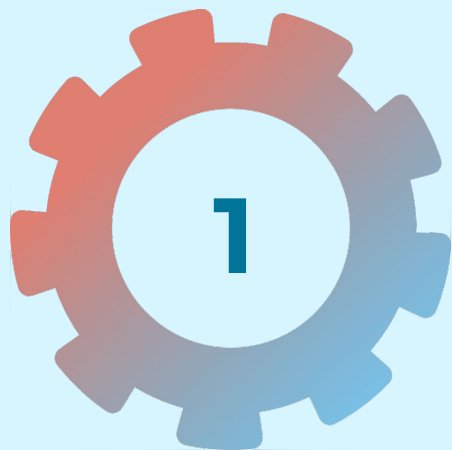


# Welcome!

Bit by Bit Week 3





# This Weeks Goals

What we will be working on

# 4 Pillars to Work on

1

## LeetCode

- Binary Search Tree Learn
- Binary Tree Learn

2

## Cracking the Coding Interview

- Trees and Graphs

3

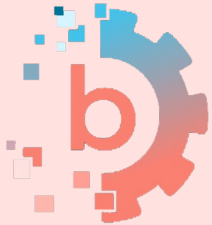
## Behavioral Question

- "Tell me about a personal project"

4

## Application Material

- Networking emails



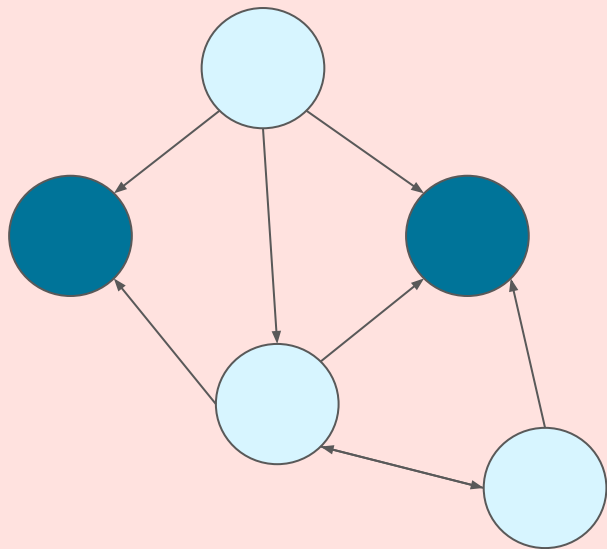


# Technical Review

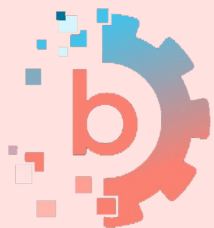
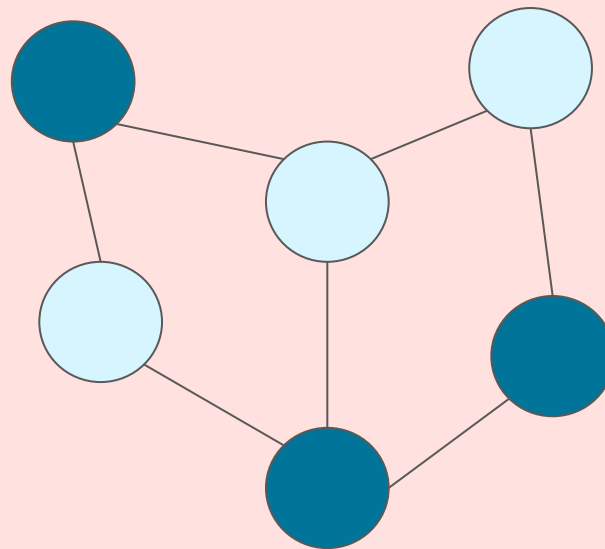
Thinking about Trees and Graphs

# Graphs

Directed (Instagram)



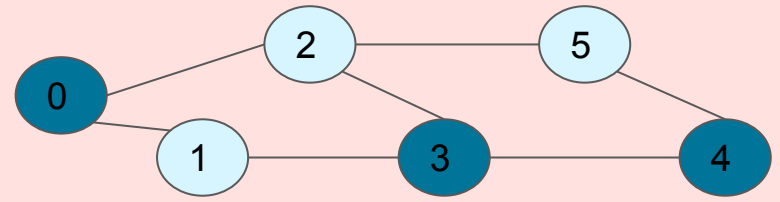
Undirected (Facebook)



Graph Key Words: (We can use DFS or BFS!)

- Locations
- Relationships
- Paths

# Graph Implementation

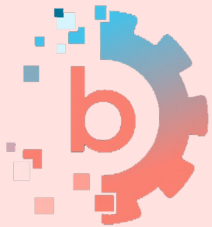


Adjacency Matrix

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

Edge List

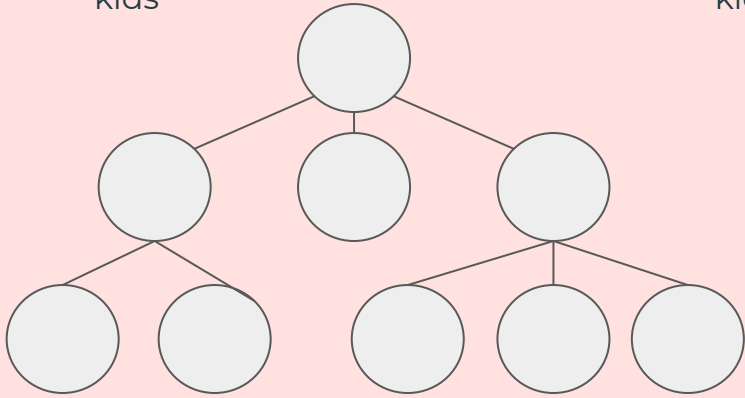
```
[  
  0 [1, 2],  
  1 [0, 3],  
  2 [0, 3],  
  3 [2, 4],  
  4 [3, 5],  
  5 [2, 4],  
]
```



# Three types of Trees

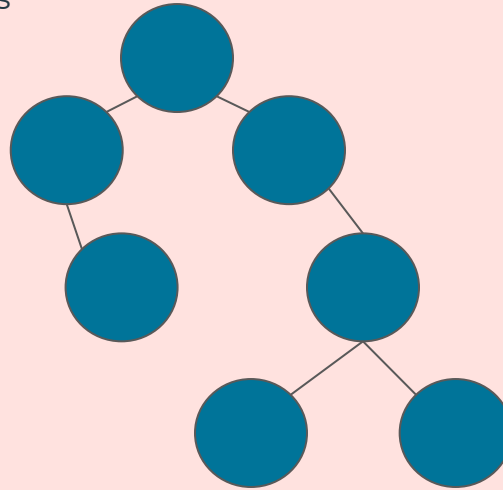
## M-ary

Every node has no more than M kids



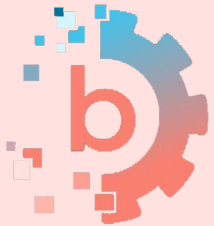
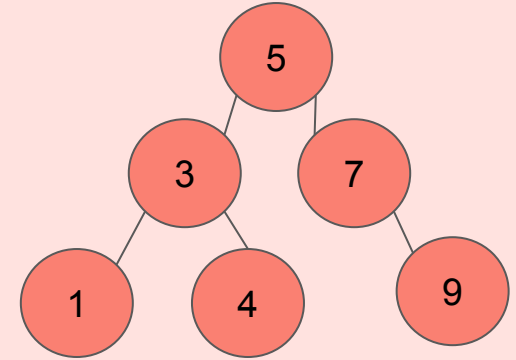
## Binary Trees

Every node has no more than 2 kids



## Binary Search Trees

Same as Binary Tree but all the left children less than the node and the right children greater than the node

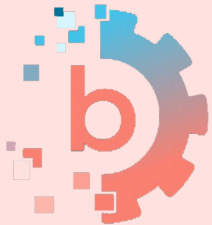


# Binary Trees

- Can have up to two children
- Typical node class has a left, a right and a data value
  - Always important to check that the children are not null!!
- Many important questions coming up to think about when approaching trees

Action	Runtime
Access Element	$O(n)$
Remove element	$O(n)$
Add element	$O(n)$
Searching	$O(n)$

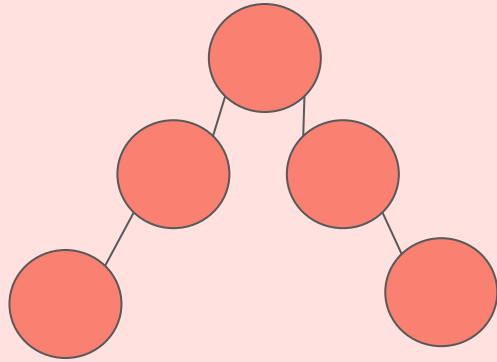
Run Time Cheat Sheet





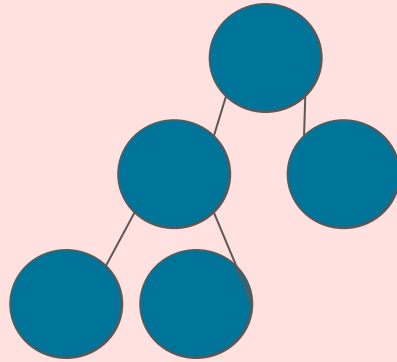
# Important Tree Clarifying Questions

*Balanced Tree?*



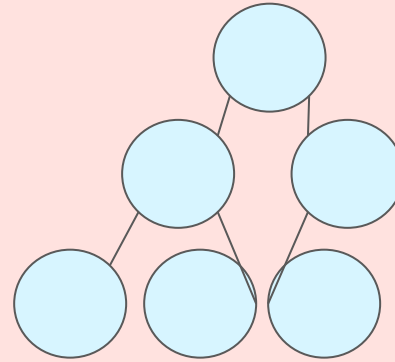
Difference between the left and the right subtree for any node is not more than one

*Full Tree?*



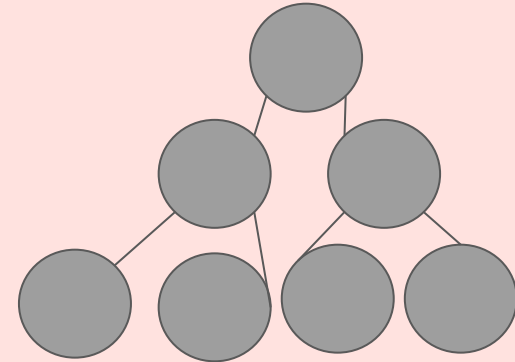
A Binary Tree is full if every node has 0 or 2 children.

*Complete Tree?*

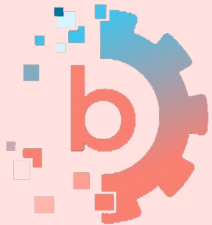


Every level, except the last, is completely filled, all nodes are as left as possible.

*Perfect Tree?*



All internal nodes have two children and all leaves are at same level.



# Tree Traversal Methods

- Top-Down Solution (Pre-Order Traversal)

```
Return if root is null
```

```
Update answer if necessary
```

```
Left_ans = function(root.left, left_params)
```

```
Right_ans = function(root.right, right_params)
```

```
Return answer
```

- Bottom-Up Solution (Post-Order Traversal)

```
Return if root is null
```

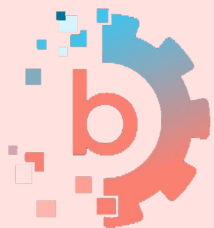
```
Left_ans = function(root.left)
```

```
Right_ans = function(root.right)
```

```
Return answer // answer = left_ans + right_ans  
+ root.val
```

**Are you passing something down the tree?**

**Can you calculate specific values as you move up the tree?**



# Common Types of Tree Problems

## Check if Valid / Is this true for the whole tree?

- base case/processing
- isLeftTrue = recurse(left)
- isRightTrue = recurse(right)
- return if this node is true && isLeftTrue && isRightTrue

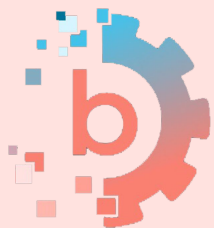
Example: Is this tree a binary search tree?

## Need to build a new tree

- base case/processing
- node.left = recurse(something); node.right = recurse(something);
- return the root

Example: Return a new tree where every node value is doubled.

**Note: Helper Functions are your friend!!!** Any time we need to return another data structure and are doing recursion (aka return a list with all the nodes in a tree that are greater than 0 or return a new tree) we want to have a helper function!



# Binary Search Tree

- Left nodes are all less than the current node.
- Right nodes are greater than the current one
- We want to keep the tree balanced
- Great for Sorting
- Always clarify how to deal with duplicate values

Action	Average	Worst
Access Element	$O(\log(n))$	$O(n)$
Remove element	$O(\log(n))$	$O(n)$
Add element	$O(\log(n))$	$O(n)$
Searching	$O(\log(n))$	$O(n)$

Run Time Cheat Sheet



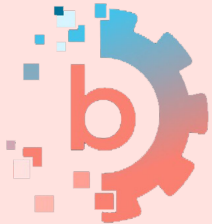


# Behavioral Tips

Projects and Networking

# “Tell me about a personal project”

- Find a project you are passionate about
- Project can be from
  - Class
  - Hackathon
  - Something else you worked on in your spare time
- This is a technical project
- Share tech stacks used or other relevant technical info



# My personal “personal project”

## General Context

I was in a class where we did XYZ and got to make an app

## “The Problem”

We knew there was a problem of XYZ on campus so we decided we were going to make a program to solve it

## The Solution

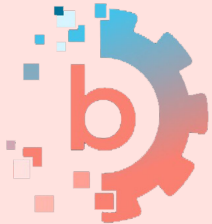
What I built and how it solves my problem

## Technical Details

We built the project with React Native

## Result

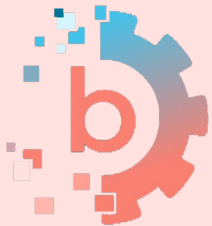
How many people use the project?  
Where is it now?



# Networking - key steps

## GOAL: To get a referral!

1. Finding companies you are interested in
2. Finding individuals you want to talk to at said companies
  - a. Recruiters
  - b. Engineers
3. Emailing individuals asking to learn more
4. Speaking on the phone
5. Asking for referrals



Annika,

Thanks for reaching out! Looks like you were in our system twice- once as a referral, and again as an application submittal not including a referral. You are still in the interview process. Best of luck on the HackerRank test!

-Sarah



# Networking - how to find companies

## Field / Impact

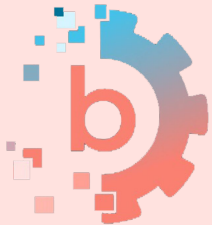
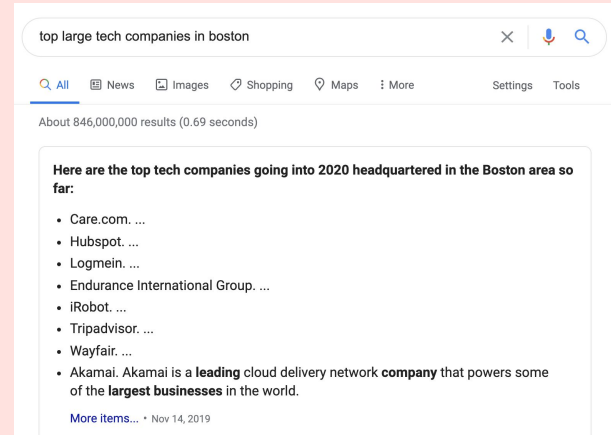
- <https://www.startus-insights.com/>
- Angellist
- Generally just Google

## Location

- Angellist, LinkedIn, or general Google Searches.

## Size

- Small companies tend to be on LinkedIn, Angellist
- Medium Companies may have funding so it's worth looking at different Venture Capital or Growth Equity portfolios
- Large companies you probably know of already



# Networking - how to find individuals

## LinkedIn

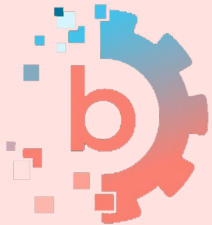
- Search by school
- Search by location
- Either directly LinkedIn them or guess their email
- Rocket Reach is great for email guessing

## School Alumni

- Differs by school
- Typically sends email directly to the alumni so no need to guess
- Creates immediate strong connections

## Other Network Groups

- Any other communities you are a part of?
- High school connections
- Friends of friends?
- Specific clubs?



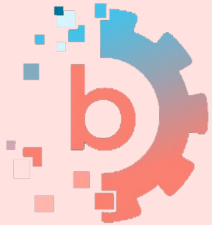
# Networking - recruiters vs engineers

## Recruiters

- Ask about application information
- Ask about timelines
- Ask about feedback

## Engineers

- Ask specific company questions (culture, job, coding, etc)
- CAN (and want to) refer you!



# Networking - sample emails



Morgan R. Sorbaro

Mon 7/16/2018 4:14 PM

To: cameronwhouser@gmail.com



Hi Ms. Houser,

My name is Morgan Sorbaro and I am a rising junior at Dartmouth College. I have been interested in coding since early high school and am now a computer science major at Dartmouth. Looking towards the future, working for Airbnb particularly excites me and I would love to hear more about the internship opportunities. Due to Dartmouth's quarter schedule I am looking for not only a summer internship in 2019 but also a winter 2019 internship (January - March) and was wondering if Airbnb had opportunities during both or either of those times. If so, what does the application timeline look like? Thank you for the information and I hope to be able to apply soon.

Best,  
Morgan Sorbaro

Email to a recruiter

On Mon, Jul 30, 2018 at 1:43 PM Morgan R. Sorbaro

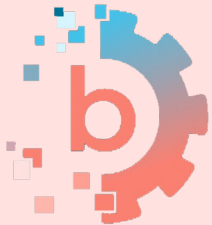
<[Morgan.R.Sorbaro.20@dartmouth.edu](mailto:Morgan.R.Sorbaro.20@dartmouth.edu)>  
wrote:

Hi Sophia,

My name is Morgan Sorbaro and I am a Dartmouth '20 studying computer science. I am interested in applying to Google for the software engineering winter internship and if you have some free time I would love to hear about your experience working there so far. I am in the middle of sophomore summer and my schedule is very flexible so let me know what works best for you!

Thanks,  
Morgan Sorbaro

Email to an Engineer





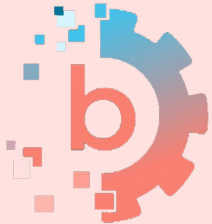
# Mock Interviews

Frameworks and booking

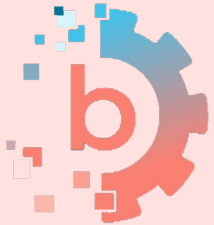
# If you have no idea....

What **do** you know?

- Explain why you are stuck
- Ask for feedback on potential directions you could go
- Go through list of Data Structures - which could work?
- Explain what you know about the chosen data structure and then explain if that helps or not in the problem
- List all assumptions you have made about the problem
- Try to think about it “like a human”

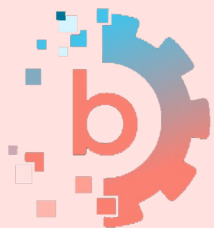


# Google Mock Interview Example



# Key Takeaways

- He asks creative clarifying questions
- Starting with brute force analysis
- Walking through examples
- Always trying to communicate
- Talking while coding - pauses to explain exactly why he is doing every line!







# Questions

What didn't make sense?