QuickLook: Dynamic Programming

CS 196 – 25 DotStar [Lecture A02]

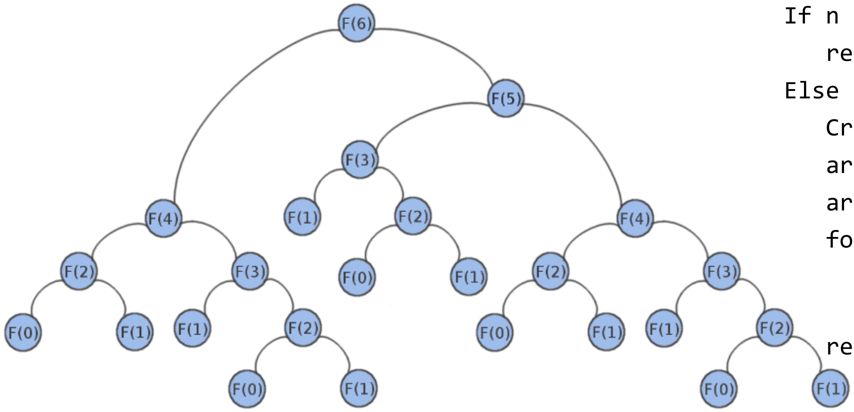
Dynamic Programming involves optimizing a problem by solving it recursively and then making it more efficient by using a data structure to avoid doing the same operation multiple times.

Nth Fibonacci

Trivial Solution

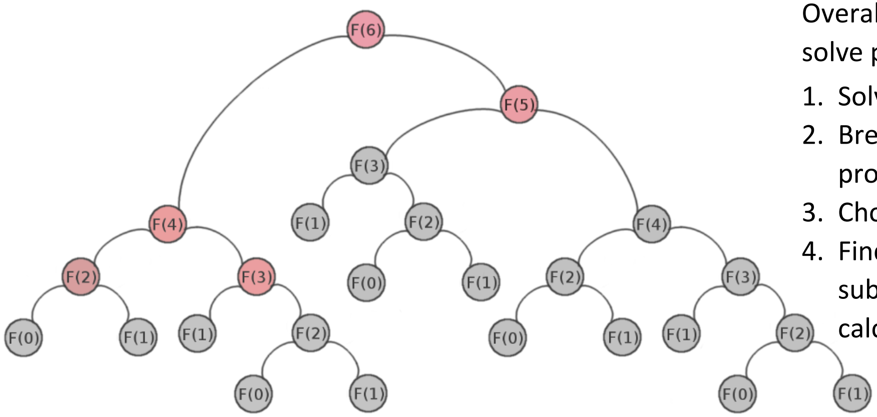
```
GetNthFib(n):
   If n is 0
     return 0
   If n is 1
     return 1
   Else
     return GetNthFib(n-1) +
       GetNthFib(n-2)
```

This works, but it's got one pretty glaring issue – we compute the same value multiple times.


F(6)
F(5)
F(3)
F(4) F(1) F(2) F(4)
F(2) F(3) F(0) F(1) F(2) F(3)
F(0) F(1) F(1) F(2) F(0) F(1) F(1) F(2)
F(0) F(1) F(0) F(1)

Better Solution

If we stored each number in an array, we wouldn't need to recalculate it multiple times. This concept of storing values into a data structure is known as **memoization**.


F(6)
F(5)
F(3)
F(4) F(1) F(2) F(4)
F(2) F(3) F(0) F(1) F(2) F(3)
F(0) F(1) F(1) F(2) F(0) F(1) F(1) F(2)
F(0) F(1) F(0) F(1)

Consider this code:

```
GetNthFib(n):
   If n is 0
      return 0
   If n is 1
      return 1
   Else
      Create an array of size n
      array[0] = 0
      array[1] = 1
      for i from 2 to (n-1)
         array[i] = array[i-1] +
          array[i-2]
      return array[n-1] + array[n-2]
```

By using the array, we solve for fewer numbers and our program is more efficient.

Wrapping Up

Overall using dynamic programming to solve problems uses the following steps

1. Solve your problem recursively
2. Break down your recursion into sub-problems
3. Choose a data structure to save time
4. Find out any dependencies between sub-problems and find a good order to calculate them

Ok, now it's your turn!

Given the following denominations

Dollar(100 cents)	Quarter(25 cents)
Dime(10 cents)	Nickel(5 cents)
Penny(1 cent)	

Find the number of ways to make changes for n cents using dynamic programming.

The first private post on piazza with the correct answer will get a Starbucks Gift Card. You can use any programming language, or pseudocode.

Find this handout online at www.akmodi.com/docs/DynamicProgramming.pdf (link is case sensitive)
Feel free to contact the lecturers: Mark Miller [mrmillr3@illinois.edu]