

Test Review, Asymptotics & Sorting

Thursday, October 1, 2015 11:37 AM

Test Topics

• Recursion

- Pseudocode
- Stack traces
- Fact, Qb, listContents

• Lists

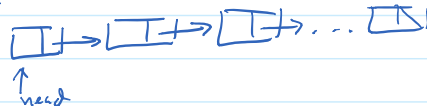
- ArrayLists

- Draw them
 - Fast ops vs slow ops
 - Know they are backed by arrays
- know random access is $O(1)$
insert at beg: $O(n)$
insert at end $O(1)$

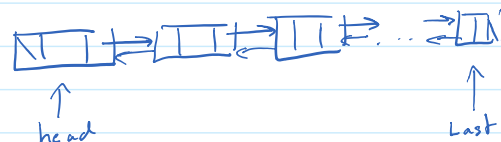
- Linked Lists

- Same as above, for singly and doubly

Singly:



Doubly:



Doubly:

- Insert at beg & end $O(1)$ in general $O(n)$
- Remove at beg & end $O(1)$ in general $O(n)$

singly excludes end

- Sequences & Cycles

- Know tort & hare algorithm and how to apply it.

- Asymptotics

- Know how to order functions by "size"
- Know that $O(g(x)) \approx \leq g(x)$

$$\Omega \approx \geq$$

$$\Theta \approx =$$

$$x \in O(x)$$

Write in asymptotics, not as exact

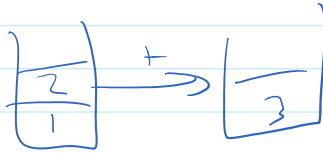
- Stacks & Queues

- Stacks $\cong$ Recursion
- RPN (stack trace)

1 2 +

}

1 2 +



• listContents

iteratively

```
def listContents (root):
    if (root is not a directory):
        print (root.name)
        return
    EndIf
    stack = new Stack
    stack.push (root)
    while (stack is not empty):
        file = stack.pop()
        print (file.name)
        if (file is a directory):
            for (child: file.getContents()): // getContents lists all children to 1 depth
                stack.push (child)
            endfor
        end if
    endwhile
```

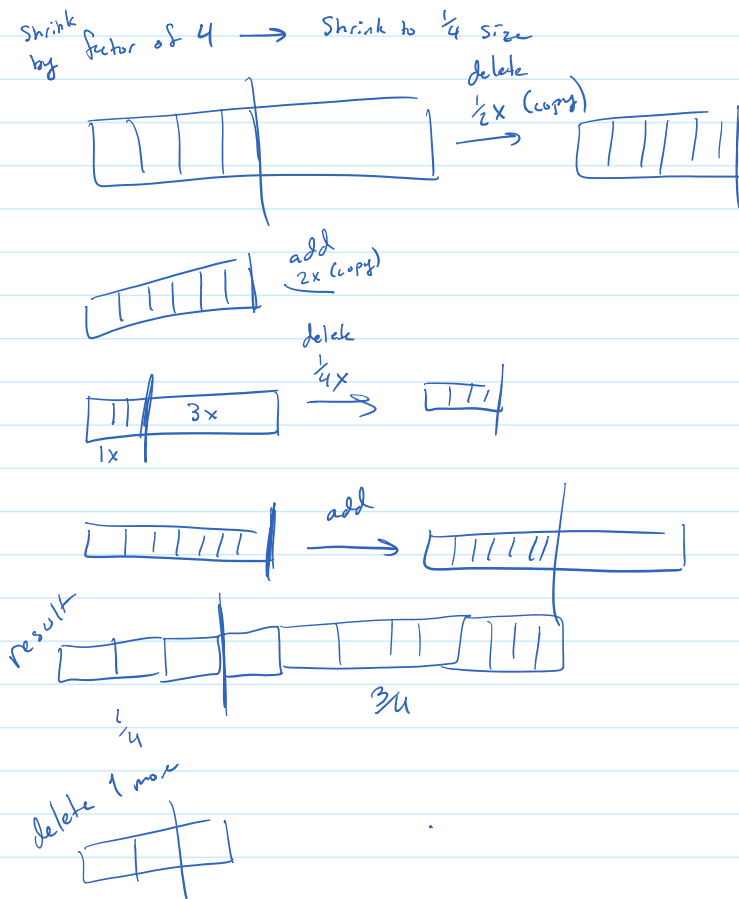
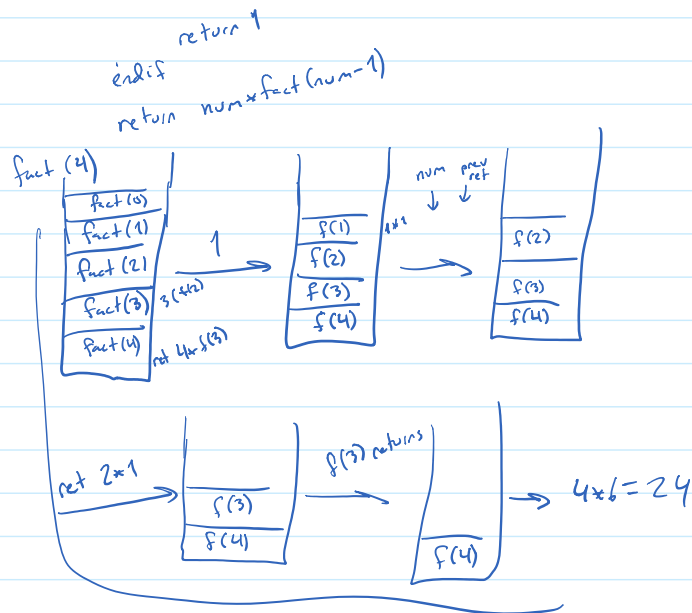
not necessary

recursive

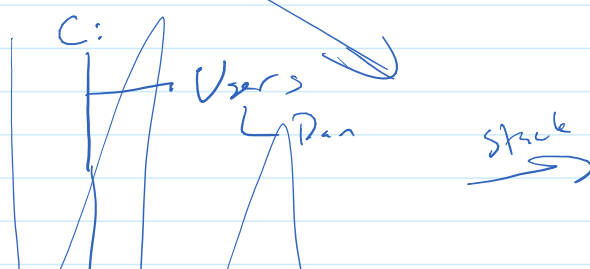
```
def listContents (file):
    if (file.isDirectory()): // is directory returns true
        // if file is a directory
        print (file.name)
        return
    end if
    print (file.name)
    for (child: file.getContents()):
        listContents (child) // recurse
    end for
```



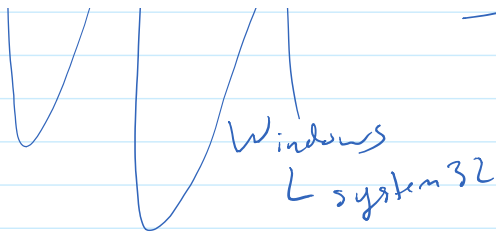
```
def fact (num):
    if (num <= 0):
        return 1
```



Stacks & queues



C:
Users
Den
Windows
system 32



Windows
system32

Queue
→

C: } 1st level
Users } 2nd level
Windows }
Den } 3rd level
system32 }

Tort & Hare

0 → 1 → 2 → 3 → 4



keys

	A	B	C
Values	C	E	D

A → C → D

0 → 3 → 6 → 2 → 5

0	1	2	3	4	5	6
3	5	5	6	3	6	2

Start on 0 ←

$O(1)$

T	H
---	---

T	H
0	0
3	6
6	5
2	2
2	0
5	3
6	6

Cycle exists

Cycle starts on 6