

CS 2110 Lec. 8 : Recursion, Linked Lists

Recursion : define something in term of itself

Recursive algorithms : solve problems by using the same algorithm on smaller problem

Ex/ Fibonacci numbers $F_n = f(n)$

$$f(n) = \begin{cases} f(n-1) + f(n-2) & \text{if } n > 1 \\ n & \text{if } n \in \{0, 1\} \end{cases}$$

< recursive case
< base case

0, 1, 1, 2, 3, 5, 8, ...

Not circular: can expand until base case

$$\begin{aligned} f(3) &= f(2) + f(1) = f(1) + f(0) + f(1) \\ &= 2 \end{aligned}$$

/** Returns: F_n . Requires: $n \geq 0$, $F_n < 2^{31}$ } implicit
*/ precond.

```
int f(int n) {  
    if (n > 1) return f(n-1) + f(n-2);  
    return n;  
}
```

Total work $\propto F_n$ calls $F_n \approx \frac{\phi^n}{\sqrt{5}}$

Ex/ Exponentiation w/ by squaring & multiplication

/* Returns x^e . Requires: $e \geq 0$ */

int pow(int x, int e) {

if (e == 0) return 1; < base case

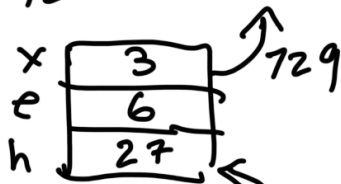
int h = pow(x, e/2); // $h = x^{\lfloor \frac{e}{2} \rfloor}$

if (e % 2 == 0) return h * h; // $(x^{e/2})^2 = x^e$

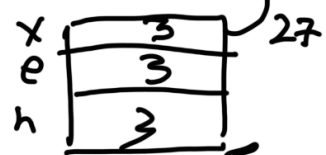
return h * h * x; // $h = x^{\lfloor \frac{e-1}{2} \rfloor}$, $(x^{\lfloor \frac{e-1}{2} \rfloor})^2 = x^{e-1}$

total calls = $\log_2 e$. Execution: stack of activation records (ARs)

pow(3, 6)



pow(3, 3)



pow(3, 1)



pow(3, 0) x ~~3~~ e ~~0~~

calls: push AR onto stack
return: pop AR off stack

works? By induction. Base case: 0 ✓
Recursive case works assuming that it works on smaller inputs? ✓
0 ✓ 1 ✓ 2 ✓ 3 ✓ ...

Integer division in Java rounds down

$$1/2 == 0$$

$$1.0/2.0 == 0.5$$

Recursive types

Type defined in terms of itself.

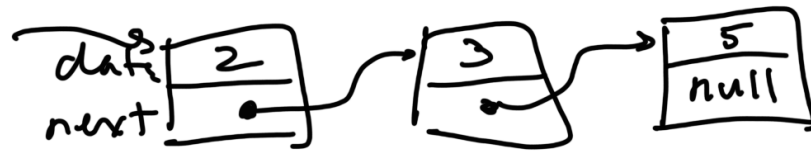
Linked list

A data structure : objects linked by pointers.

Linked list has node objects with:

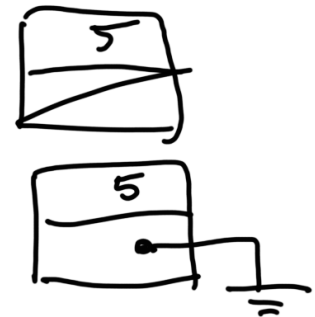
1. data
2. pointer to next node in list. (if any)

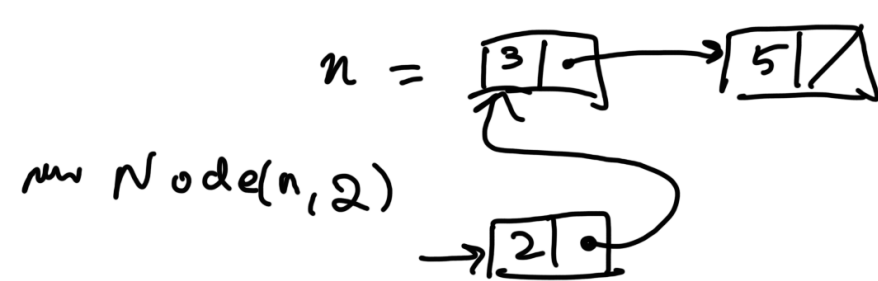
Ex/ list (2, 3, 5) representation:



```
class Node {  
    object data;  
    Node next; <recursion  
}
```

or:



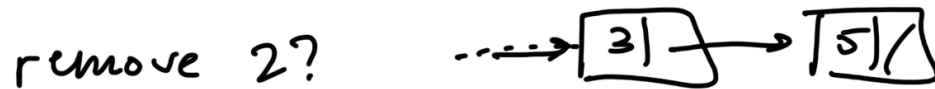
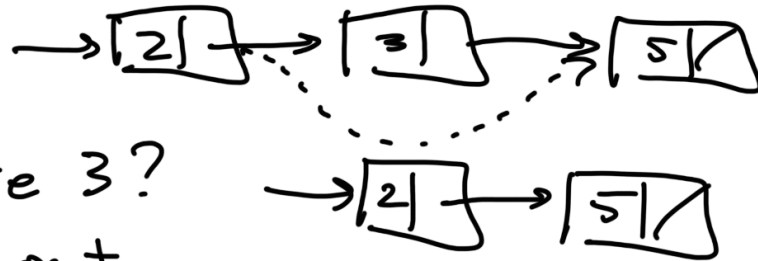


$(3, 5)$

$(2, 3, 5)$

- Iterative algorithms can always be made recursive.
- But: stack of ARs is slower than 1 AR.

Removing



How to "go back" to prev. node?

