LeetCamp

A project dedicated to DS&A

Agenda

1. Introductions
2. What is LeetCamp?
3. Why study DS&A?
4. Solve Problems!

Introductions

Andrew Fennell

- Class of 2022 - Graduating in December
- Computer Engineering major
- Interned at Dell and CACI @ NASA

Me (real) practicing LeetCode

Patrick Apgar

- Class of 2023

- ESET major
  - Minor in Comp Sci and Cybersecurity

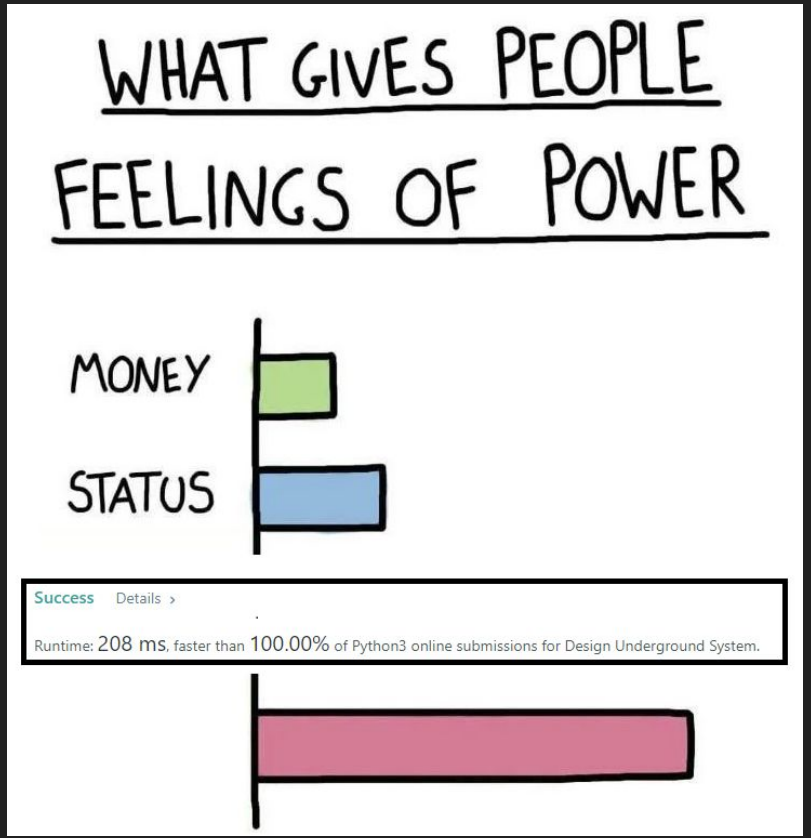
- Interned at IBM and Accenture

How I look when I start practicing LeetCode

What is LeetCamp?

What is LeetCamp?

- A dedicated time to learn about Data Structures and Algorithms (DS&A)
- Problem solving group!
- Learn to crush your technical interviews!



What is LeetCamp?

- You can't be 100% prepared for any question in a technical interview.

- You can take the right approach to preparation to figure it out on the spot!

Why study DS&A?

Technical Interviews

- Technical interviews are how companies evaluate their candidates' technical abilities
- Typically LeetCode questions
- DS&A and Object-Oriented Design questions

- Don't worry! If this sounds scary, that's why we are here right now!

Think Differently

- DS&A mindset
  - Ask the right questions
  - Asking feasibility questions

- Optimization mindset
  - Designing good solutions

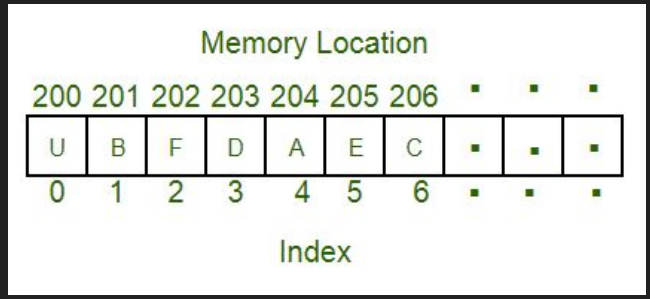
- Gain a deeper level of understanding about your solutions

What are Data Structures?
What are Algorithms?

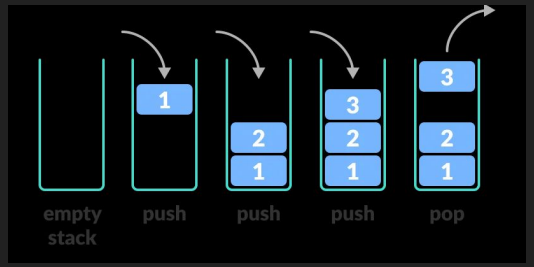
Data Structures

GeeksForGeeks: *“A data structure is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently.”*

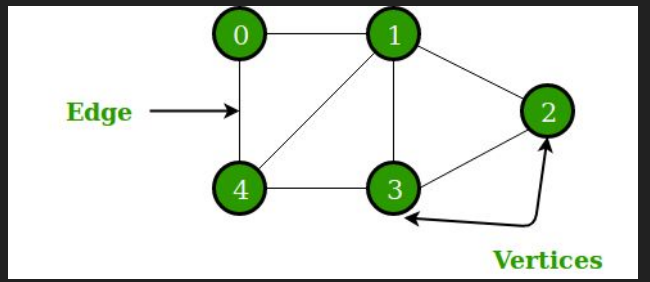
- A represent data
- A way to efficiently store and retrieve data



Array



Stacks



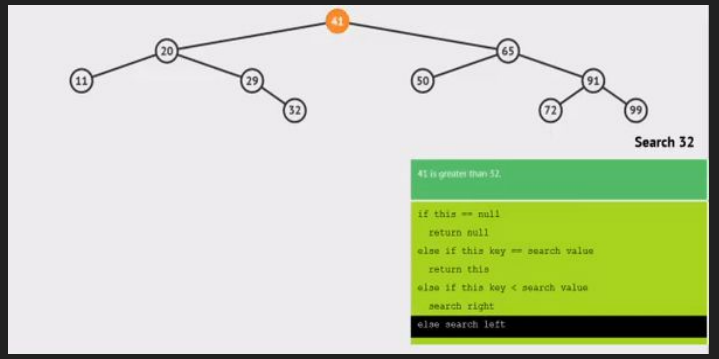
Graphs

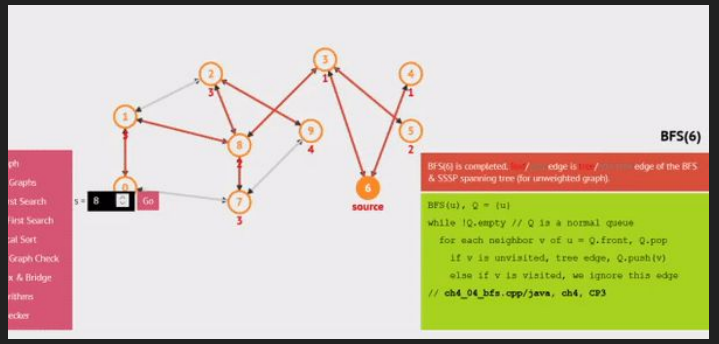
Algorithms

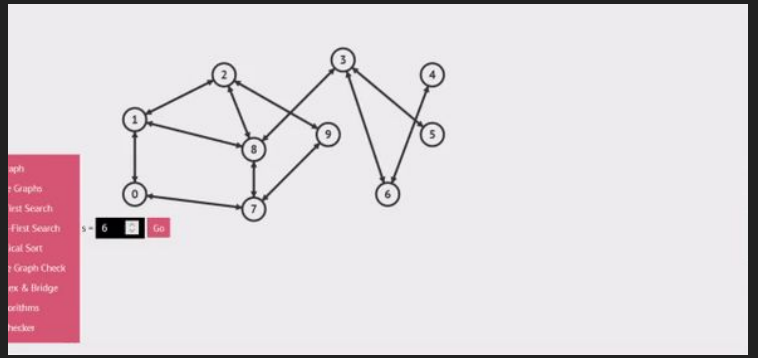
GeeksForGeeks: “*A set of rules to be followed in calculations or other problem-solving operations.*”

OR “A procedure for solving a mathematical problem in a finite number of steps that frequently by recursive operations.”

- A set of steps that lead to a solution
- Efficiency is key
- Many types of algorithms







Let’s solve some problems!

Problem 1 - Two Sum

- What is the first solution that comes to mind?
  - Is this the optimal solution?
  - How do you know?
    - Are we using everything given to us?

- Is there a data structure that we can use to solve this problem?
  - What about a hashmap (Python “dictionary”)?
    - Can recall previous values and associate it with an index
  - Would this help improve time complexity?

- Let’s implement our solution!

Problem 2 - Running Sum of 1d Array

- What is the problem asking for?

- What's your initial approach to solving the problem?

- Is there any way to optimize? Could we perhaps do this in O(1) space?

Problem 3 - Valid Anagram

- What is the first solution that comes to mind?
  - Is this the optimal solution?
  - How do you know?
    - Are we using everything given to us?

- Is there a data structure that we can use to solve this problem?
  - What about a hashmap (Python “dictionary”)?
    - Key = letter -> Value = count

- Let’s implement out solution!

Problem 4 - Is Subsequence

- Initial approach to the problem?

- Are you thinking of using nested for loops?

- Could there perhaps be a more efficient way to solve the problem?
  - Can we solve this in O(1) space in one pass?

- What about the two pointers approach?

Until next time...
Start practicing

Practice Problems

- Get started here:
  - Contains Duplicate
  - Valid Palindrome
  - Roman to Integer

- A bit more challenging:
  - Two Sum II
  - Longest Substring w/o Repeating Characters