

Starting at 9:05 PM

DARE TO DREAM

11:30 PM

Support @ scalar.com

agoda

① Dare to Dream

↳ Intro to DSA

Weeks

DSA

② 1 LD 101 ← Low level Design

9:12

1 st...



Naman Bhatia

SWE @ Google
Amazfit
Clueplex

Low Level Design

HLD Master Classes

~~2020 grad~~

+91 - 999620 8771

Agenda

→ Scalar Core Values

~~3rd Context~~ → How to remain motivated

→ Real world problem that uses DSA

→ AMA

← Doubt Session

Ask Me Anything

Performance
Equation

= Potential

Time Waste

Distr

Motiv

Procrast

Incent

Laziness

Interference

Performance = Potential - interference

- ① Dont distract Classes
- ② Dont open multiple tabs
- ③ Use earphones
- ④ BE WITH PRESENT

→ Support @ Scaler.com



② Good mentor interaction

↑ 9:41

①

Hard Work

②

Consistent

Support @ Scaler
.com



60 classes of DSA



It's over
6 months

⇒ 1/600 questions

96%
96%

LeetCode Weekly Contests

Initially people put a lot



1% Better Every day

$$T_0 \quad \textcircled{+1\%} \quad T_2$$
$$1 \rightarrow 1.01 \rightarrow 1.02$$

$$\sqrt[n]{(1.01)^{365}} \approx \sqrt[34-38]{}$$

100 → 100%

$$(0.99)^{365} \rightarrow 0.025$$

2.5%

↑ # 1-percent-beta

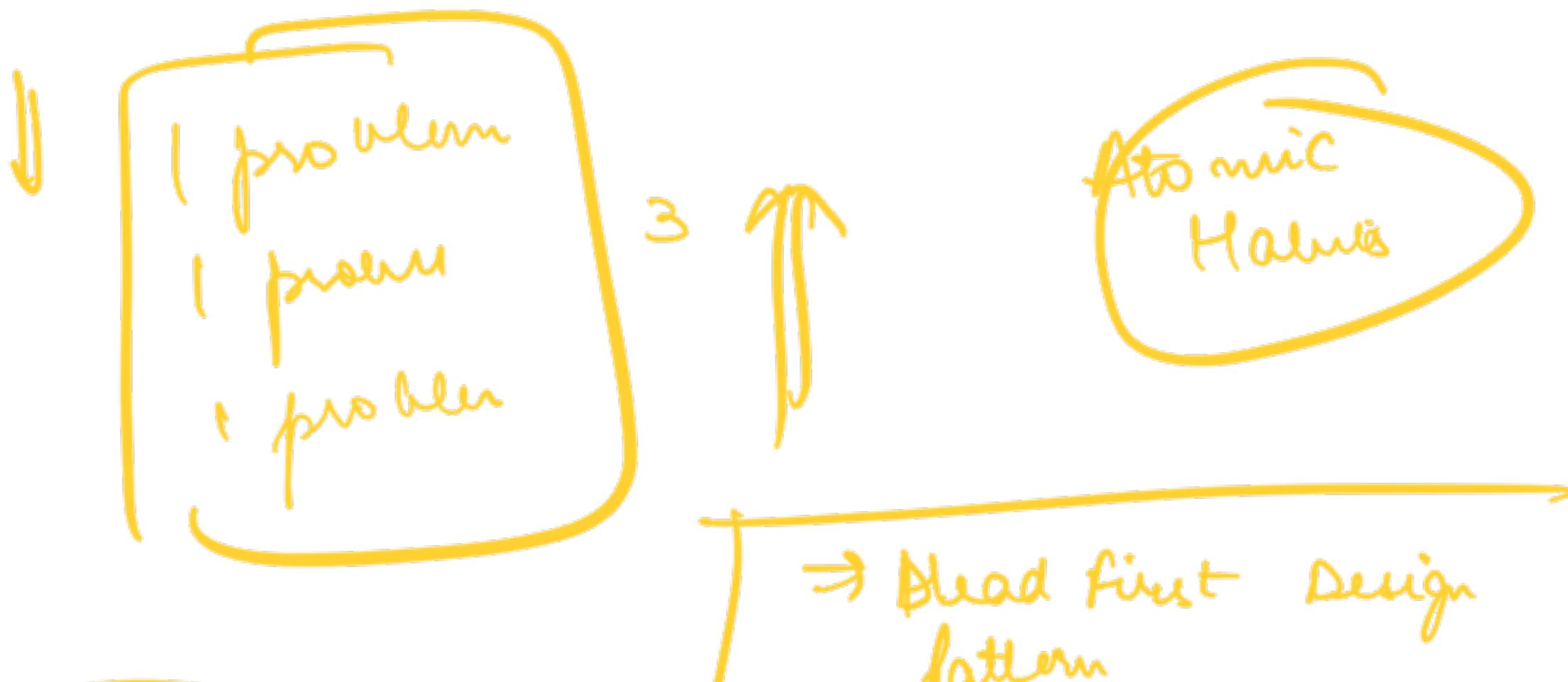
Problem Solving streak

365+ days

Motivation is temporary

- Giving motivation is a business
- → Motivation doesn't lead to action

Action leads to motivation



Streaks

→ Clean Code

6 days a week
3 hrs a day

→ Make small goals

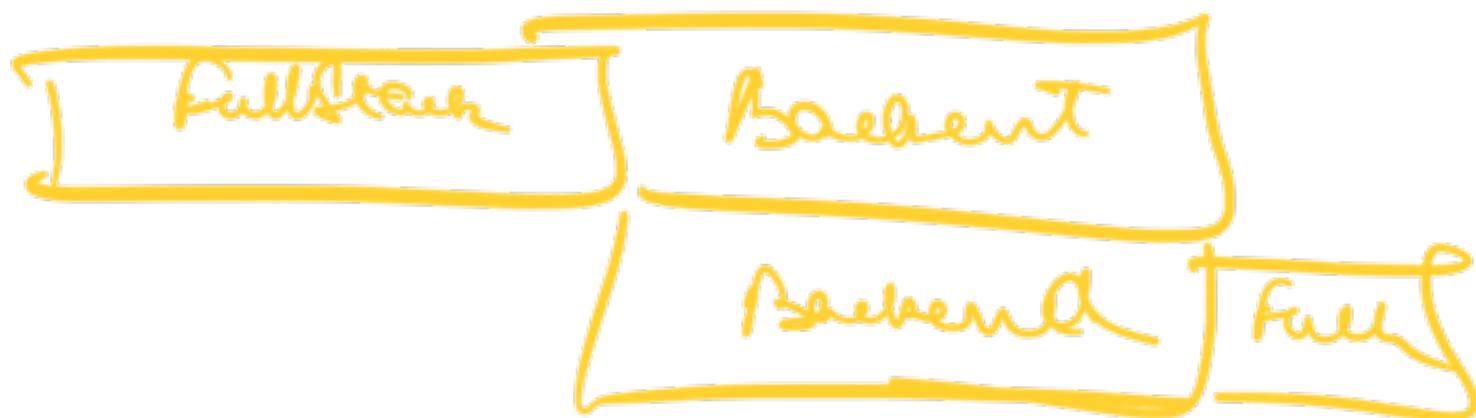
→ Be consistent

⌚ increase the goals

2-5kg \Rightarrow 1 week

5kg \Rightarrow 2 weeks

7.5 \Rightarrow 4w



II

Data Structures and

Algorithms

Recipe: Steps to cook a dish

Algorithm:

Steps to solve a problem

way to do something

Shortest Path Algorithm

→ way to find shortest path from A to B

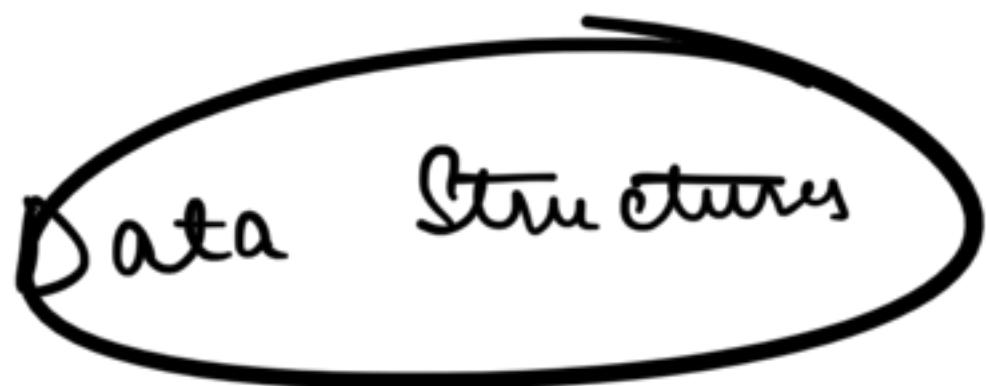
Searching Algo

Used to + search in files in our

→ Way to store data
array

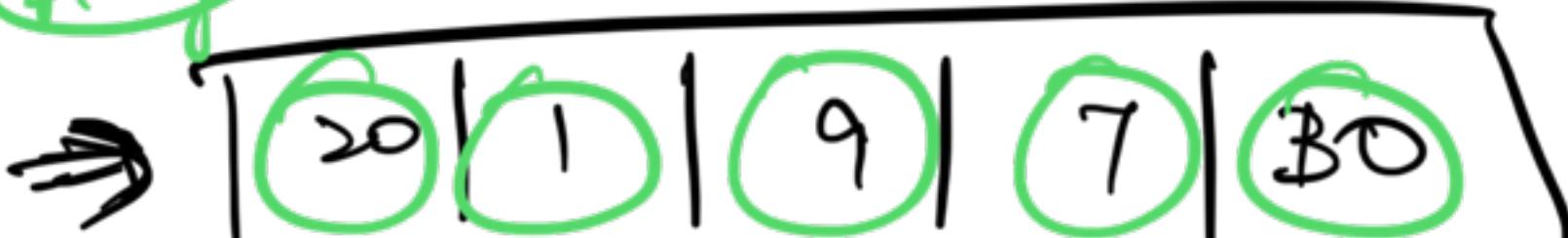
Algorithm

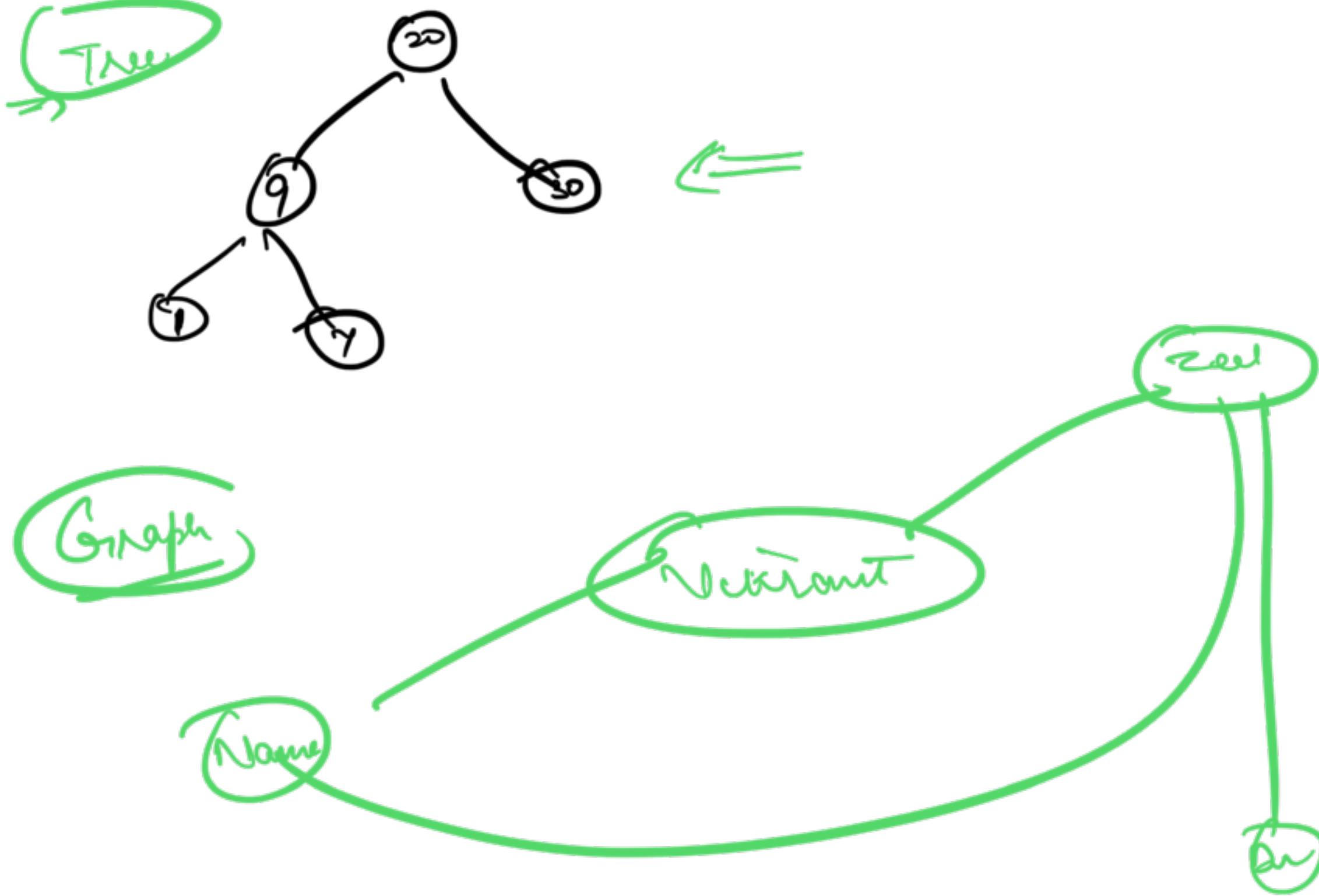
Step by Step approach to solve a problem



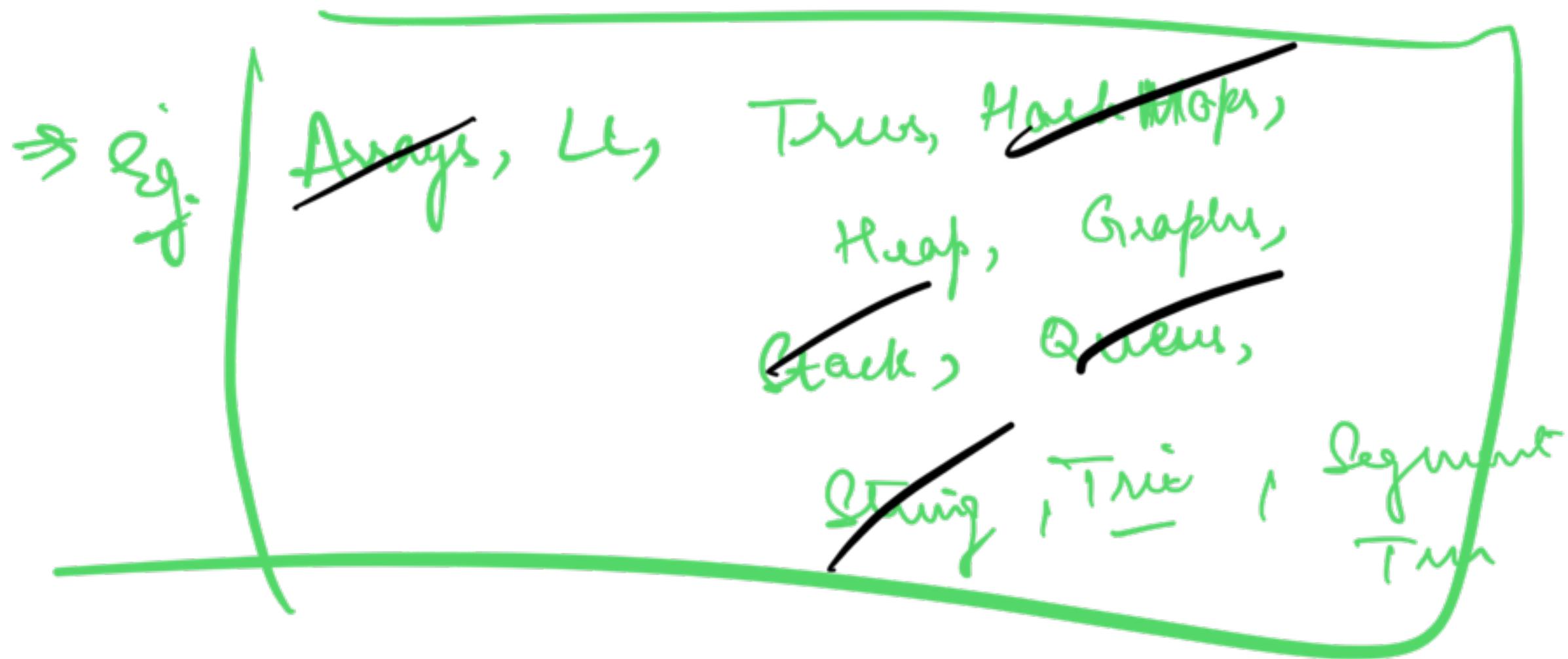
Way to store data

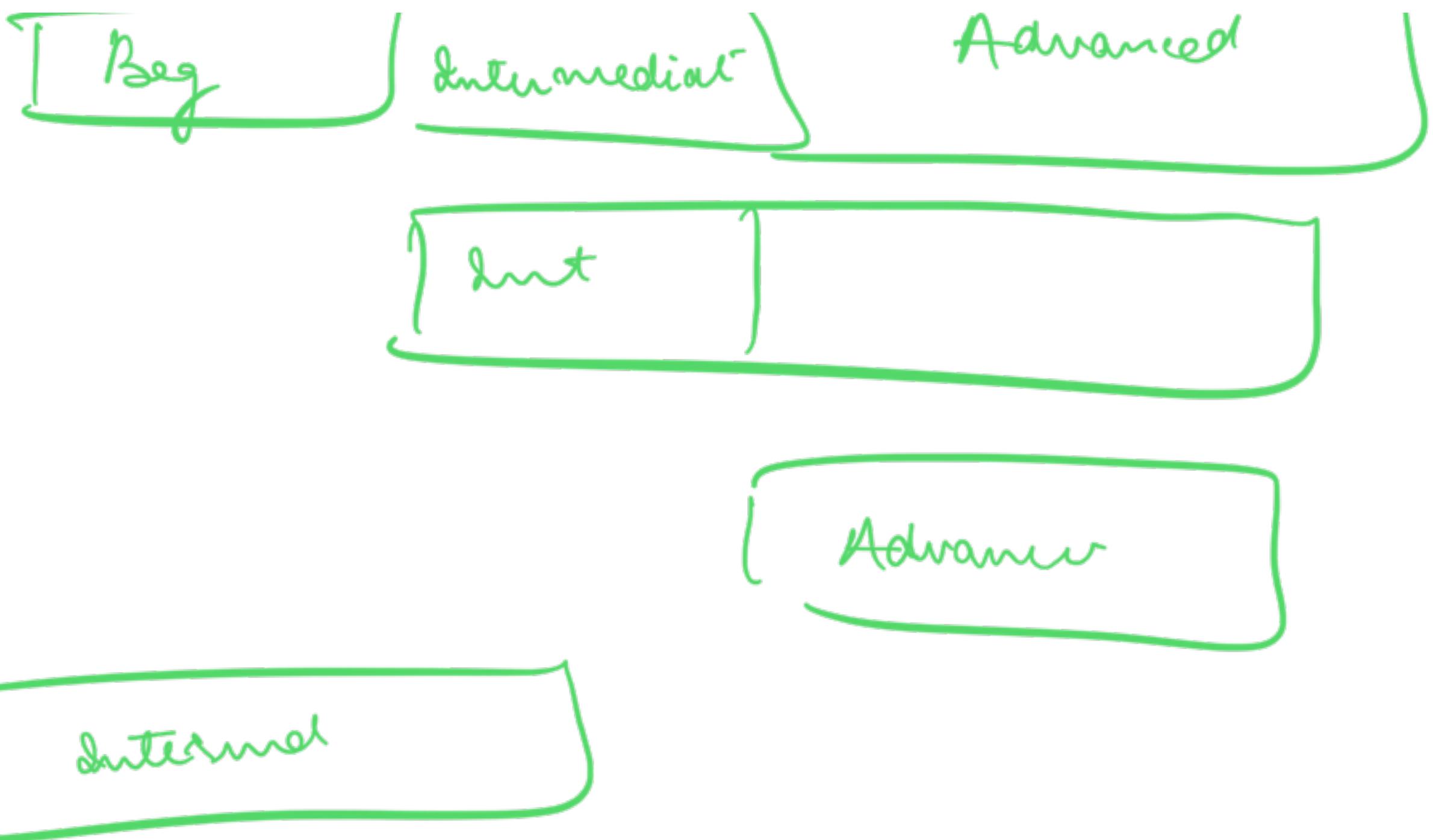
array





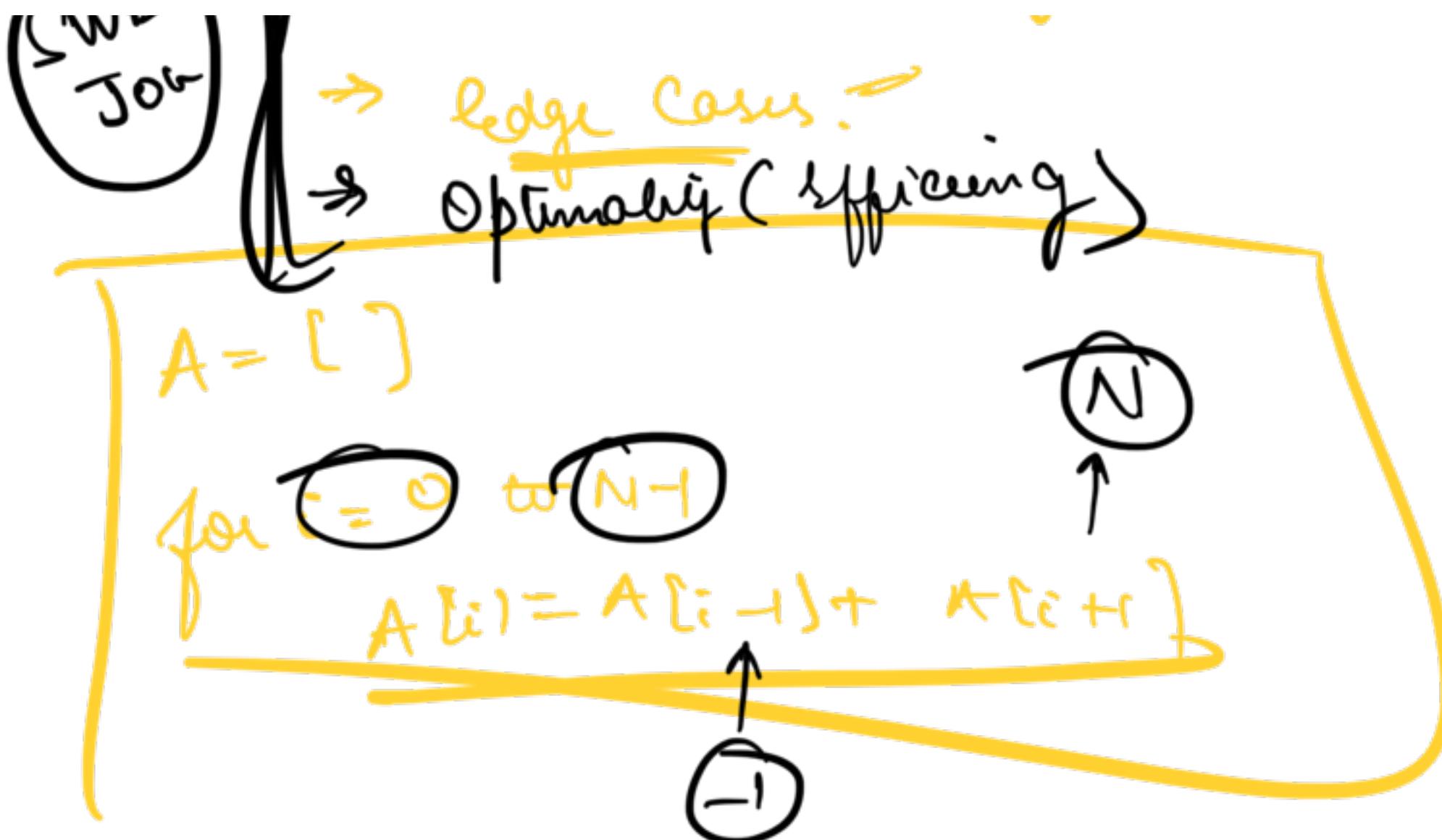
→ We will choose the data structure depending upon the type of operations we need to do on the data





Why DSA questions are asked

⇒ Problem Solving Skills =



9:00 AM

Very Simple Problem Statement

- Encountered Before
- Build from Basics

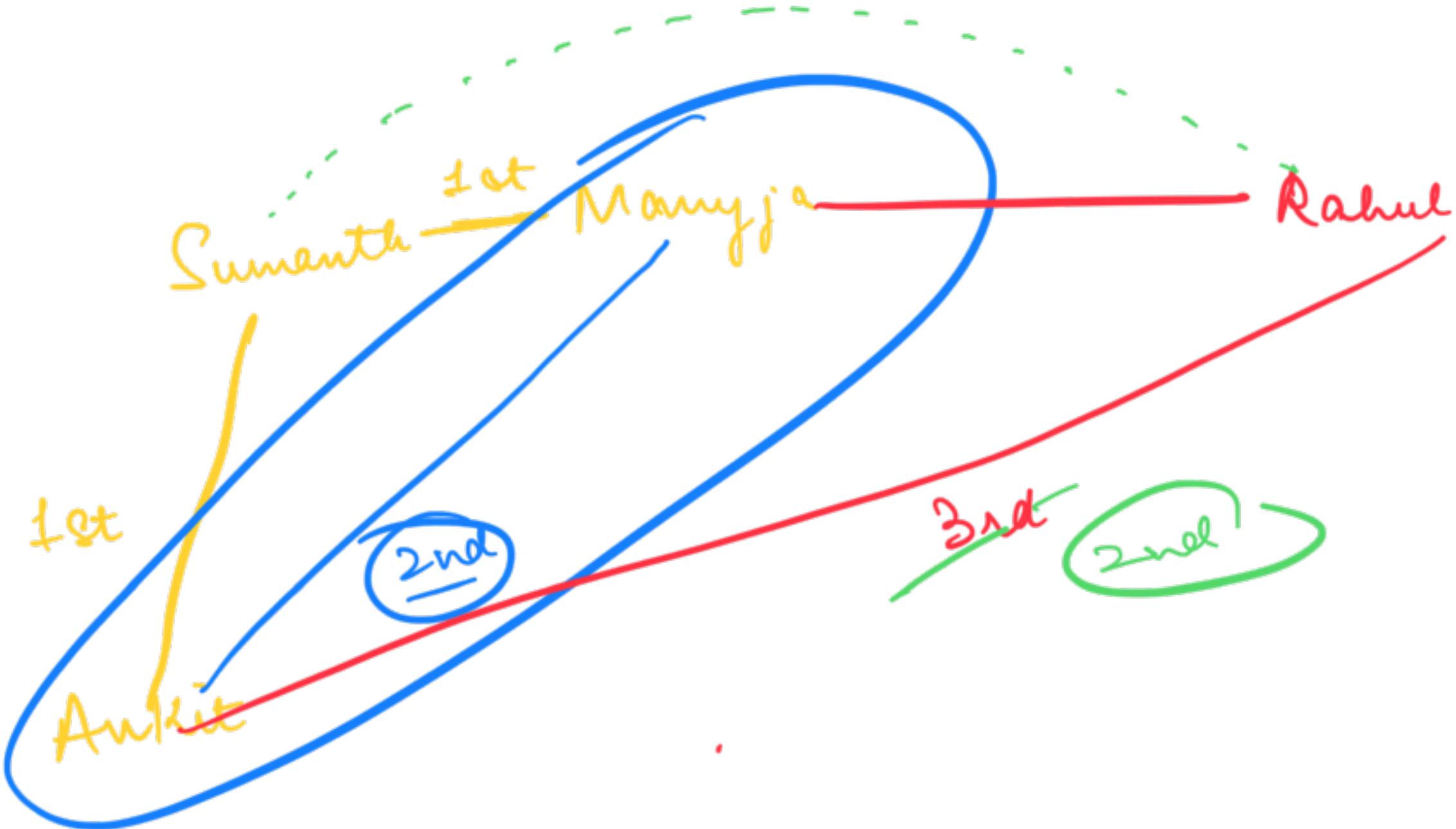


Linked In



Degree of Connectivity

Sumanth 1st Manyo



Ques. Given 2 linked In Profiles

and a way to find all the

— in a branching brolike.

Connections of a profile

Find the degree of conn between



Ques.

Given 2 LinkedIn Profiles A and B

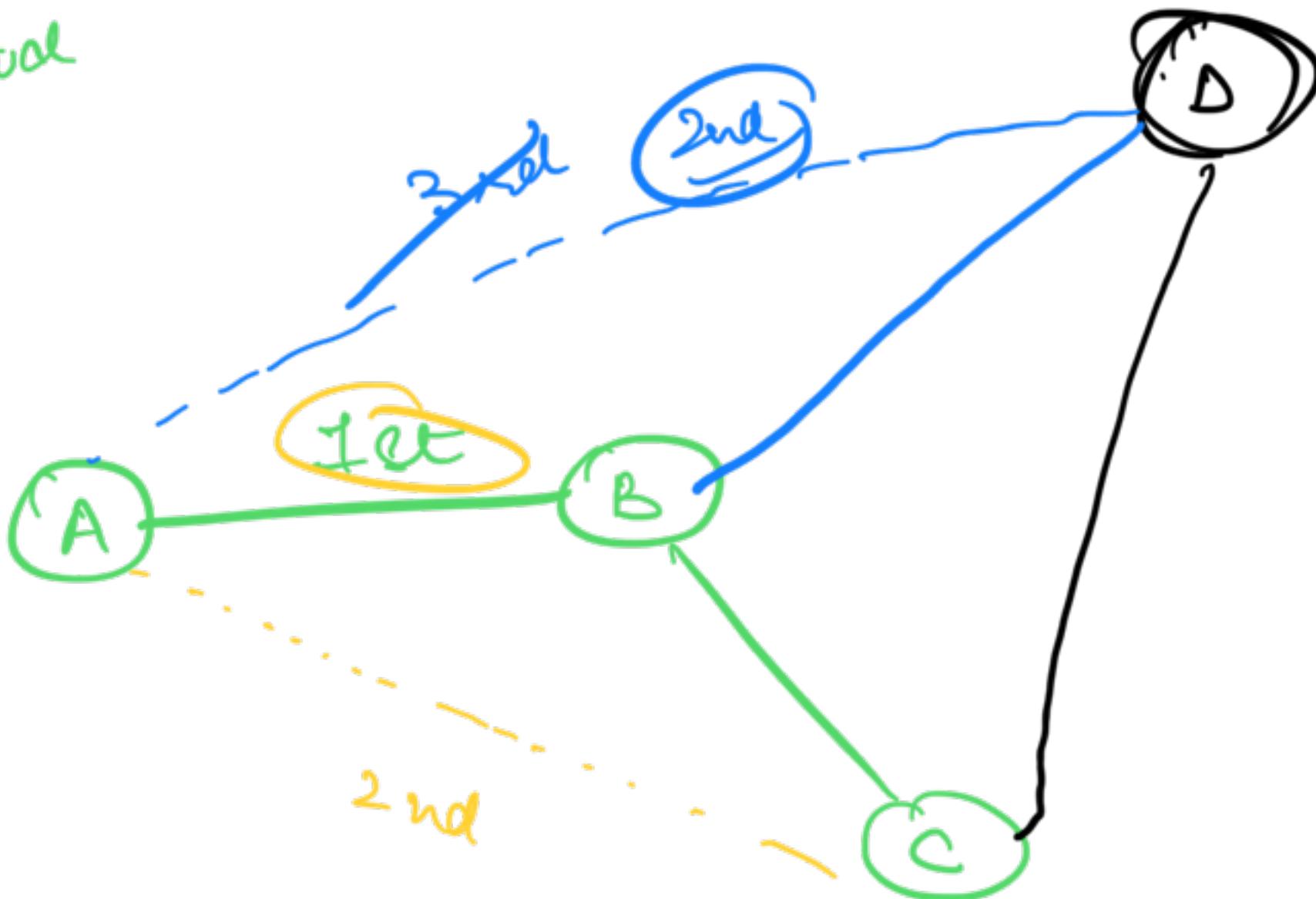
and a method that returns list

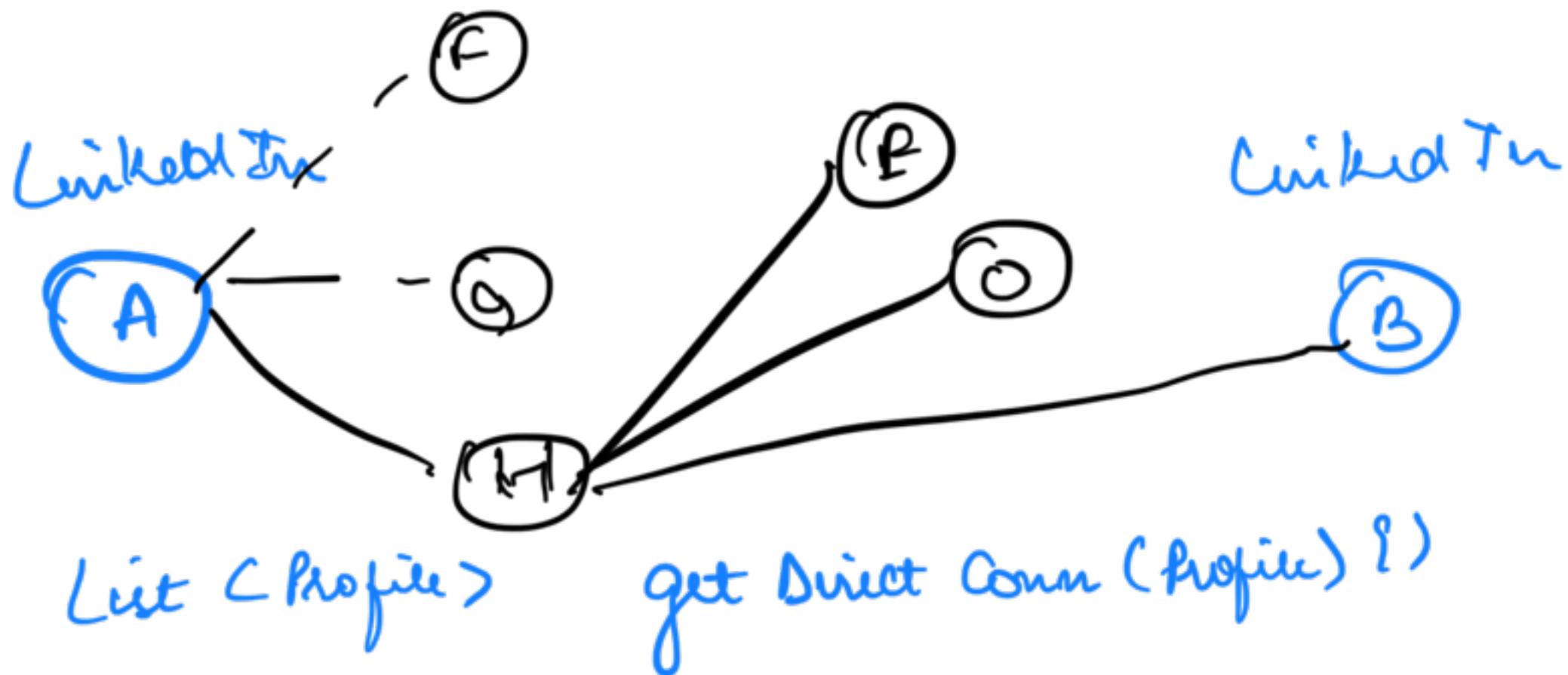
of direct connections for any LinkedIn
profile, TELL THE DEGREE OF
connection between A and B

CONNECTION

Degree of Connection

length of the path by which 2 nodes are connected





get Direct Connection (A) $\Rightarrow 2$
 ↓
 F, G, H

get Direct Conn (H)
 |||

β, ϵ, \circ

Question

Given 2 linkedIn profiles

A and B

Find the degree of connection b/w those

profiles.

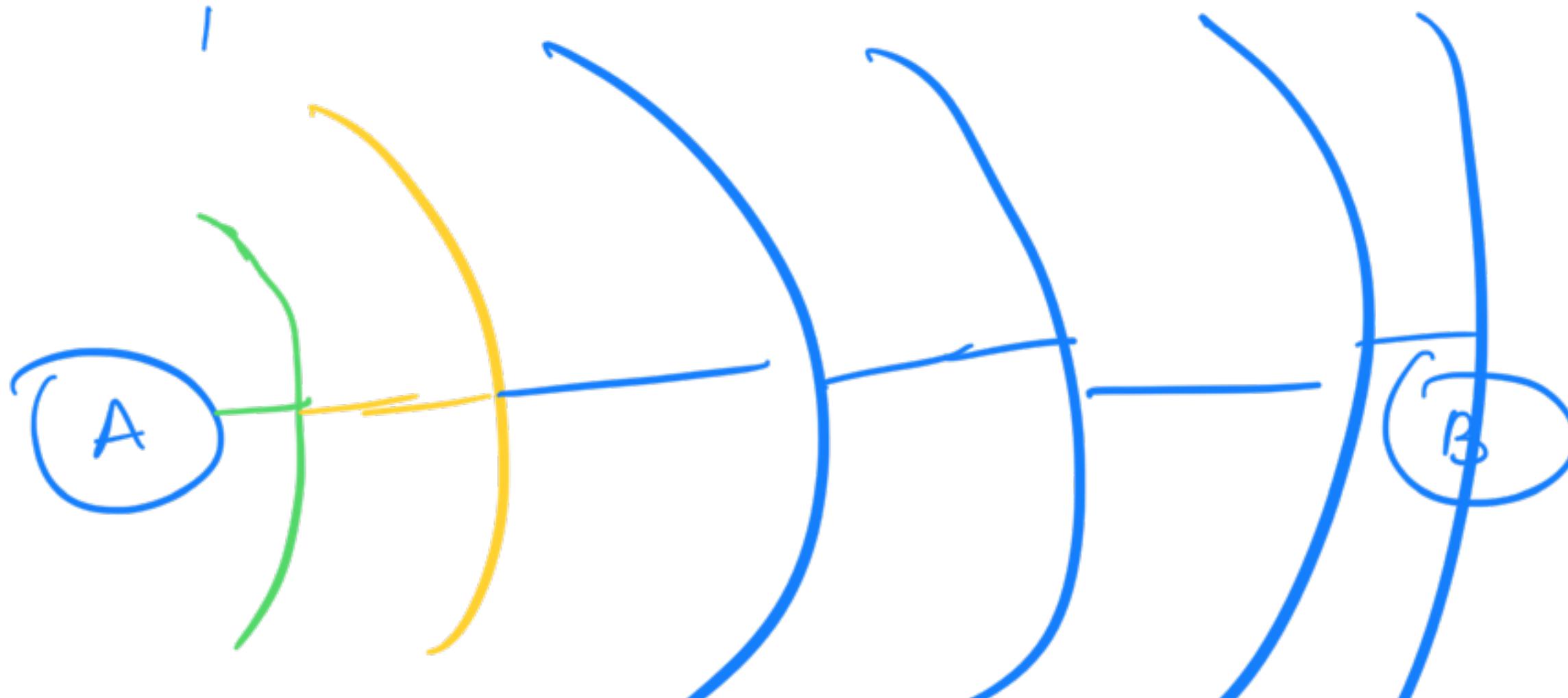
1st
2nd
3rd +

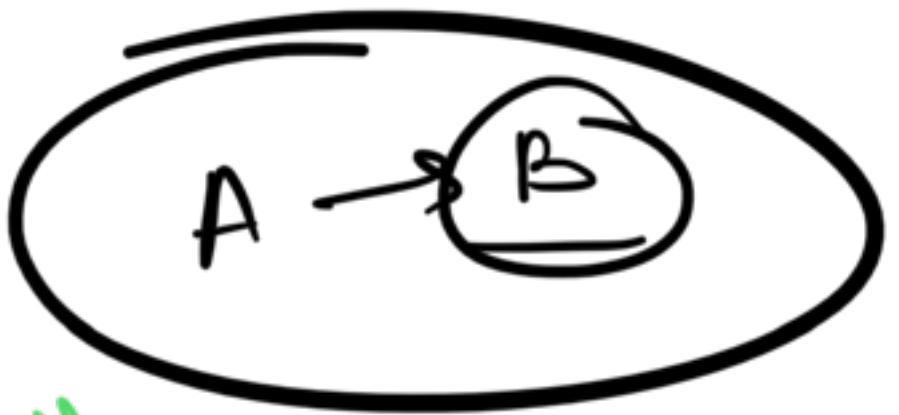
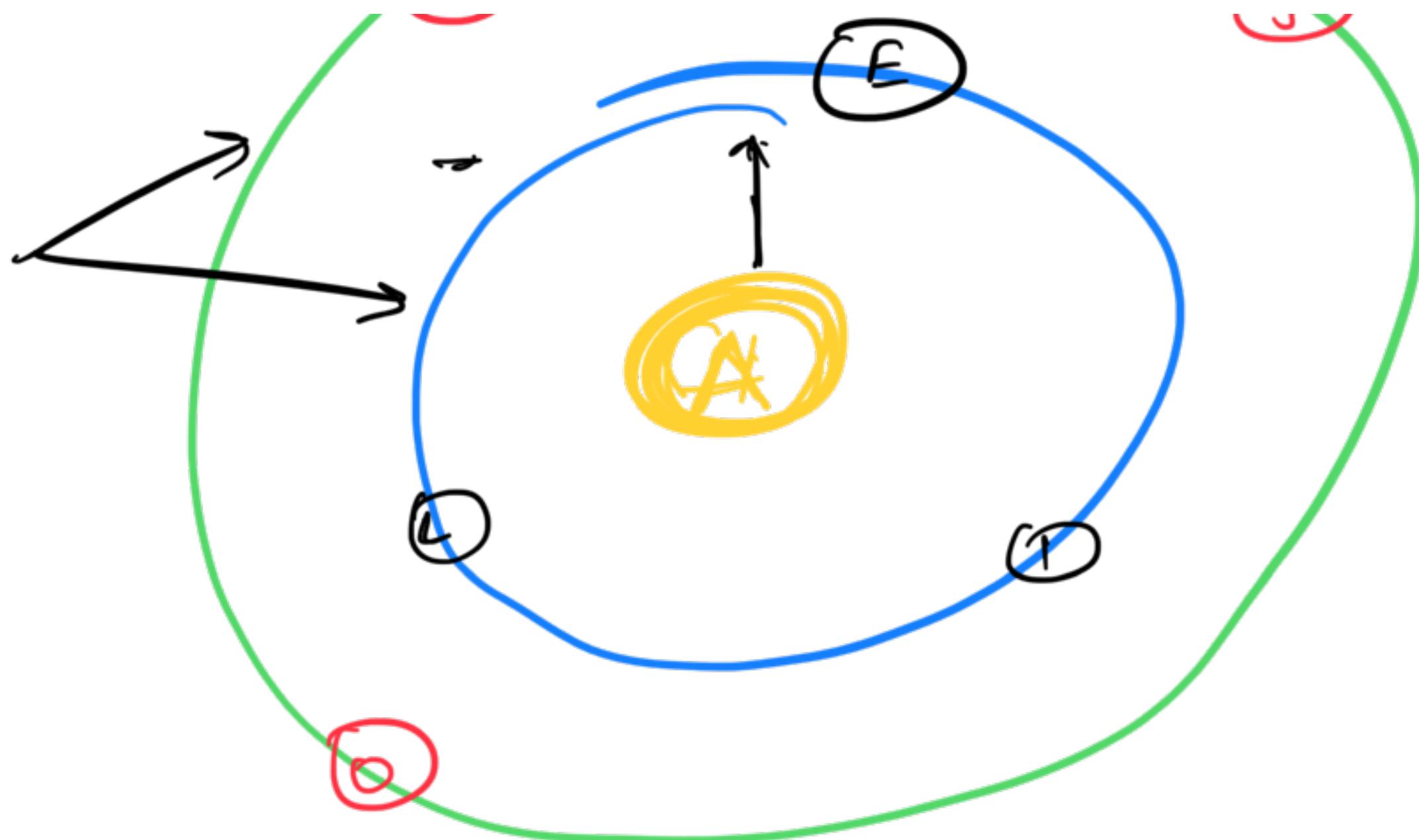
list <Profile> get Direct Connections (Profile)



1st Degree connection

SOLUTION





First Degree Conn = find Direct Conn (A)

for profiles in first Degree Connections :

 if profile == B :

 return 1

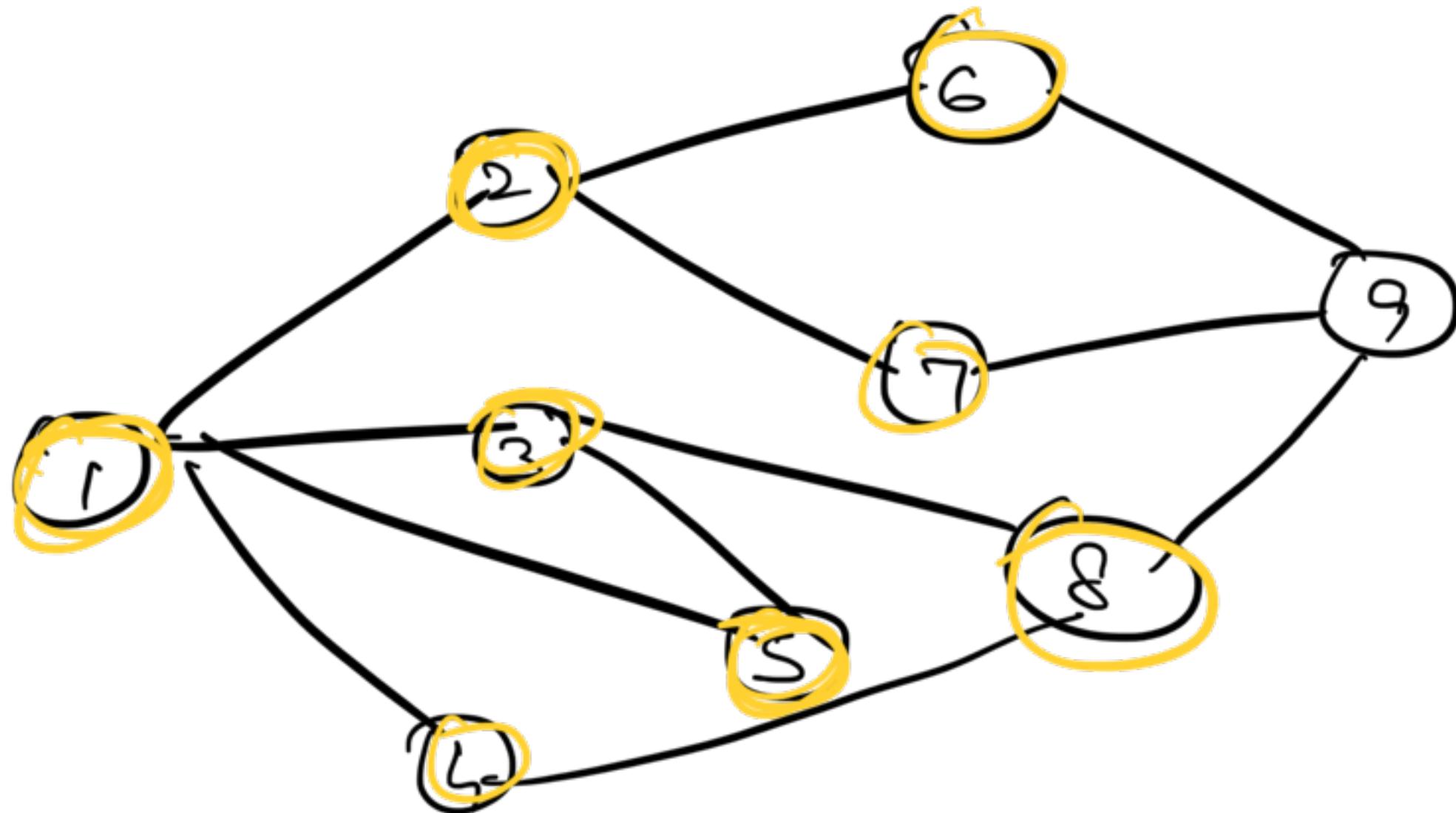
Second Degree Connections = []

for profile in First Degree Connection :

 Second Degree Connection.append(

 get Direct Connection(profile))

)



$$A = \boxed{1}$$

$$B = \boxed{9}$$

first Degree Connection = [② ③ ⑤ ⑦ ⑨]

Second Degree Connections = [⑥, ⑦, ①, ④, ⑤, ①, ③, ①, ⑧, ①]

③ +

get Degree DF Connection (A, B)

⑩⁶ → first Deg Conn = get Direct Conn (A)

→ second Deg Conn = [] (empty array)

for Conn in first Deg Conn:
 - loop 1

Second Deg Conn. upper

get Direct Conn (conn)

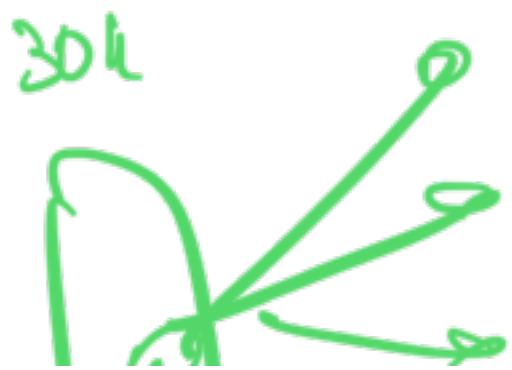
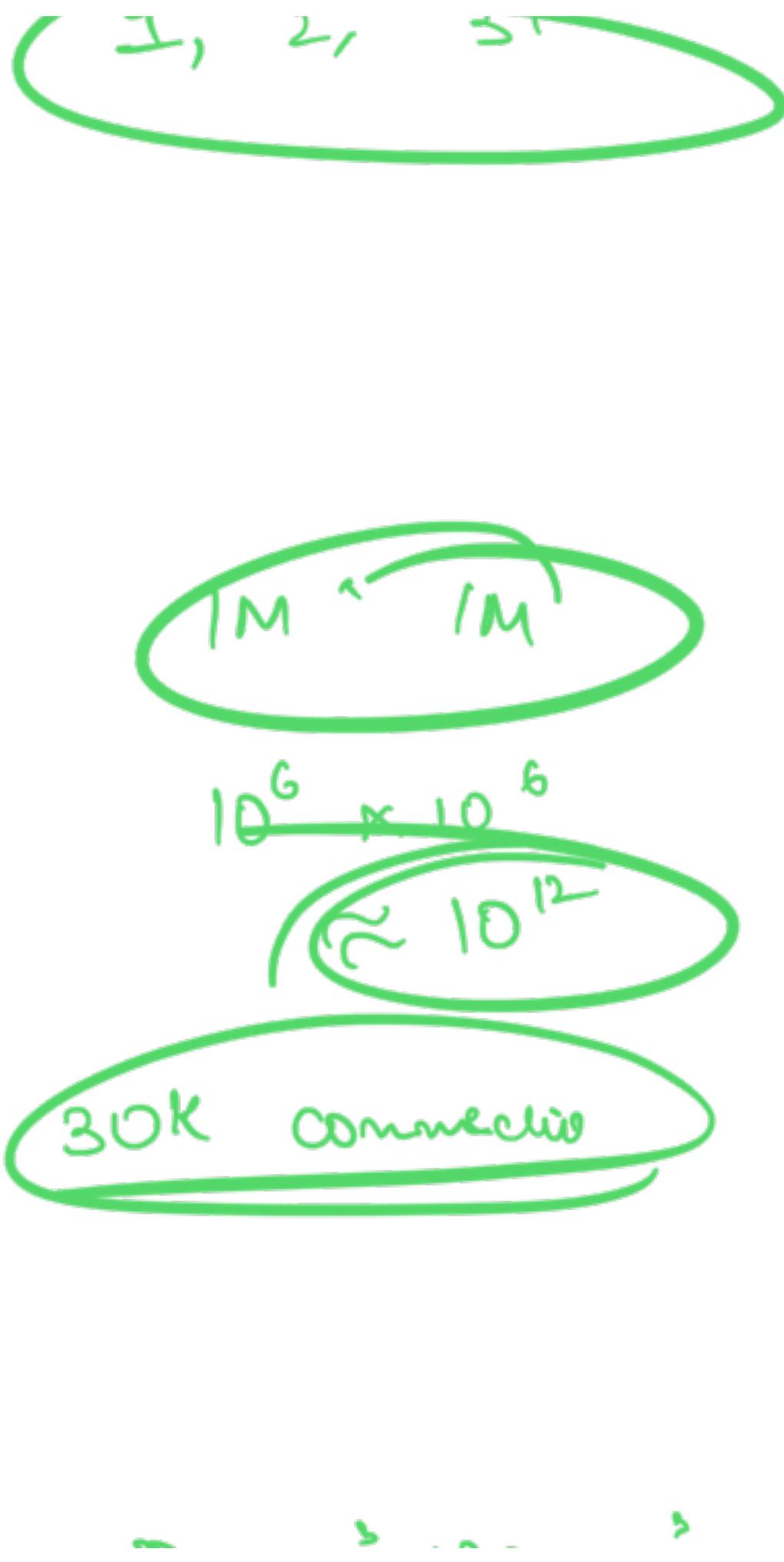
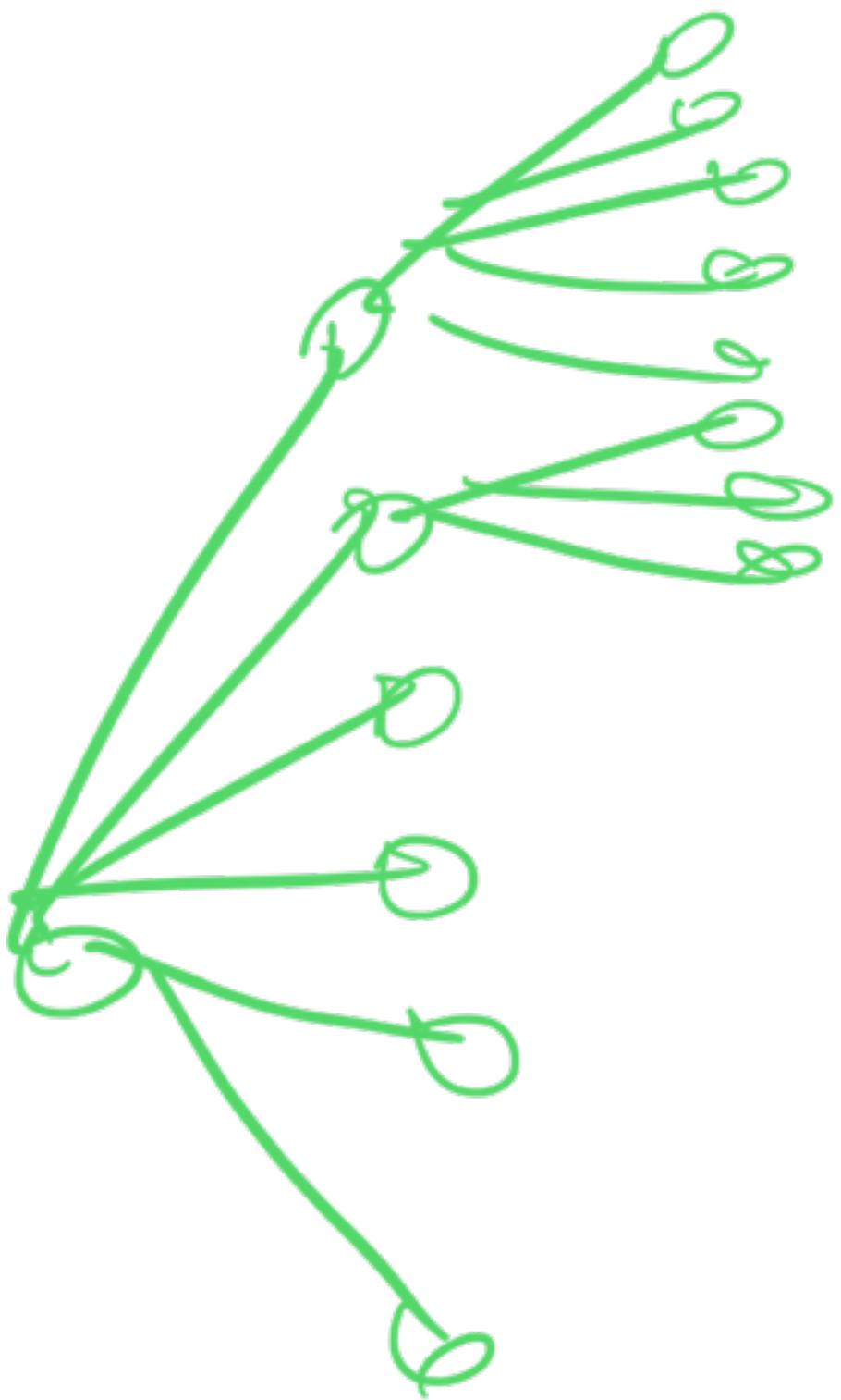
→ adding
all items

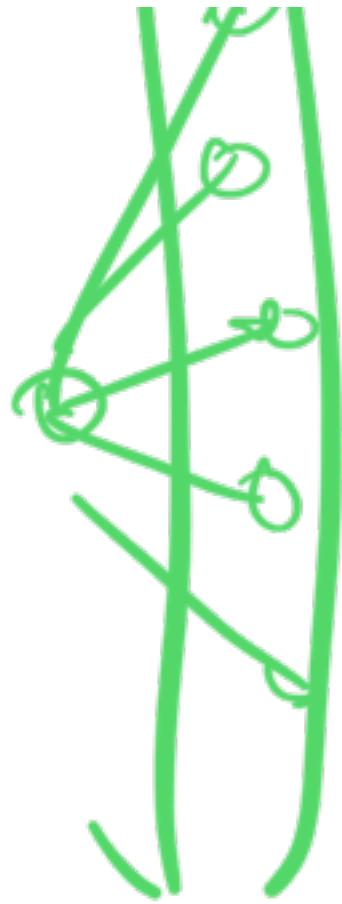
in to
second key

for conn in first Deg Conn:
if conn == B:
return 1

→ [for conn in second Deg Conn:
if conn == B:
return 2]

return 3⁺





$$30 \times 10 \times 30 \times 10$$
$$\rightarrow 900 \times 10^6$$
$$\rightarrow 9 \times 10^8$$

$$2.5 \text{ GHz} \rightarrow 2.5 \times 10^9 \text{ op/sec}$$
$$1 \text{ GHz} \rightarrow 1 \times 10^9 \text{ op/sec}$$

10^{12} op^z needed to search of
someone is a 2nd com.

$$\frac{10^{12}}{10^9} \text{ sec} \Rightarrow 10^3 \text{ sec} = 1000 \text{ sec} \\ \approx 16 \text{ min}$$

$$9 \times 10^4 \approx 10^9 \text{ sec}$$

1 person can have 30K

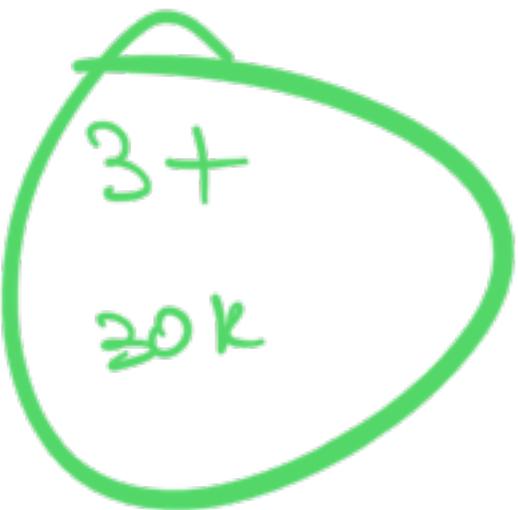
first deg Conn \rightarrow 30K

second deg Co \rightarrow 30K \times 30K

$$\rightarrow \cancel{30K} \cdot 30K \times 10^3 \times 30 \times 10^3$$

$$\rightarrow 900 \times 10^6$$

$$\approx 9 \times \underline{\underline{10^9}}$$



(A)

1

2+

1st

(B)

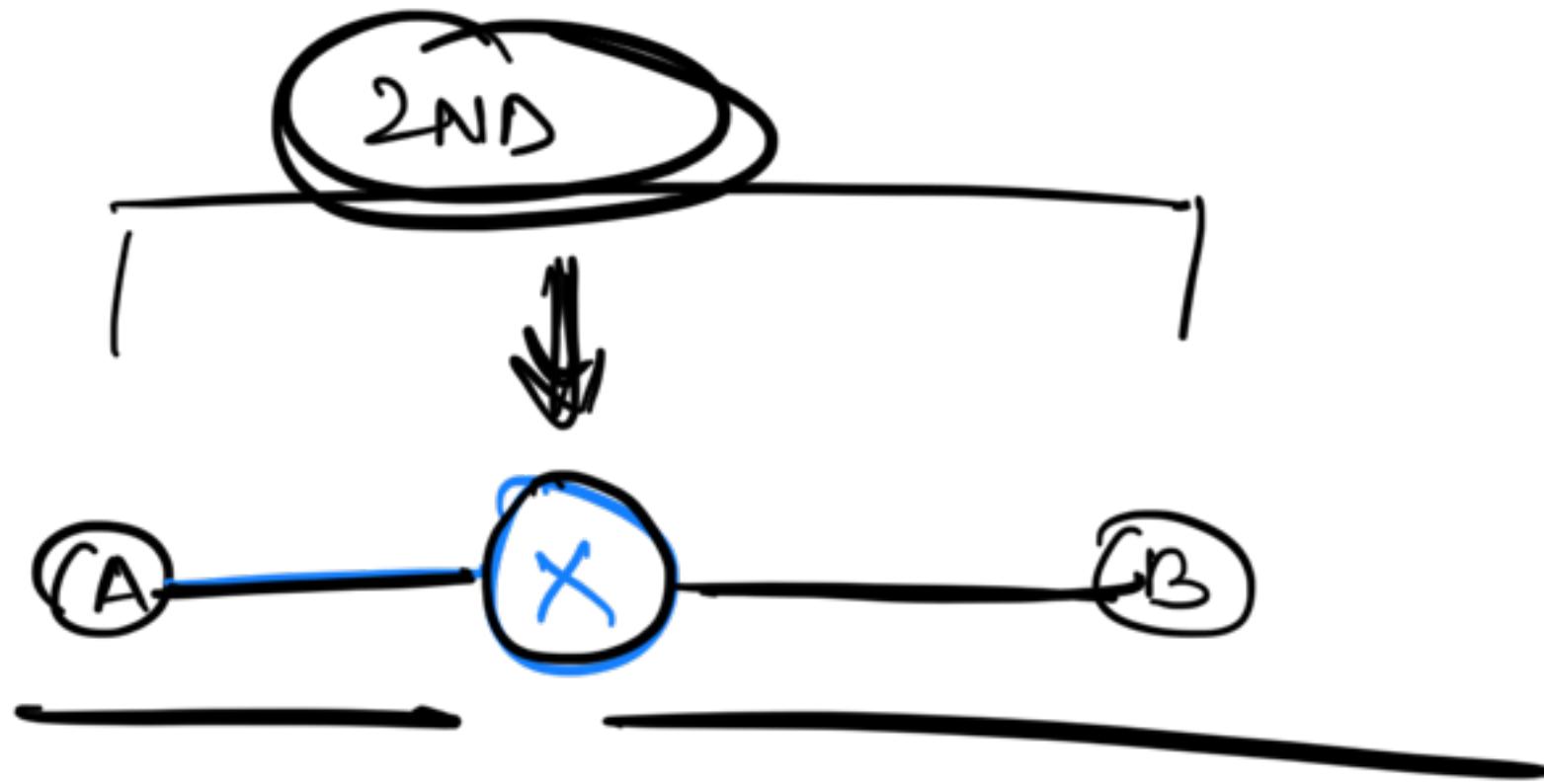
first check direct conn
and DONE

1st conn



① If A and B are directly conn
→ ANSWER IS 1

get Direct Conn (A)
and check if B exists there



→ first deg A = get Direct Conn (A)

→ first deg B = get Direct Conn (B)

Check if there is a common element in both

If YES :

return 2

Else

return 3+

E_Set =

$$E = [\underline{90} \quad 10, \quad 21, \quad 11, \quad 16, 1] \Rightarrow N$$

$$F = [\underline{21}, \quad 2, \quad 3, \quad 8, \quad 9] \Rightarrow M$$

\Rightarrow for $[i = 0 \text{ to } N-1]$ \Leftarrow 30 K
...

```

for [j = 0 to M-1]           ← 30k
    if E[i][j] == P[j]
        return true
    → 30k * 30k
return false
    → 9 * 10^8

```

$$E\text{-Set} \stackrel{?}{=} \{ \}$$

for i = 0 to N-1

E-Set. put(\bar{c} \in $[r]$)

~~for $i = 0$ to $M-1$:
 if E -set · cont~~

return true

return false



Check if a particular value



$$\frac{9 \times 10^8}{6 \times 10^4} \Rightarrow$$

1 GHz

$$\frac{6 \times 10^9}{10^9} \rightarrow 6 \times 10^{-5} \text{ sec}$$

$$\Rightarrow 0.00006 \text{ sec}$$

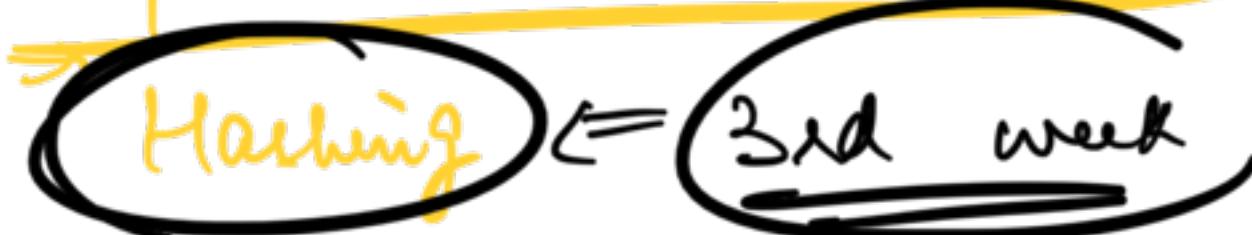
→ **0.06 ms**

Data Structure



→ Allows to store multiple values

→ Not stored in order

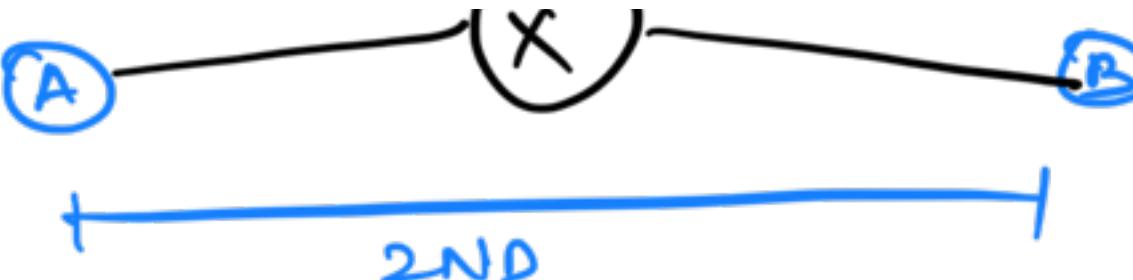


→ Allows to check if an element exists within it in O(1)

VS array



~~NoP^m~~



~~Op^m~~

get Connection Degree (A, B)

30K

first Deg ConnA

get Direct Conn (A)

Check
Common

for profile in first Deg ConnA:
if profile == B:
return 1

30K

first Deg ConnB =

get Direct Conn (B)

connB - Set = {}

for profile in first Deg ConnB:

connB - Set. add (profile)

for profile in first Deg ConnA:

60K Op^m

< 1 ms

if command-line contains (profile):
 return 2
else:
 return 3



C++ → unordered_set

Java → HashSet

Python → set

JS → Set

C# → Dictionary