

# Lecture 12: Something about Sneezewort

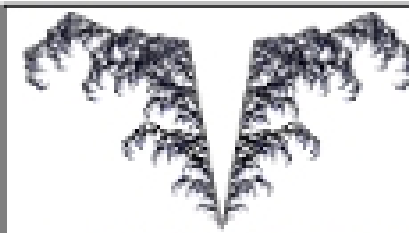
From *Illustrations of the British Flora*  
(1824) by Walter Hood Fitch  
<http://www.comlab.virginia.edu/CS211/12a/>



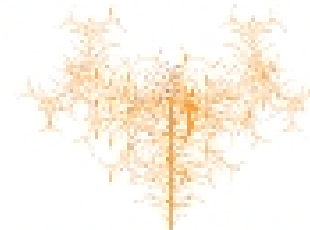
530. *Adonis vernalis* L.  
*Sneezewort*

David Evans

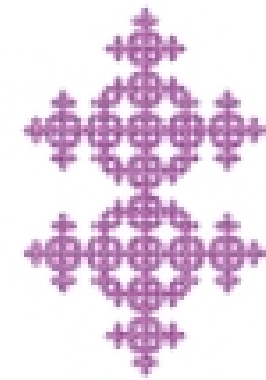
<http://www.comlab.virginia.edu/cs211/>



"V" Fruberry  
By Andrew Jason, Scott Blatod

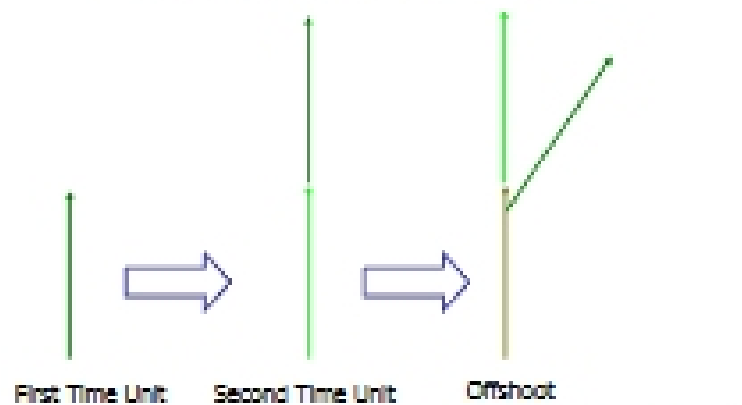


Robot Car Man  
By Jamie Jean & Walter Berger



After the Incident  
By Ben Morrison and Lisa Peterson

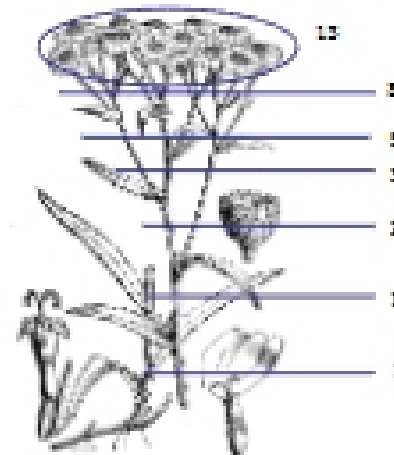
## Sneezewort Growth



First Time Unit    Second Time Unit    Offshoot

Could we model Sneezewort with P53 code?

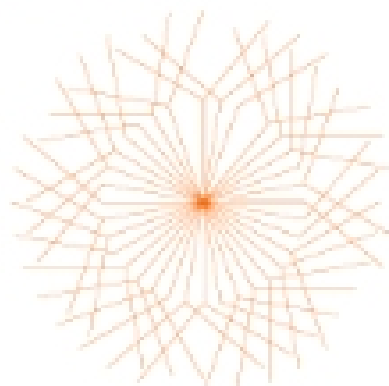
## Sneezewort Numbers



pink  
By Jessica Geis, Ellen Clarke

## Fibo Results

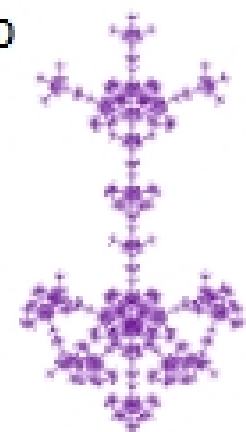
- > (fibo 2)
- 1
- > (fibo 3)
- 2
- > (fibo 4)
- 3
- > (fibo 10)
- 55
- > (fibo 60)
- Still working...



At least we finished.  
by Dmitry Soronov and Sara Alrough

## Tracing Fibo

```
> (require-library "trace.ss")
> (trace fibo)
(fibo)
> (fibo 3)
|((fibo 3)
|  (fibo 2)
|  1
|  (fibo 1)
|  1
|  2
|  2
```



Apple Arrow  
By Rachel Lathbury and Andrea Yoon



## 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?



Nickolas McCaskill  
Chris Hooc

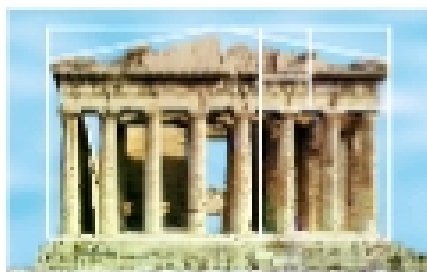
## Cost of Fibonacci Procedures

```
(define (fib n)
  (if (= n 1) (= n 2)
      1))
(define (fast-fib n)
  (define (fib-helper a b left)
    (if (= left 0)
        b
        (fib-helper (+ a b) (- left 1))))
  (fib-helper 1 1 (- n 2)))
```

Input	fib	fast-fib
$m$	$q$	$z = mk$
$m+1$	$q^{+\Phi}$	$(m+1)k$
$m+2$	at least $q^2$	$(m+2)k$

$\Phi = \frac{1}{2} (\sqrt{5} + 1) \approx 1.618033988749895...$   
 $\sim \frac{1}{2} (\text{fast-fib } 50) (\text{fast-fib } 50) = 1.618033988749895$

## The Golden Ratio

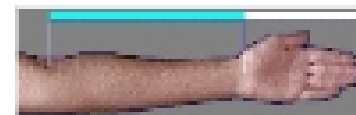


Parthenon



Nautilus Shell

## More Golden Ratios



<http://www.farkle.org/casayam18004.html>  
 By Olegiy Stoklov

## PS2 Question

```
(define (find-best-hand hands)
  (car (sort hands higher-hand?)))
```

```
(define (find-best lst cf)
  (if (= 1 (length lst)) (car lst)
      (pick-better cf (car lst) (find-best (cdr lst) cf))))
(define (pick-better cf num1 num2)
  (if (cf num1 num2) num1 num2))
(define (find-best-hand hands)
  (find-best hands higher-hand?))
```

Which is better and by how much?

## Simple Sorting

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



*easy blue tree*  
 By Victor Malaric, Palani Williams