



The Beauty and Joy of Computing

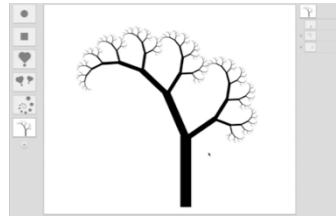
Lecture #10 Recursion II



UC Berkeley EECS
Sr Lecturer SOE
Dan Garcia

RECURSIVE DRAWING

Toby Shachman created this amazing spatial programming language called "Recursive Drawing" that allows you to create drawings (even recursive ones) without typing a line of code. It's a great example of a next-generation interface...



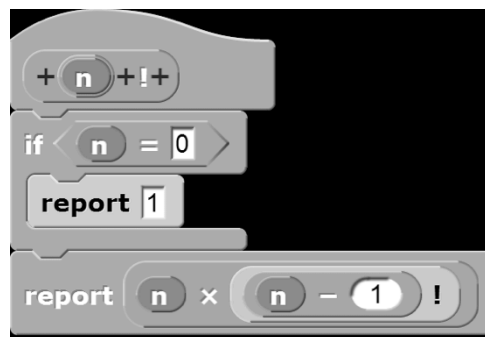
recursivedrawing.com



How the Computer Works ... n!

- **Factorial(n) = n!**
Inductive definition:
 - $n! = 1$, $n = 0$
 - $n! = n * (n-1)!$, $n > 0$
- **Let's act it out...**
 - "contractor" model
 - 5!

n	n!
0	1
1	1
2	2
3	6
4	24
5	120



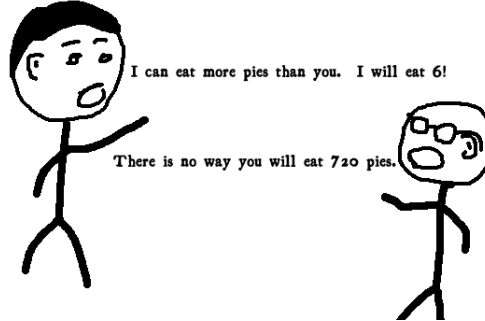


Order of growth of # of calls of n!

- a) Constant
- b) Logarithmic
- c) Linear
- d) Quadratic
- e) Exponential

(source: FallingFifth.com)

PIE-EATING CONTEST



Garcia



en.wikipedia.org/wiki/Fibonacci_number
www.ics.uci.edu/~epstein/161/960109.html

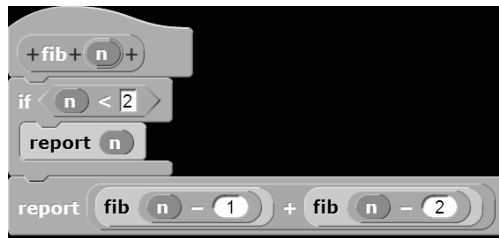
How the Computer Works ... fib(n)

Inductive definition:

- $fib(n) = n$, $n < 2$
 - $fib(n) = fib(n-1) + fib(n-2)$, $n > 1$
- $$F(n) := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

Let's act it out...

- "contractor" model
- fib(5)



n	fib(n)
0	0
1	1
2	1
3	2
4	3
5	5



Leonardo de Pisa
aka, Fibonacci



Let's now: trace... (gif from Ybungalobill@wikimedia)



Garcia



Order of growth of # of calls of fib(n)

- a) Constant
- b) Logarithmic
- c) Linear
- d) Quadratic
- e) Exponential



Chimney of Turku Energia, Turku, Finland featuring Fibonacci sequence in 2m high neon lights. By Italian artist [Mario Merz](#) for an environmental art project. (Wikipedia)



Counting Change (thanks to BH)

- Given coins {50, 25, 10, 5, 1} how many ways are there of making change?
 - 5
 - 2 (N, 5P)
 - 10
 - 4 (D, 2N, N5P, 10P)
 - 15
 - 6 (DN, D5P, 3N, 2N5P, 1N10P, 15P)
 - 100?

```

+ Count + Change + amount + Using + coins : +
if amount < 0 or empty? coins
  report 0
if amount = 0
  report 1
report Count Change amount Using all but first of coins +
       Count Change amount - item 1 of coins Using coins

```

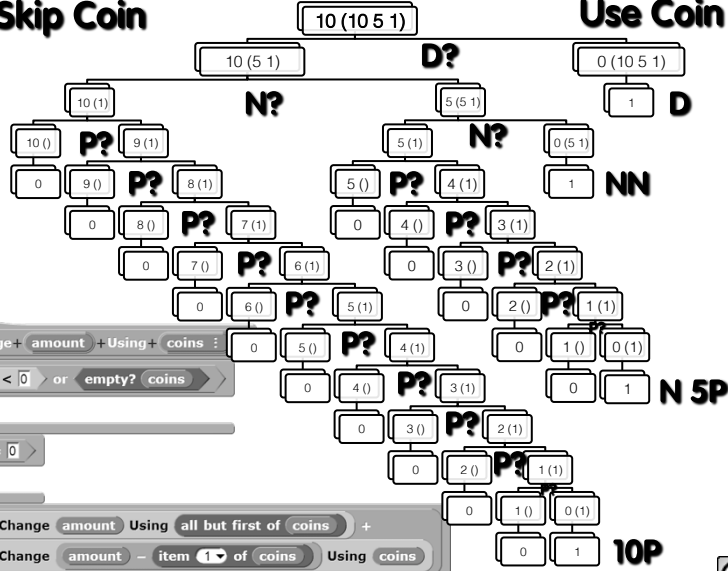




Call Tree for "Count Change 10 (10 5 1)"

← Skip Coin

Use Coin →



```

+Count-Change+ amount +Using+ coins :
if amount < 0 or empty? coins
report 0
if amount = 0
report 1
report Count-Change amount Using all but first of coins +
Count-Change amount - item 1 of coins Using coins
  
```



"I understood Count Change"

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree



img4.joyreactor.com/pics/post/drawing-recursion-girl-275624.jpeg





Summary

- It's important to understand the machine model
- It's often the cleanest, simplest way to solve many problems
 - Esp those recursive in nature!
- Recursion is a very powerful idea, often separates good from great (you're great!)

Menger Cube by Dan Garcia

