

8-9:30?

Big-O

Topics include:

- Binary Trees / Binary Search Trees (Coding)
- Hash Tables
- External Chaining
- Linear Probing (Coding)
- Quadratic Probing
- Double Hashing
- AVL Trees (Coding)
- (2,4) Trees
- Splay Trees
- Sorting Algorithms
 - $O(n^2)$ (Bubble, Insertion, Selection)
 - Clever Sorts (Merge, Quick, LSD Radix)
 - Efficiency for all of the above $O()$
 - Anything covered in class, recitation, or on homework even if not on this list (no skiplist)

End of Stream
Jach
2/27

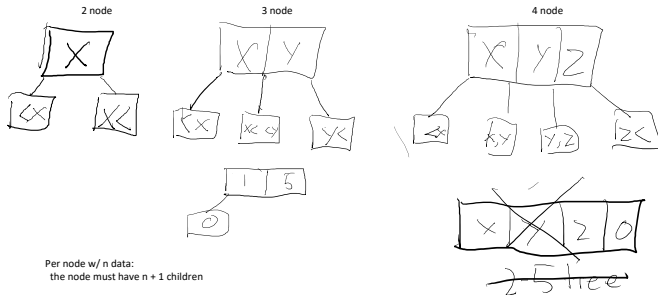
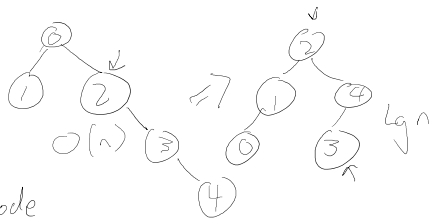
Sort: Best | ~~Avg~~ | Worst

2-4 Tree

Monday, July 2, 2018 8:07 PM

Binary Search Tree
Binary Search Property
AVL Tree
nodes w/
+ 1 data

1-3 data in node
2-4 children per node



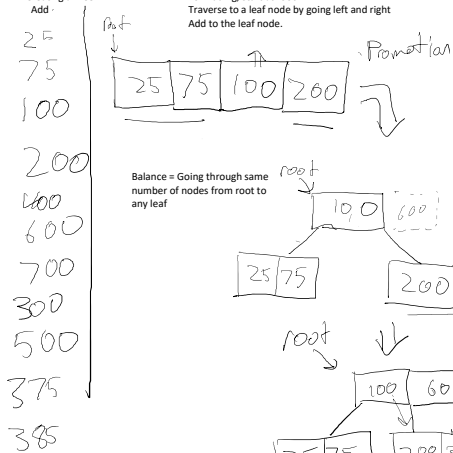
Per node w/ n data:
the node must have n + 1 children

Creating a Tree:

Add -

For Adding, start at root.

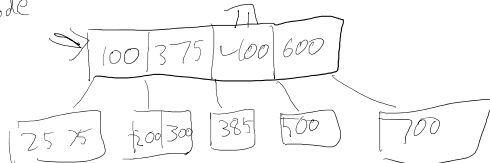
Traverse to a leaf node by going left and right
Add to the leaf node.



Balance = Going through same
number of nodes from root to
any leaf

Order 4 → B-Tree

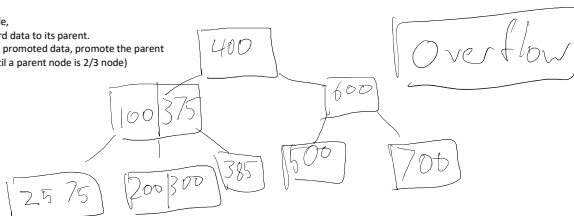
5 node



If adding to a 4 node leaf node,

Then add and promote the 3rd data to its parent.

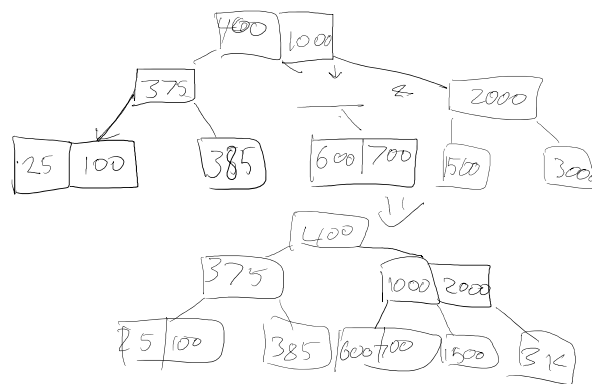
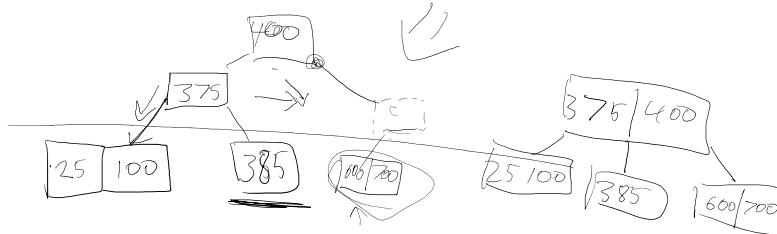
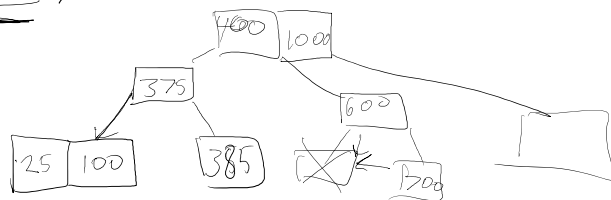
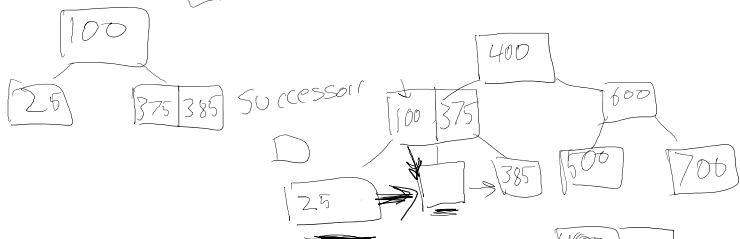
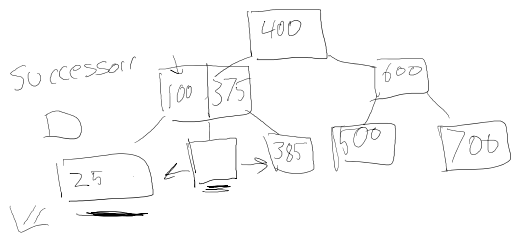
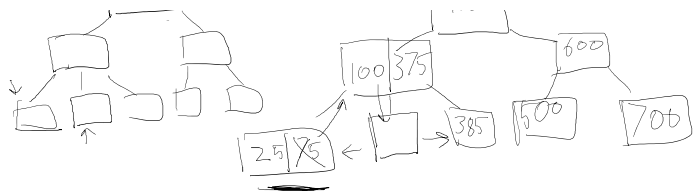
If the parent cannot hold this promoted data, promote the parent
Node as well. (repeat this until a parent node is 2/3 node)



Overflow

Remove





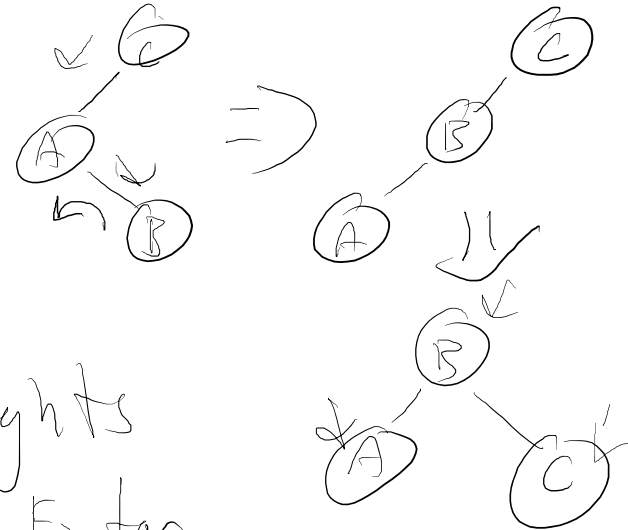
- BST $O(\log n)$

- Balance

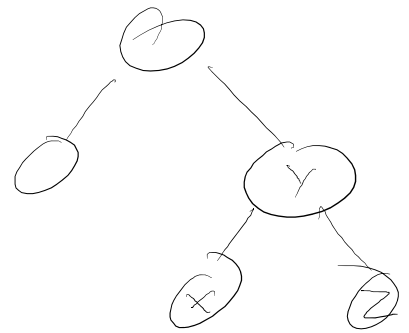
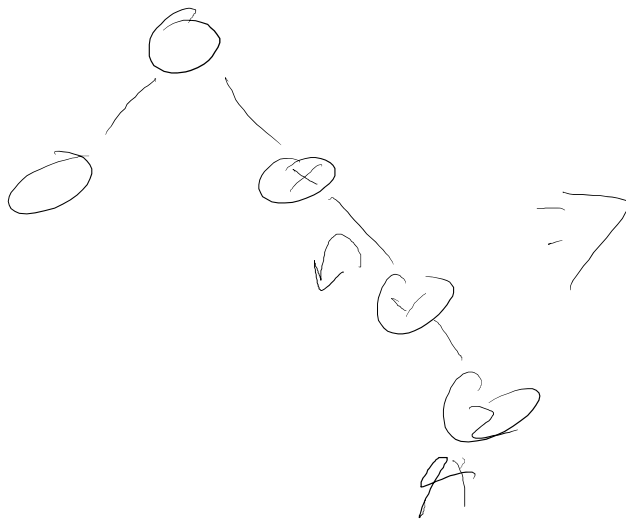
↳ rotations



Left-Right

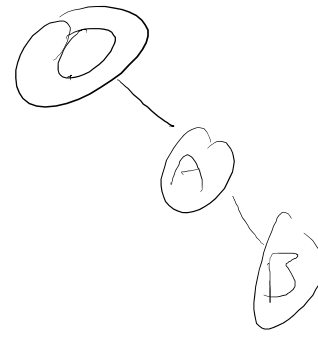


update heights
Balance Factor



Splays

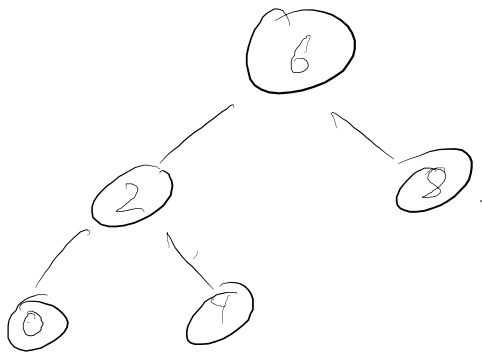
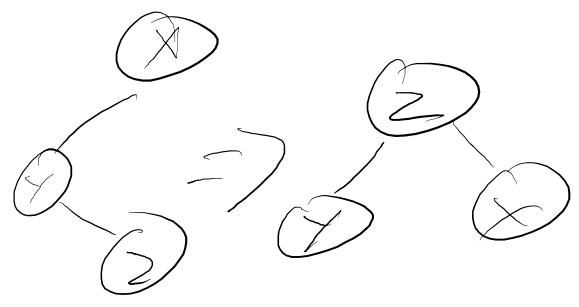
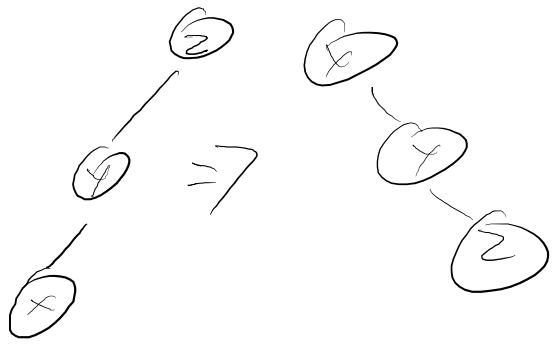
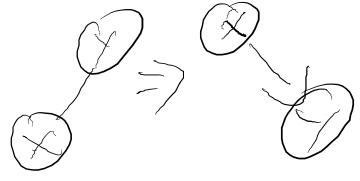
$\rightarrow O(n)$
amortized $O(\log n)$



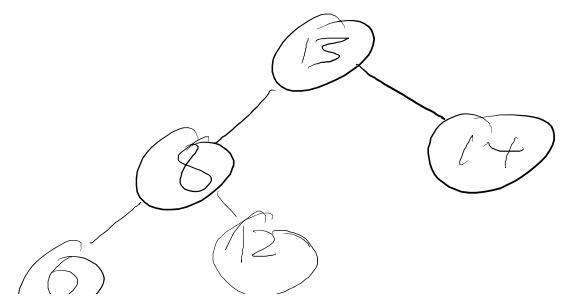
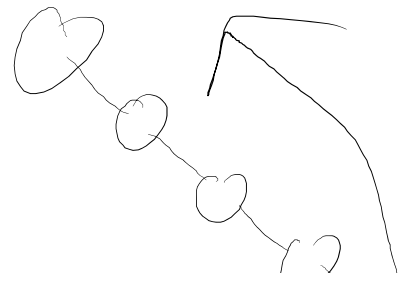
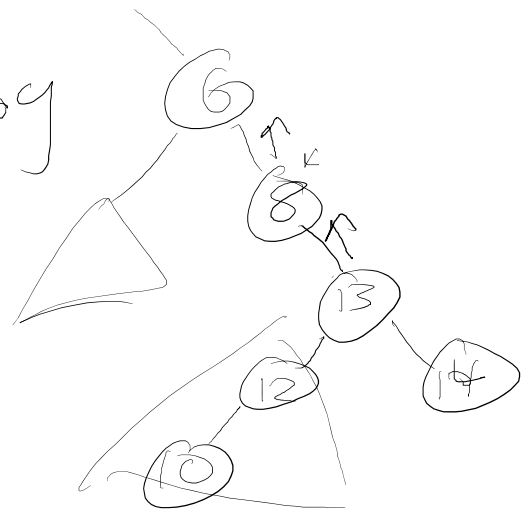
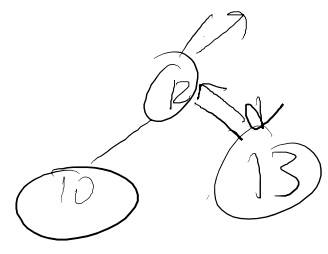
Zig(x)

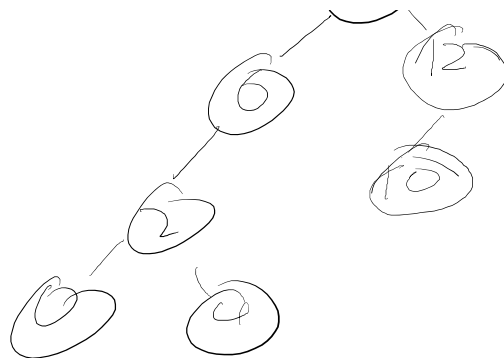
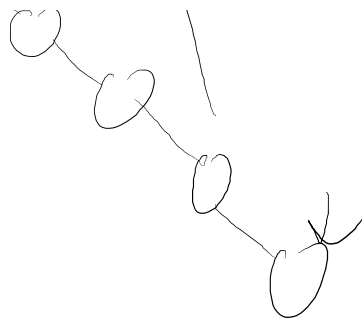
Zig-Zig(x)

Zig-Zag(z)



Zig-Zag



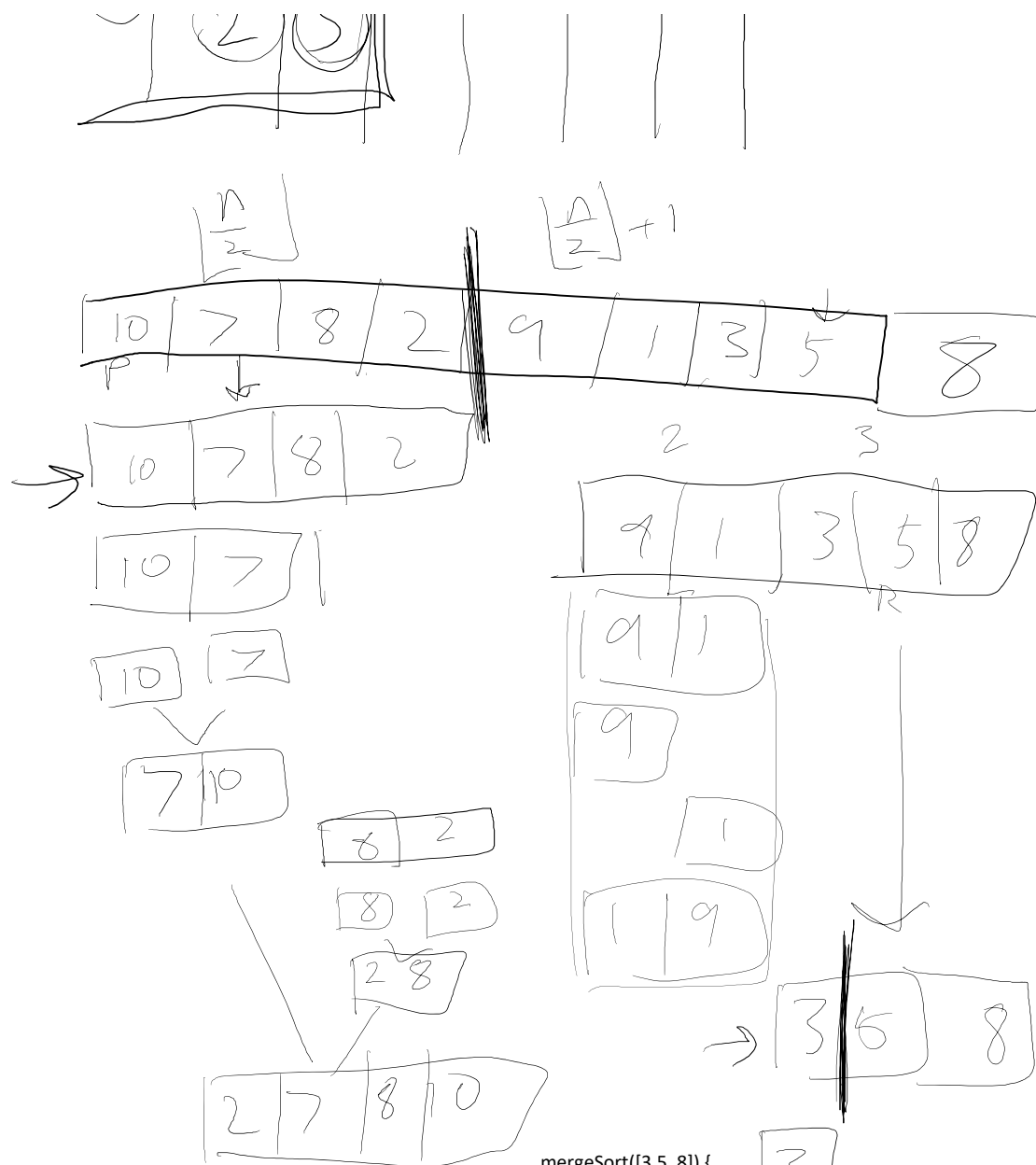


- Stability - duplicates stay in order
- inplace - no extra lists or arrays
- adaptive - ends early could be faster than $O(n \log n)$
- ~~online - not having to resort the entire list if new data is added.~~

	Selection	insertion	Bubble
Stable	X	✓	✓
inplace	✓	✓	✓
adaptive	X	✓	✓

Quick Merge





```
mergeSort([3,5, 8]) {
  L = MergeSort([3, 5])
  R = MergeSort([8])
```

```
}
```

