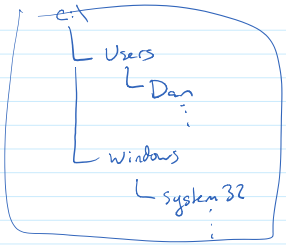
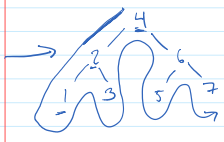
Trees & Test Review

Thursday, November 12, 2015 11:32 AM



listContents which prints out files in this order

"In-order traversal" (pre-order, post-order)



1 2 3 . . .

depthFirstSearch(root)

```
public void depthFirstSearch(Node node) {
  /* Choose an ordering for left,
   * this, and right
   */
  if (node == null) return;
  depthFirstSearch(node.left);
  print(node.elem);
  depthFirstSearch(node.right);
}
```

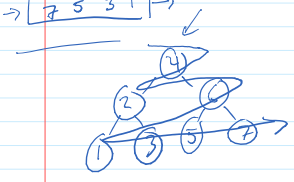
1 2 3 4 5 6 7

depthFirstSearch(4)
depthFirstSearch(2)
depthFirstSearch(1)
// depthFirstSearch(null)
print(1)
print(2)
dfs(3)
print(3)
print(4)
dfs(6)
dfs(5)
print(5)
print(6)
dfs(7)
print(7)

```
public void dfs(node) {
  stack = ∅
  stack.push(node)
  while (stack is not empty) {
    curNode = stack.pop()
    print(curNode)
    stack.push(curNode.left)
    stack.push(curNode.right)
  }
}
```

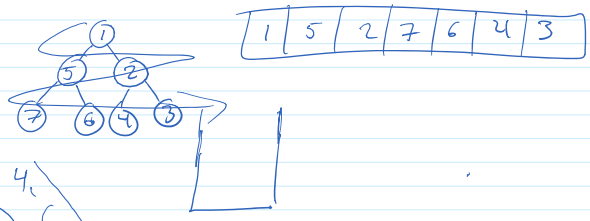
pre-order

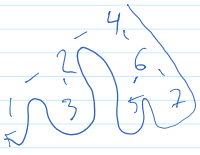
breadth-first-search




```
public void bfs(node) {
  queue = ∅
  queue.enqueue(node)
  while (queue is not empty) {
    curNode = queue.dequeue()
    print(curNode)
    queue.enqueue(curNode.left)
    queue.enqueue(curNode.right);
  }
}
```

4 2 6 1 3 5 7





4 6 7 5 2 3 1

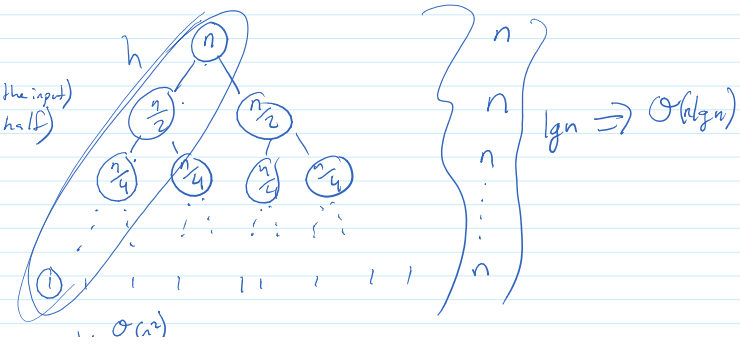
Test Review

Topics – Syllabus

Asymptotics
- Analyze mergesort

Sorting
- Recognize sorting algorithms
- Know merge, partition, heapify

```
foo(input):
  linear()
  foo(half the input)
  foo(other half)
```



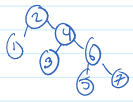
- recognize sorting algorithms
- Know merge, partition, heapify
- Know the table of asymptotics

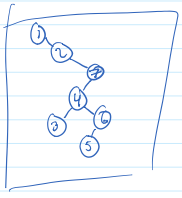
Heaps
- heapify, deleteMin, insert, getMin, size

Binary Search Trees
- insert, lookup, size
- rotations

dfs/bfs

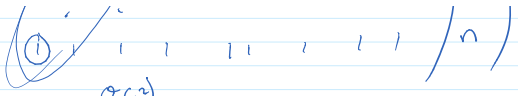
right 4 ✓
left 6 ✓
right 2 ✓
left 4





size n

foo (input):
constant()
linear()
constant()
foo (left)
constant()
foo (right)
constant()



Worst: $O(n^2)$

$O(1)$

$O(n)$

MinHeap
Max Heap

Heapsort

```
Heapsort ( array):                    Ω(n)
    heap = heapify(array)             O(n lg n)
    sorted array = ∅
    while (heap is not empty) {
        sortedArray.add( heap.deleteMin)
    }
    return sortedArray
}
```

Insert for Heaps