LeetCamp

A project dedicated to DS&A

Agenda

1. Introductions (again)
2. Time Complexity
3. Last Week’s Solution
4. New Data Structure
5. Let’s Solve Some Problems!

Introductions (again)

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

Time Complexity

What is Time Complexity?

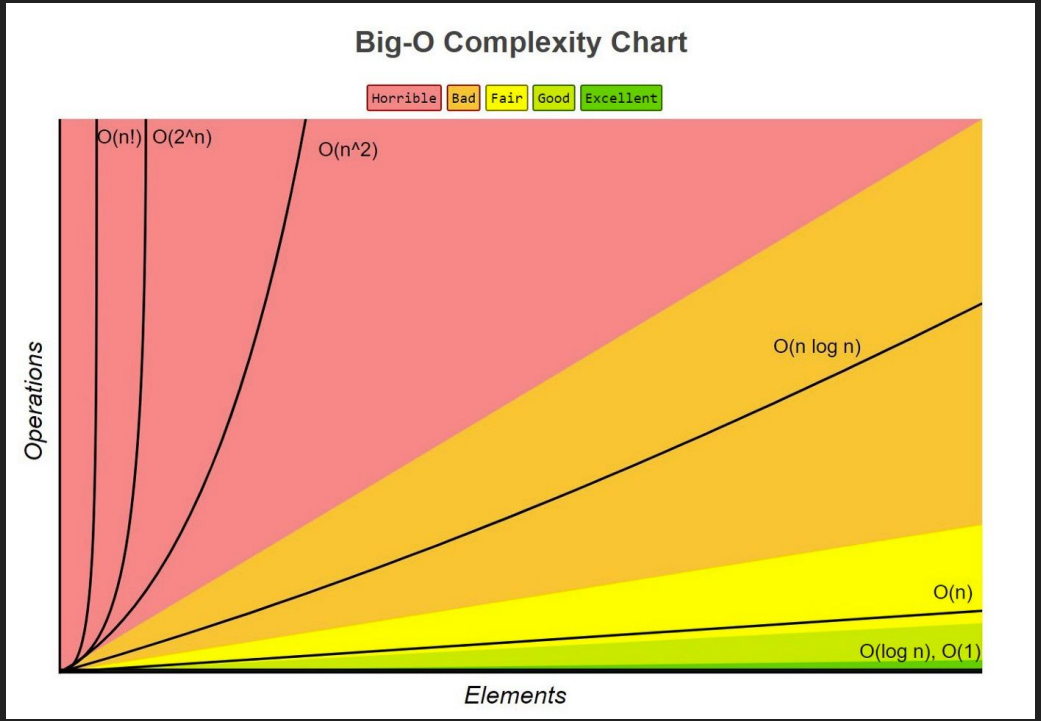
- How we measure the performance of your algorithms
- It's necessary how well, or how poorly, your code performs as more data is provided for the algorithm to process
- Various levels of complexity:
  1. O(1)
  2. O(log N)
  3. O(N)
  4. O(k * log N)
  5. $O(N^2)$
  6. $O(2^N)$
  7. O(N!)

Time Complexity - Examples

Imagine a classroom of 100 students in which you gave your pen to one person. You have to find that pen without knowing to whom you gave it.

- **$O(n^2)$**: You go and ask the first person in the class if he has the pen. Also, you ask this person about the other 99 people in the classroom if they have that pen and so on,
- **O(n)**: Going and asking each student individually is O(N).
- **O(log n)**: Now I divide the class into two groups, then ask:
  - “Is it on the left side, or the right side of the classroom?”
  - Then I take that group and divide it into two and ask again, and so on.
  - Repeat the process till you are left with one student who has your pen.
- O(1): Yelling....
  - You yell: “WHO HAS MY PEN?”
  - The person yells back: “ME RIGHT HERE!”

Time Complexity - Chart



https://www.bigocheatsheet.com/

Last Week’s Solutions

Practice Problems (given last week)

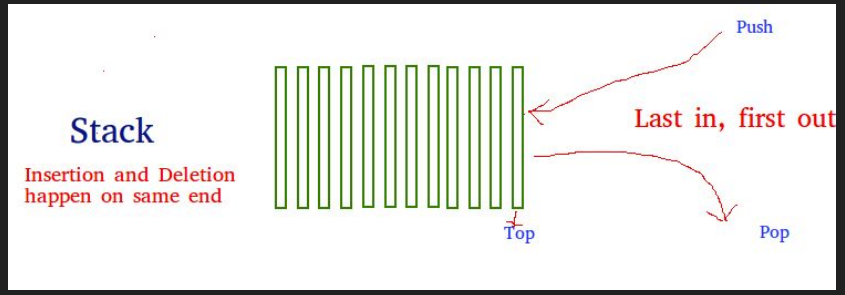
- Get started here:
  - Contains Duplicate - Solution
  - Valid Palindrome - Solution
  - Roman to Integer - Solution

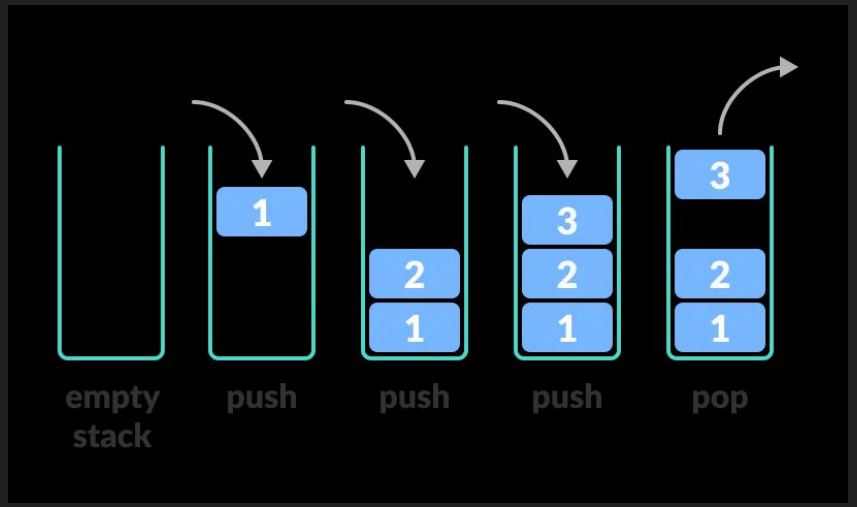
- A bit more challenging:
  - Two Sum II - Solution
  - Longest Substring w/o Repeating Characters - Solution

New Data Structure:
Stacks

What is a stack?

- Stack is a linear data structure which follows a particular order in which the operations are performed
- Remember stacks with FILO (First In Last Out)
- https://www.geeksforgeeks.org/stack-data-structure/





Let’s solve some problems!

Problem 1 - Valid Parenthesis

- 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 stack?
  - Would this help improve time complexity or space complexity?

- Let's implement our solution!

https://leetcode.com/problems/valid-parentheses/

Problem 2 - Backspace String Compare

- 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?
  - Spoiler: We can, but it doesn't use a stack!

https://leetcode.com/problems/backspace-string-compare/

Problem 3 - Reverse Polish Notation (In Groups!)

- 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 stack?
  - Would this help improve time complexity or space complexity?

- Let's implement our solution!

https://leetcode.com/problems/evaluate-reverse-polish-notation/

Until next time...
Keep practicing

Practice Problems

- Get started here:
  - Remove All Adjacent Duplicates in String
  - Implement Queue Using Stacks (Relevant to CSCE 221!)
  - Remove Outermost Parentheses

- A bit more challenging:
  - Minimum Remove to Make Valid Parenthesis

- If you want an even bigger challenge:
  - Design a Text Editor: https://leetcode.com/problems/design-a-text-editor/
  - Parsing a Boolean Expression: https://leetcode.com/problems/parsing-a-boolean-expression/