

# 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

1. Today we're going to explore 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 reflect on how the “bottom up” approach is different from the other two? (Hint: They are all trees but also different types of trees. This is a key insight in my opinion in idea in understanding dynamic programming)

```
1  # recursive fibonacci
2  def F(n):
3      if n == 0: return 0
4      elif n == 1: return 1
5      else: return F(n-1) + F(n-2)

7  # "top down" memoized recursive fibonacci
8  memo = {}
9  def Fib(n):
10     if n in memo: return memo[n]
11     if n == 0: f = 0
12     elif n == 1: f = 1
13     else: f = Fib(n-1) + Fib(n-2)
14     memo[n] = f
15     return f

17 # "bottom up" iterative fibonacci
18 def fib(n):
19     fn = [0,1]
20     for i in range(n >> 1):
21         fn[0] += fn[1]
22         fn[1] += fn[0]
23     return fn[n % 2]
```