Costs and Sneezewort and Growth

532. Achillea Ptarmica L.
Sneezewort

One-Slide Summary

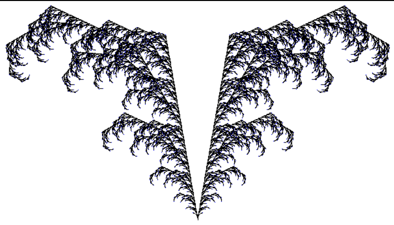
- The basic recursive computation of Fibonacci can take quite a while. **There are faster ways**.
- We can formally measure and evaluate the cost of a computer program. We abstract away details such as processor speed and instead measure how **the solving time increases as the input increases**.
- **g is in O(f)** iff there exist positive constants c and n_0 such that $g(n) \le cf(n)$ for all $n \ge n_0$.
- If g is in O(f) we say that f is an **upper bound** for g.

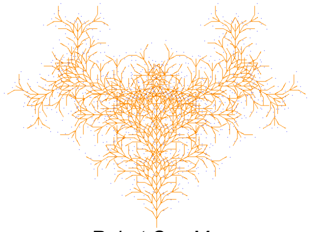
PS3 Attempts

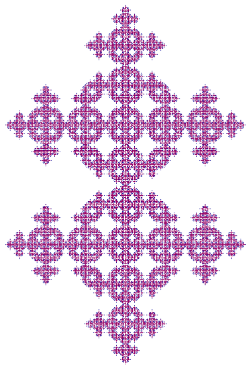
- As of Wednesday, only 17 of you had submitted anything for PS3.
  - ... with a high student score of 18/23.
- Presumably the rest of you have allocated 13+ hours between now and Tuesday. :-)
  - Starting problem sets in this class “at the last minute” is the conceptual equivalent of Getting Involved in a Land War in Asia.

Outline

- Sneezewort and Fibonacci
- **Cost** of computing Fibonacci
- **Cost** of sorting
- Intro to Big-Oh Notation

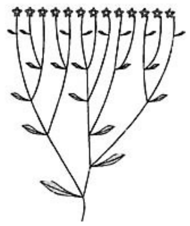

"V" shrubbery
by Andrew Jesien, Becky Elstad


Robot Cav Man
by Jamie Jeon & Walter Borges

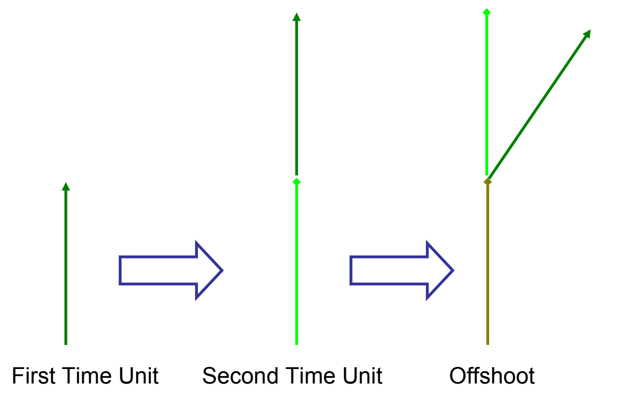

After the Incident
by Ben Morrison and Liz Peterson

Sneezewort

- Achillea ptarmica is real.
- It is “moste efficacious in the inflaming of the braine, and [is] therefore much used in Confusing and Befuddlement Draughts, where the wizard is desirous of producing hot-headedness and recklessness.”
  - Order of the Phoenix, p.18
- Sneezewort's pattern of development displays the Fibonacci sequence.

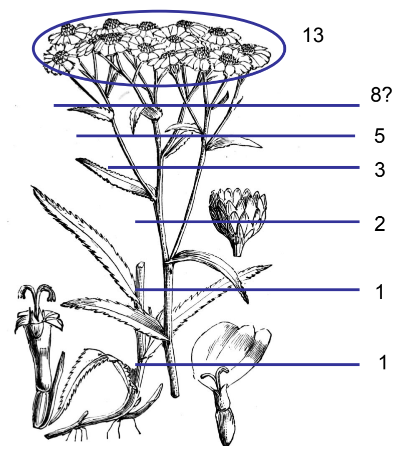


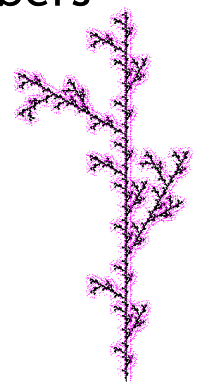
Sneezewort Growth



Could we model Sneezewort with PS3 code?

Sneezewort Numbers

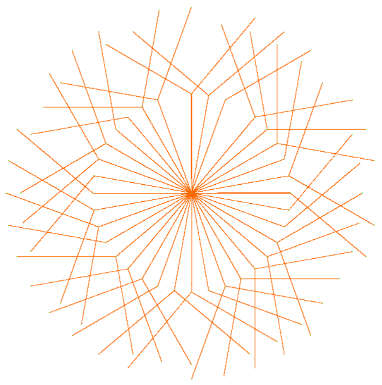




pink
by Jessica Geist, Ellen Clarke

Fibo Results

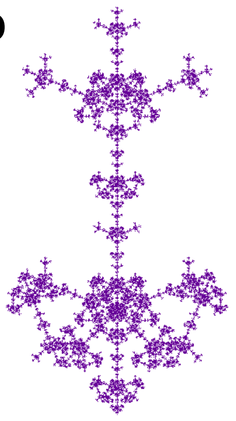
```
>>> fibo(2)
1
>>> fibo(3)
2
>>> fibo(4)
3
>>> fibo(10)
55
>>> fibo(60)
Still working…
```



At least we finished.
by Dmitriy Semenov and Sara Alspaugh

Tracing Fibo

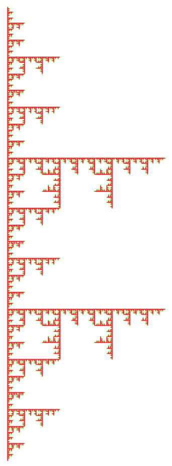
```
>>> (fibo 3)
| fibo(3)
|       fibo(2)
|       1
|       fibo(1)
|       1
|2
2
```



Purple Arrow
by Rachel Lathbury and Andrea Yoon

```
>>> fibo(5)
|fibo(5)
| fibo(4)
| |fibo(3)
| | fibo(2)
| | 1
| | fibo(1)
| | 1
| |2
| |fibo(2)
| |1
| 3
| fibo(3)
| |fibo(2)
| |1
| |fibo(1)
| |1
| 2
|5
5
```

A right-wing Christmas - awwwwww......
by Andrew Baker & Emily Lam



To calculate fibo(5) we calculated:
fibo(4) 1 time
fibo(3) 2 times
fibo(2) 3 times
fibo(1) 2 times } 5 times total
= 8 calls to fibo
= fibo(6), interestingly ...

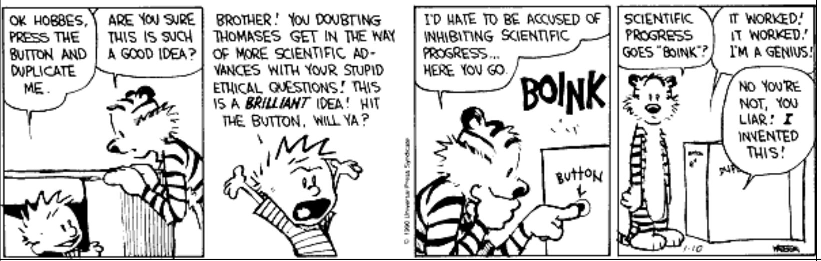
How many calls to calculate fibo(60)?

Liberal Arts Trivia: History

- This 20th-century American inventor is credited with the phonograph, the carbon telephone transmitter, the practical electric light, and the phrase "Genius is one percent inspiration, ninety-nine percent perspiration." He fought against Nikola Tesla's alternating current in the so-called War of the Currents.

Liberal Arts Trivia: Film Studies

- In this Oscar-nominated 2006 film, David Bowie is almost torched by Thomas Edison's goons but invents a teleportation machine for Wolverine so that he can defeat Batman in a magic trick competition because he thinks Batman killed his wife.


OK HOBBES, PRESS THE BUTTON AND DUPLICATE ME.
ARE YOU SURE THIS IS SUCH A GOOD IDEA?
BROTHER! YOU DOUBTING THOMASES GET IN THE WAY OF MORE SCIENTIFIC ADVANCES WITH YOUR STUPID ETHICAL QUESTIONS! THIS IS A BRILLIANT IDEA! HIT THE BUTTON, WILL YA?
I'D HATE TO BE ACCUSED OF INHIBITING SCIENTIFIC PROGRESS... HERE YOU GO.
BOINK
BUTTON
SCIENTIFIC PROGRESS GOES "BOINK"?
IT WORKED! IT WORKED! I'M A GENIUS!
NO YOU'RE NOT, YOU LIAR! I INVENTED THIS!

Liberal Arts Trivia: Physics

- Count Alessandro Antonio Anastasio Volta was a 19th-century Italian physicist. Volta studied what we now call capacitance, developing separate means to study both electrical potential *V* and charge *Q*, and discovering that for a given object they are proportional. His experiments in “animal electricity”, in which two different metals were connected in series with frog's legs, eventually led to his most famous discovery. What was it?

#14

fast_fibo

```
# 1 + 1 = 2
#     1 + 2 = 3
#         2 + 3 = 5
#             3 + 5 = 8
#                 a + b = fibo(n)
def fast_fibo(n):
  def fib_helper(a, b, left):
    if left <= 0:
      return b
    return fib_helper(b, a+b, left-1)
  return fib_helper(1, 1, n-2)
```

#15

Fast-Fibo Results

```
>>> fast_fibo(10)
55
>>> a = time(); print fast_fibo(60); \
      print time()-a
1548008755920
7.10487365723e-05     seconds
```

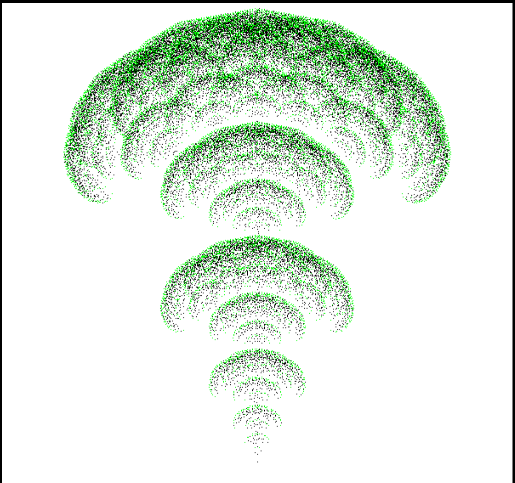
The original fibo would take at least 2.5 Trillion applications. A 2.5 GHz computer does 2.5 *Billion* simple operations per second, so 2.5 Trillion applications operations take ~1000 seconds.
Each application of fibo involves hundreds of simple operations…

#16

```
# The Earth's mass is 6.0 x 10^24 kg
>>> mass_of_earth = 6 * pow(10,24)
# A typical rabbit's mass is 2.5 kilograms
>>> mass_of_rabbit = 2.5
>>> (mass_of_rabbit * fast_fibo(60)) / mass_of_earth
6.450036483e-013
>>> (mass_of_rabbit * fast_fibo(120)) / mass_of_earth
2.2326496895795693
```

According to Bonacci’s model, after less than 10 years, rabbits would out-weigh the Earth!

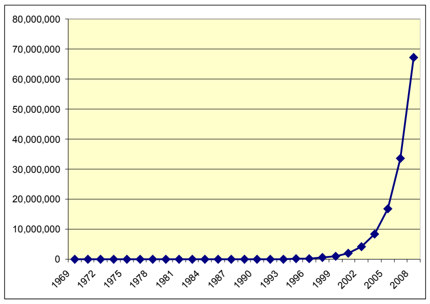
#17


Broccoli Fallout by Paul DiOrio, Rachel Phillips

Evaluation Cost

Actual running times vary according to:

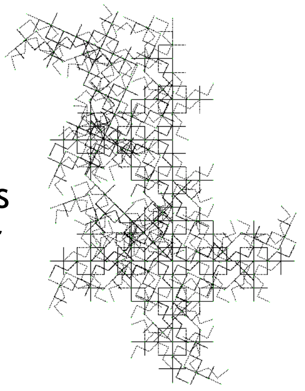
- How fast a processor you have
- How much memory you have
- Where data is located in memory
- How hot it is
- What else is running
- etc...



Moore's "Law" – computing power doubles every 18 months

Measuring Cost

- How does the cost scale with the ***size of the input***?
- If the input size increases by one, how much longer will it take?
- If the input size *doubles*, how much longer will it take?



Untitled
Nokomis McCaskill
Chris Hooe

Cost of Fibonacci Procedures

```
def fibo(n):
  if n <= 2:
    return 1
  return fibo(n-1) + fibo(n-2)
```

```
def fast_fibo(n):
  def fib_helper(a, b, left):
    if left <= 0:
      return b
    return fib_helper(b, a+b, left-1)
  return fib_helper(1, 1, n-2)
```

Input	fibo	fast_fibo
m	q	mk
$m+1$		$(m+1)k$
$m+2$	at least q^2	$(m+2)k$

Cost of Fibonacci Procedures

```
def fibo(n):
  if n <= 2:
    return 1
  return fibo(n-1) + fibo(n-2)
```

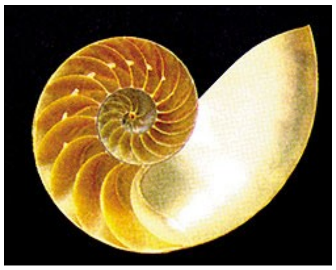
```
def fast_fibo(n):
  def fib_helper(a, b, left):
    if left <= 0:
      return b
    return fib_helper(b, a+b, left-1)
  return fib_helper(1, 1, n-2)
```

Input	fibo	fast_fibo
m	q	mk
$m+1$	$q*\Phi$	$(m+1)k$
$m+2$	at least q^2	$(m+2)k$

Φ = (1 + (sqrt 5)) / 2 = "**The Golden Ratio**" ~ 1.618033988749895...
~ fast-fibo(61) / fast-fibo(60 = 1.618033988749895

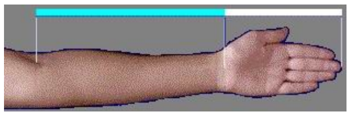
The Golden Ratio

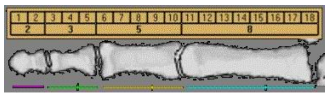
Parthenon

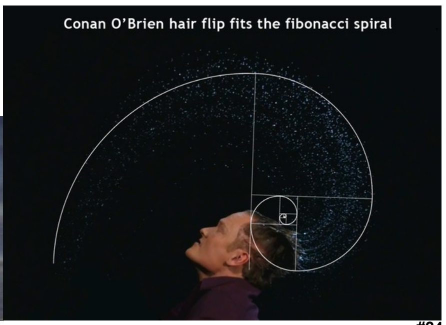


Nautilus Shell

More Golden Ratios







Sorting

```
>>> def ascending(x,y): return x - y
>>> sorted([5,2,4,1,3], ascending)
[1, 2, 3, 4, 5]
>>> def descending(x,y): return y - x
>>> sorted([5,2,4,1,3], descending)
[1, 2, 3, 4, 5]
>>> def longer(x,y): return len(x) - len(y)
>>> sorted([“allons”,”enfants”,”de”],longer)
[“de”, “allons”, “enfants”]
```

#25

“You're The Best Around”

```
def find_best(things, better):
  if len(things) == 1: return things[0]
  return pick_better(better, things[0], \
    find_best(things[1:], better)

def pick_better(better, a, b):
  return (a if better(a,b) else b)
```

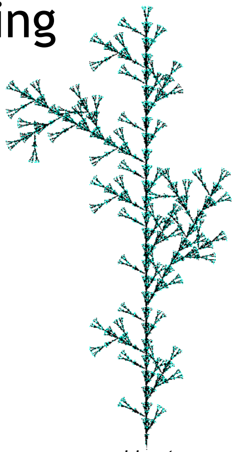
```
def find_best(things, better):
  return sorted(things, better)[0]
```

Which is faster and by how much?

#26

Simple Sorting

- Can we use find_best to implement sort?
  - Yes!
- Use find_best(lst) to find the best
- Remove it from the list
  - Adding it to the answer
- Repeat until the list is empty



crazy blue tree
by Victor Malaret, Folami Williams

#27

Simple Sort

```
# cf = comparison function
def sort(lst, cf): # simple sort
  If not lst: return []
  best = find_best(lst, cf)
  return [best] + sort( \
    remove(lst, best), cf)

# remove(lst,x) = filter ... elt != x ...
```

#28

Sorting

```
def sort(lst, cf): # simple sort
  If not lst: return []
  best = find_best(lst, cf)
  return [best] + sort( \
    remove(lst, best), cf)
```

```
def find_best(things, better):
  if len(things) == 1: return things[0]
  return pick_better(better, things[0], \
    find_best(things[1:], better)

def pick_better(better, a, b):
  return (a if better(a,b) else b)
```

How much work is sort?

#29

Sorting Cost

- What grows?
  - n = the number of elements in lst
- How much work are the pieces?

  find-best:

  remove:

#30

Sorting Cost

- What grows?
  - n = the number of elements in lst
- How much work are the pieces?
  - find-best: work scales as n (increases by one)
  - remove: work scales as n (increases by one)
- How many times does sort evaluate find-best and delete?

Sorting Cost

- What grows?
  - n = the number of elements in lst
- How much work are the pieces?
  - find-best: work scales as n (increases by one)
  - remove: work scales as n (increases by one)
- How many times does sort evaluate find-best and delete? n
- Total cost: scales as

Sorting Cost

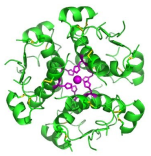
- What grows?
  - n = the number of elements in lst
- How much work are the pieces?
  - find-best: work scales as n (increases by one)
  - remove: work scales as n (increases by one)
- How many times does sort evaluate find-best and delete? n
- Total cost: scales as n^2

Sorting Cost

```
def sort(lst, cf): # simple sort
  if not lst: return []
  best = find_best(lst, cf)
  return [best] + sort(remove(lst, best), cf)
def find_best(things, better):
   if len(things) == 1: return things[0]
   return pick_better(better, things[0], \
     find_best(things[1:], better)
```

If we double the length of the list, the amount of work *approximately* quadruples: there are twice as many applications of find-best, and each one takes twice as long

Liberal Arts Trivia: Medicine



- Nicolae Paulescu was a 20th century physiologist and professor of medicine. He is considered the true discoverer of hormone that causes most of the body's cells to take up glucose from the blood. His first experiments involved an aqueous pancreatic extract which, when injected into a diabetic dog, proved to have a normalizing effect on blood sugar levels. Name the hormone.

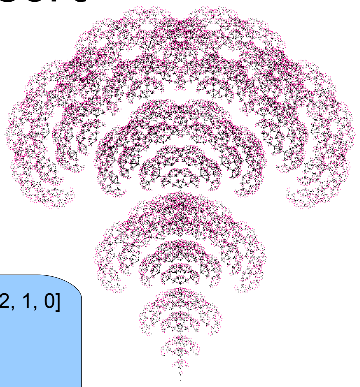
Liberal Arts Trivia: Sailing

- Name the collection of apparatus through which the force of the wind is transferred to the ship in order to propel it forward – this includes the masts, yardarms, sails, spars and cordage.

Timing Sort

```
def find_best(things, better):
      if len(things) == 1: return things[0]
      rest = find_best(things[1:], better)
      return things[0] if better(things[0], rest) else rest

def sort(lst, cf):
      if not lst: return []
      best = find_best(lst, cf)
      lst.remove(best) # tricky!
      return [best] + sort(lst, cf)

def descending(x,y): return x > y

def time_sort(k):
      a = time.time()
      sort(range(k), descending)
      return time.time() - a

print sort(range(10), descending)
print time_sort(20)
print time_sort(40)
print time_sort(80)
print time_sort(160)
```

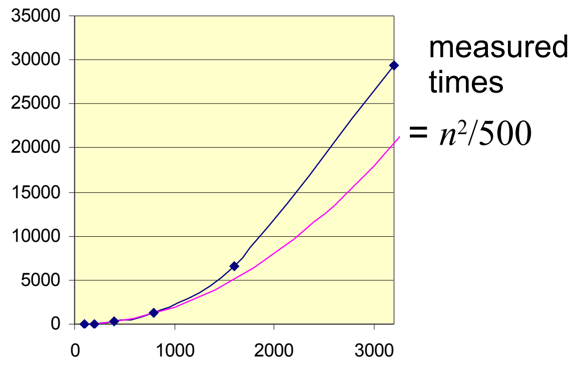
```
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
0.00014
0.00057
0.00234
0.00960
```



Cherry Blossom
by Ji Hyun Lee, Wei Wang

#37

Timing Sort



#38

Growth Notations

- $g \in O(f)$ ("Big-Oh")
  g grows **no faster than** f (f is upper bound)
- $g \in \Theta(f)$ ("Theta")
  g grows **as fast as** f (f is tight bound)
- $g \in \Omega(f)$ ("Omega")
  g grows **no slower than** f (f is lower bound)

Which one would we most like to know?

#39

Meaning of O ("big Oh")

g is in $O(f)$ iff:

There are positive constants c and n_0 such that

$$g(n) \leq cf(n)$$

for all $n \geq n_0$.

O Examples

g is in $O(f)$ iff there are positive constants c and n_0 such that $g(n) \leq cf(n)$ for all $n \geq n_0$.

Is n in $O(n^2)$?
Is $10n$ in $O(n)$?
Is n^2 in $O(n)$?

T-Mobile @Home Package

Get unlimited nationwide calling from your home phone with T-Mobile @Home service.
$9.99 per month + **$999,999.99** phone price

More details ›

O Examples

g is in $O(f)$ iff there are positive constants c and n_0 such that $g(n) \leq cf(n)$ for all $n \geq n_0$.

Is n in $O(n^2)$? Yes, $c = 1$ and n_0=1 works.
Is $10n$ in $O(n)$? Yes, $c = 1/10$ and n_0=1 works.
Is n^2 in $O(n)$? No, no matter what c we pick, $cn^2 > n$ for big enough n ($n > c$)

#42

Revenge of O Examples

g is in $O(f)$ iff there are positive constants c and n_0 such that $g(n) \le cf(n)$ for all $n \ge n_0$.

Is $n+5$ in $O(n^2)$?

Is n^2-100 in $O(n)$?

Is n^2 in $O(n^3)$?

#43

Revenge of O Examples

g is in $O(f)$ iff there are positive constants c and n_0 such that $g(n) \le cf(n)$ for all $n \ge n_0$.

Is $n+5$ in $O(n^2)$? Yes, $c = 1$ and $n_0=3$ works.

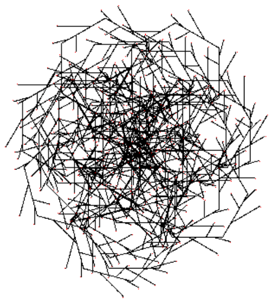
Is n^2-100 in $O(n)$? No, no matter what c we pick, $cn^2\text{-}100 > n$ for big enough n.

Is n^2 in $O(n^3)$?
Yes, $c = 1$ and $n_0=1$ works.
Yes, $c = 2$ and $n_0=77$ works.
Yes, $c = 55$ and $n_0=102$ works.

#44

Homework

- Course Book Chapter 7
  - Has a formal notation for this kind of analysis!
- Problem Set 3 due soon!


The Mask
by Zachary Pruckowski, Kristen Henderson

#45