    return Fibonacci(n-1) + Fibonacci(n-2)
```

- Sin embargo este es muy muy lento. ¿Por qué?

Recursión en Fibonacci

- El código **recursivo** para encontrar Fibonacci es:

```
def Fibonacci(n):  
    if (n == 0):  
        return 0  
    if (n == 1):  
        return 1  
    return Fibonacci(n-1) + Fibonacci(n-2)
```

- Sin embargo este es muy muy lento. ¿Por qué?
- ¡Está **calculando** muchísimas veces lo mismo!

Recursión en Fibonacci

- El código **recursivo** para encontrar Fibonacci es:

```
def Fibonacci(n):  
    if (n == 0):  
        return 0  
    if (n == 1):  
        return 1  
    return Fibonacci(n-1) + Fibonacci(n-2)
```

- Sin embargo este es muy muy lento. ¿Por qué?
- ¡Está **calculando** muchísimas veces lo mismo!
- ¿Cómo podríamos hacerlo mejor utilizando un poco más de memoria?

Recursión en Fibonacci

- El código **recursivo** para encontrar Fibonacci es:

```
def Fibonacci(n):  
    if (n == 0):  
        return 0  
    if (n == 1):  
        return 1  
    return Fibonacci(n-1) + Fibonacci(n-2)
```

- Sin embargo este es muy muy lento. ¿Por qué?
- ¡Está **calculando** muchísimas veces lo mismo!
- ¿Cómo podríamos hacerlo mejor utilizando un poco más de memoria?
- Pues si guardamos todas las soluciones hasta el momento.

Programación Dinámica en Fibonacci

Consideremos el siguiente código:

```
def Fibonacci(n):  
    if (n == 0):  
        return 0  
    L = [0,1]  
    for i in range(2,n):  
        L.append(L[-1] + L[-2])  
    return L[-1]+L[-2]
```

También calcula Fibonacci, pero utilizando **programación dinámica**.
Veamos qué tanto se acelera.

Programación Dinámica

- Programación Dinámica es el arte de sacrificar espacio o memoria para obtener rapidez.

Programación Dinámica

- Programación Dinámica es el arte de sacrificar espacio o memoria para obtener rapidez.
- En términos generales, le llaman programación dinámica a crear una tabla gigante e ir la actualizando poco a poco con ciertas reglas hasta que ya no se modifique más.

Programación Dinámica

- Programación Dinámica es el arte de sacrificar espacio o memoria para obtener rapidez.
- En términos generales, le llaman programación dinámica a crear una tabla gigante e ir la actualizando poco a poco con ciertas reglas hasta que ya no se modifique más.
- Al igual que **divide y vencerás**,

Programación Dinámica

- Programación Dinámica es el arte de sacrificar espacio o memoria para obtener rapidez.
- En términos generales, le llaman programación dinámica a crear una tabla gigante e ir la actualizando poco a poco con ciertas reglas hasta que ya no se modifique más.
- Al igual que **divide y vencerás**, programación dinámica utiliza soluciones de subproblemas para resolver el problema grande.

Programación Dinámica

- Programación Dinámica es el arte de sacrificar espacio o memoria para obtener rapidez.
- En términos generales, le llaman programación dinámica a crear una tabla gigante e ir la actualizando poco a poco con ciertas reglas hasta que ya no se modifique más.
- Al igual que **divide y vencerás**, programación dinámica utiliza soluciones de subproblemas para resolver el problema grande.
- Se utiliza cuando las soluciones de los problemas pequeños se intersectan (como en el caso de Fibonacci).

0-1 knapsack

Problema

Puedes cargar máximo K kilos. Tienes n artículos. Cada artículo i tiene un valor v_i y un peso p_i . Cuánto es el máximo valor total que puedo cargar?

<https://practice.geeksforgeeks.org/problems/0-1-knapsack-problem/0>

Un problema parecido aquí:

<https://www.hackerrank.com/challenges/unbounded-knapsack>