
CS310

Data Structures

K. Raven Russell
krusselc@gmu.edu
George Mason University

Midterm!



Today

- **Last Lecture**

- Tree Traversals

- **Today**

- Practice and Review



Review



Topics

- Big-O
- Lists
 - Dynamic Array Lists
 - Linked Lists
- Stacks
- Queues
- Priority Queues
- Sets and Maps
 - Basics
 - Hashing
- Trees
 - Tree Basics
 - Tree Traversals



Big-O

- For the typical algorithm that you use to perform calculations by hand, determine the running time to add two N-digit integers. Give a (tight) big-O bound.
- For the following program fragment, give a (tight) big-O analysis of the running time:
 - `for(int i = 1; i < n; i = i * 2)`
 `sum++;`

Dynamic Arrays

- You have a dynamic array with 2 elements in it and a capacity of 3. The capacity doubles when additional space is needed and shrinks by 50% (rounding down) when the size is at or below 50% capacity. What will be the capacity of the list after the following operations have been performed (answer should be a single integer number):

Add, Add, Remove, Remove, Add

Linked Lists

- Given the code sample:

```
class Node {  
    int value;  
    Node next;  
}
```

If the variable "head" is a reference to the Node at the start of a linked list 1->2->3

- what is the value of "**head.next.value**"?
- how do you reference the null link paired with the 3 (for example, if you wanted to add another node to the end)


```
static class Node<T> {
    T data;
    Node<T> next;
    public Node(T data) { this.data = data; }
}

public static void main(String[] args){
    Node<Integer> node1 = new Node<>(20);
    Node<Integer> node2 = new Node<>(10);
    Node<Integer> node3 = new Node<>(30);
    node1.next = node2;
    node2.next = node3;
    System.out.println(method2(node1));
}

public static int method2(Node<Integer> node){
    if (node == null) return 0;
    else return node.data+method2(node.next);
}
```

Linked List Code Reading

Given the definitions on the left, what would be the output when we run the main method?

Stacks and Queues

- What will be the contents of the data structure after the given sequence of operations if (a) the structure is a stack and (b) the structure is a queue.

```
Structure struc = new Structure();  
struc.add(4);  
struc.add(8);  
if (struc.peek() > 5)  
    struc.remove();  
struc.add(1);  
struc.add(6);  
struc.remove();
```

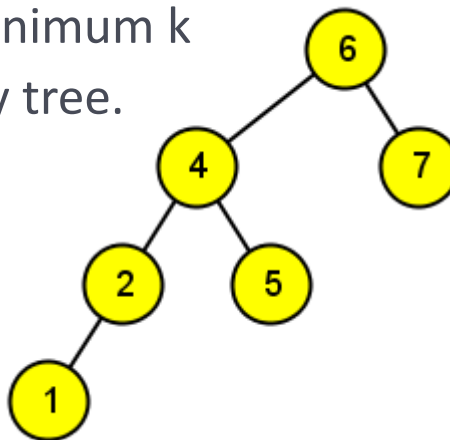
Hashing

- Given a hash table that uses linear probing and has a fixed table size of 10 and a hash function $h(x) = x$. Draw the hash table after the following operations:
 - add 4370
 - add 1321
 - add 6171
 - add 9679
 - remove 1321
 - remove 6171
 - add 1129
- What would the table look like if we had used separate chaining?

Tree Basics

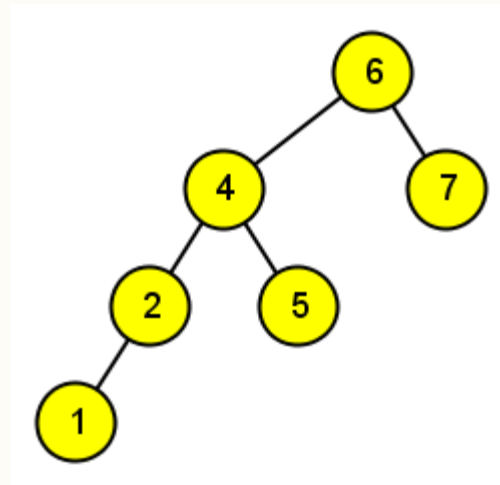
- For the following tree, determine if the tree is:
 - Full
 - Perfect
 - Nearly-complete
 - Balanced
 - Degenerate
- Additionally, identify:
 - the root
 - the inner nodes
 - the leaves
 - the children of node 4
 - the minimum k
 - the height of the tree
 - the depth of node 2

Assuming the minimum k visible for a k-ary tree.



Tree Traversals

Given the tree below, which type of traversal is “6 4 2 1 5 7”?




Code

- Given an array representing a list of integers, write the code to convert it into the equivalent linked list (using only nodes, not a linked list class). Make sure to define your node class.



Answers

- 
- $O(n)$ and $O(\lg n)$
 - 3
 - 2
 `head.next.next.next`
 - 60
 - `[4,1] <- top`
 and
 `[8 1 6] <- back`
 - `[4370,1129,TS,null,null,
 null,null,null,null,9679]`

 `[--] -> 4370`
 `[null]`
 `[null]`
 `[null]`
 `[null]`
 `[null]`
 `[null]`
 `[null]`
 `[--] -> 9679 --> 1129`
 - Nothing from the first list, 6, 6/4/2,
 1/5/7, 2/5, 2, 3, 2
 - Depth-First Pre-Order