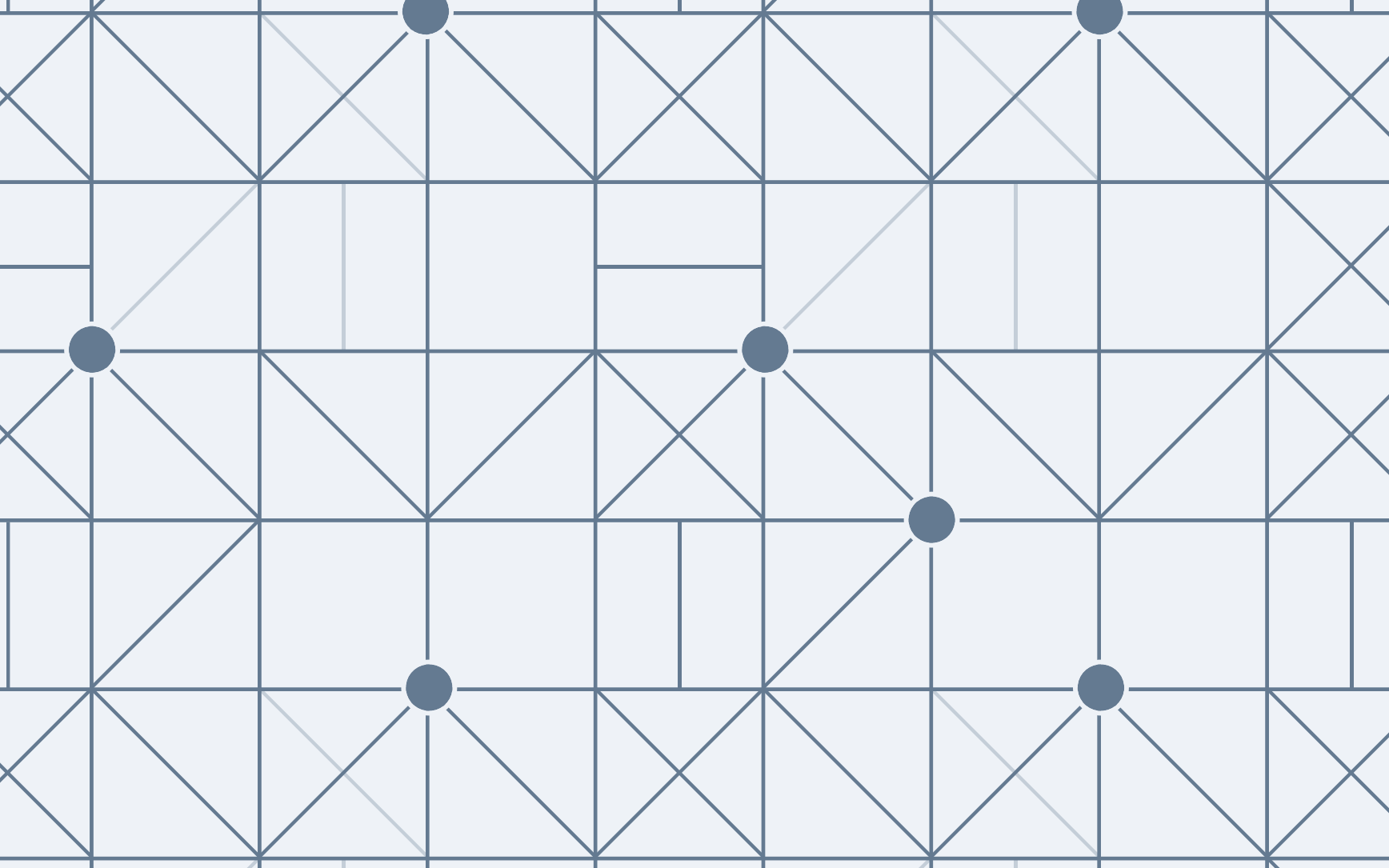


Ritik Ramuka (M3 instructor)

Microsoft (SWE) { Powerlange Rodev }

- tier 3 college.
- IPU MSIT - 2019 - 2023
- Born Delhi
- Currently in Blr. }

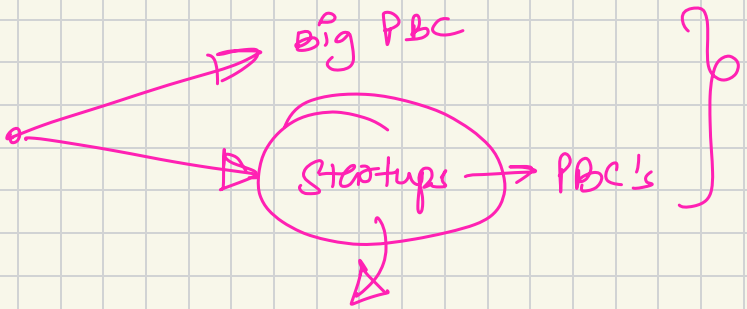
→ 3.5 yr teaching experience.

- PepCoding { DSA, DS/ML, Blockchain }
- Accio Jobs { 1.5 yrs, DSA, PM }
- BossCoder { 6 m, DSA }

- twice intern at MS.
- 2022 inter { application mail }
- freelancer
- gold medalist (Rank 5 Delhi) in maths olympiad.

2018 → year drop → joined college {2019-23}
BTech CSE

→ {FTE Microsoft, Amazon, startups}



- right set of skills are important to focus on.

Trends → Skills

Fast Pace

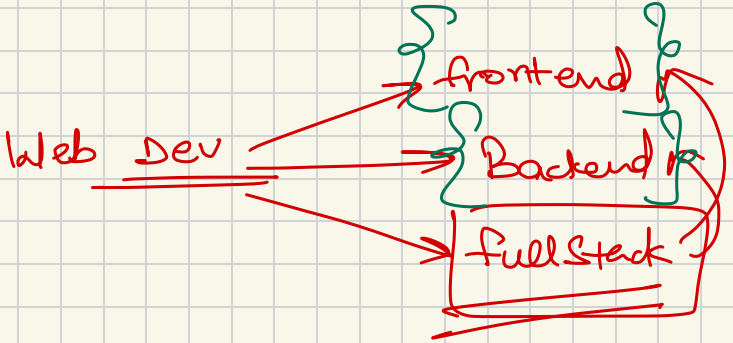
Top PBC's

- DSA
- Basic of Dev
- Basic
  - OOPs
  - CN
  - OS
  - DBMS

Startups

- Basic DSA
- Dev

Solve Problems → something new



Module 3 Syllabus

- DS ◦ Stacks (3)
- DS ◦ Queues (2)
- Algo ◦ BinarySearch (3)
- DS ◦ Hashing (4)
- DS ◦ Binary Trees (4)
- ◦ Revision (2)

18

Module 3

LC | gfg

- 60
- 20 (Hard)
- 30 (Med)
- 10 (easy)

} 60

◦ **Data Structures**

- efficient way to store data

Algorithm

-

Cook tea

Home · steps

- Boil Milk
- add-tea leafs, sugar, ginger

brain · steps

- Hot water
- milk powder, tea bag

Approach Problem

Q → Easy {10 min}

→ Medium {30 min} – 45 min

→ Hard {30 min – 1 hr}

see a sol^n.

dry run

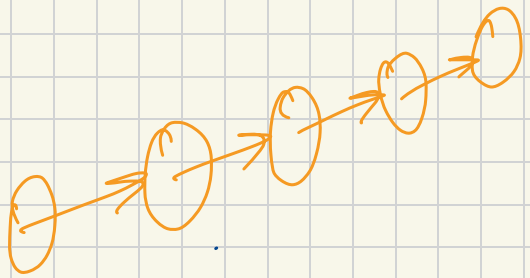
given + what? + {try figuring out DS | Algo}

Brute force → Approach → Basic edge case

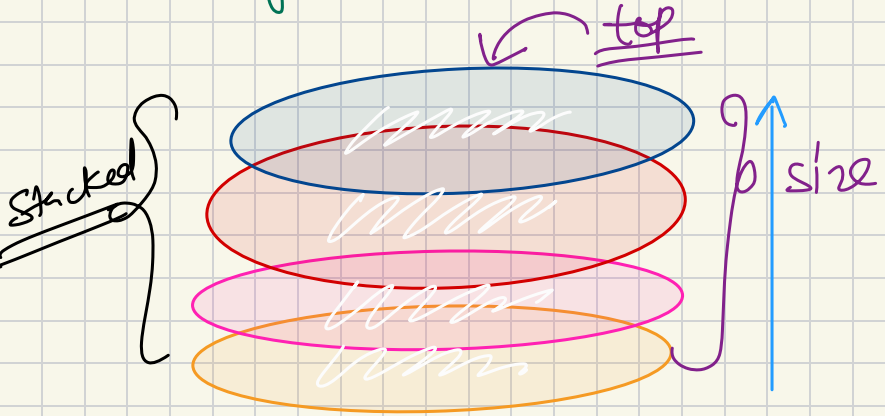
→ try to optimize this

Stack Data Structure { Linear Data Structure }

- arrays
- arraylist
- LinkedList
- doubly LinkedList
- Strings
- StringBuilders



Washing some dishes



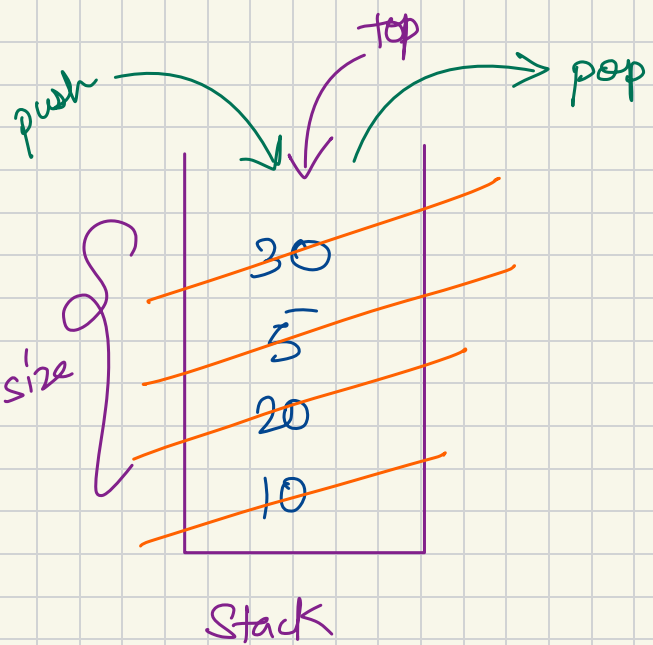
order of use

order of washing

(LIFO)

↳ last in, first out.

Last dish washed, is the first to be used.



pushed (10)
pushed (20)
pushed (5)
pushed (30)

pop() ⇝ 30
pop() ⇝ 5
pop() ⇝ 20
pop() ⇝ 10

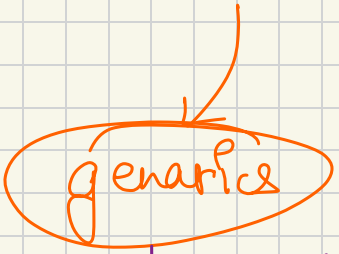
LIFO

{Last in, first out}

Stack

Class in Java

```
Stack<E> stack_name = new Stack<>();
```



→ OOPs

- Wrapper Classes & classes

Methods

- push(val) ⟶ adds the ele. at the top of the stack
- pop() ⟶ removes and returns topmost element of the stack.
- size() ⟶ No. of ele in the stack
- peek() ⟶ returns the value of the topmost ele. of stack.

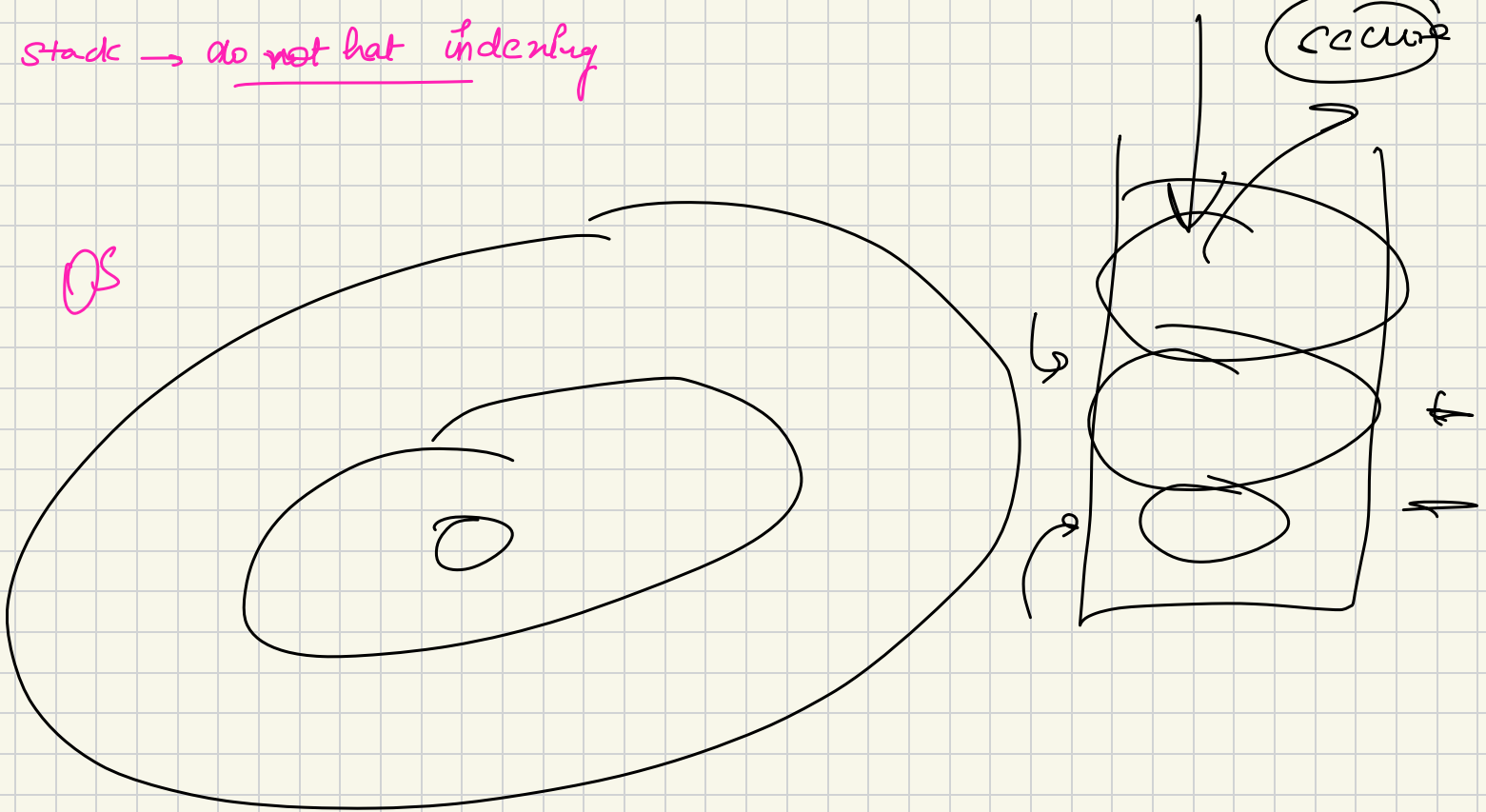
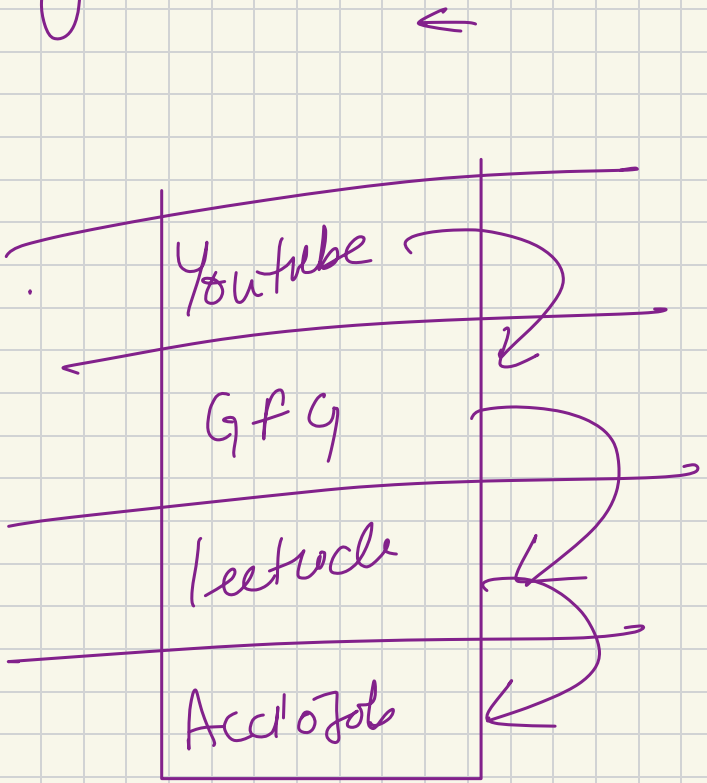
{ TC : O(1)
SC : O(1) }

Stack ⟶ Linear DS

↳ follows LIFO

↳ Methods: push, pop, peek, size } TC: O(1), SC: O(1)

stack → do not that indexing



- Browser History