



# Dynamic Programming Algorithms

# Learning Objectives

---

1. Define Dynamic Programming
2. Implement Simple Dynamic Programming Problems



# Dynamic Programming

---

Dynamic programming algorithms solve complex problems by breaking them down into simpler, overlapping subproblems and solving each subproblem only once, storing the results for later use.



# Factorial

*Recursive or Iterative*

$$1! = 1$$

$$2! = 2 * 1!$$

$$3! = 3 * 2!$$

*Dynamic Programming*

$$F(n) = n * F(n-1)$$

	0	1	2	3	4	5
F(n)						



# Factorial Implementation

```
1 unsigned long factorial(unsigned long n)
2 {
3     if (n == 0) {
4         return 1;
5     }
6     return n * fac(n - 1);
7 }
8
9
10
11
12
```



# Factorial Dynamic Programming Implementation

```
1 unsigned long memoized_fac(unsigned long n)
2 {
3     /* Fancy initialization of the static map with an initializer list Maps 0->1 */
4     static map<unsigned long, unsigned long> memo = {
5         {0, 1},
6     };
7
8     map<unsigned long, unsigned long>::iterator lookup = memo.find(n);
9     if (lookup != memo.end()) {
10         return lookup->second;
11     } else {
12         unsigned long result = n * memoized_fac(n - 1);
13         memo[n] = result;
14         return result;
15     }
16 }
```

