TCSS 343 - Week 4

Jake McKenzie

August 18, 2018

Dynamic Programming

"An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

...

Richard Bellman's **Principle of Optimality**

"What we choose means more than what was handed to us by chance."

...

Ada Palmer

" If 'dynamic programming' didn't have such a cool name, it would be known as 'populating a table'".

...

Mark Dominus

Today we're going to look at dynamic programming. Below are three implementations of the fibonacci algorithm that I wrote in python. I want you to draw the "tree" for each then reflecting on how the "bottom up" apprach is different from the other two? (Hint: They are all trees but the bottom up approach is something else you know of also. This is a key idea in idea in understanding dynamic programming)

```python
# recusive fibonacci
def F(n):
    if n == 0: return 0
    elif n == 1: return 1
    else: return F(n-1) + F(n-2)
```

```python
# "top down" memoized recursive fibonacci
memo = {}
def Fib(n):
    if n in memo: return memo[n]
    if n == 0: f = 0
    elif n == 1: f = 1
    else: f = Fib(n-1) + Fib(n-2)
    memo[n] = f
    return f
```

```python
# "bottom up" iterative fibonacci
def fib(n):
    fn = [0,1]
    for i in range(n >> 1):
        fn[0] += fn[1]
        fn[1] += fn[0]
    return fn[n % 2]
```