

# DARE TO DREAM

- ① WHAT WE NEED FROM YOU - vs what are responsible for
  - ② Overview of curriculum
  - ③ Example / problem → solve it together
- 

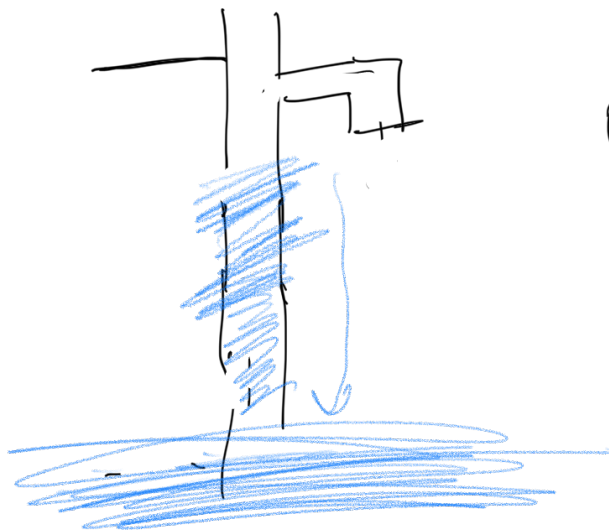
① Hard work.

↳ YOU

② Direction, structure, Smart work  
↳ OUR

③ Consistency → 1% better everyday.

$$\boxed{(1.01)^{365} = 37.8}$$
$$\downarrow$$
$$\frac{1}{(0.99)^{365}} = 1$$
$$(0.99)^{365} = 0.01$$



① Gamification

② Social

---

Interviews

CURRICULUM

Become better  
engineer

...linked | Data structures and

① Coding accuracy

SDEL  $\hookrightarrow$  EM  $\rightarrow$  H $^{\infty}$

algorithms

↳ I real ~~is~~ life example

- ② Efficiency as productivity
- ③ 5% cases of the problem

# CS Fundamentals

- ↳ OS, Computer Networks, Database
- ↳ some examples

CORE

## DESIGN

Important

## Low level design

Very importa

## High level design

important  
as you &  
senior

3+ yrs  $\hookrightarrow$  super important

50 chots per day

20 Billion chats per day

1 day

$20 \times 10^9$  chats.

200 bytes

1000 bytes  $\approx$  1 kB

1000 KB

1 MB

1000 MB

21

100 GB

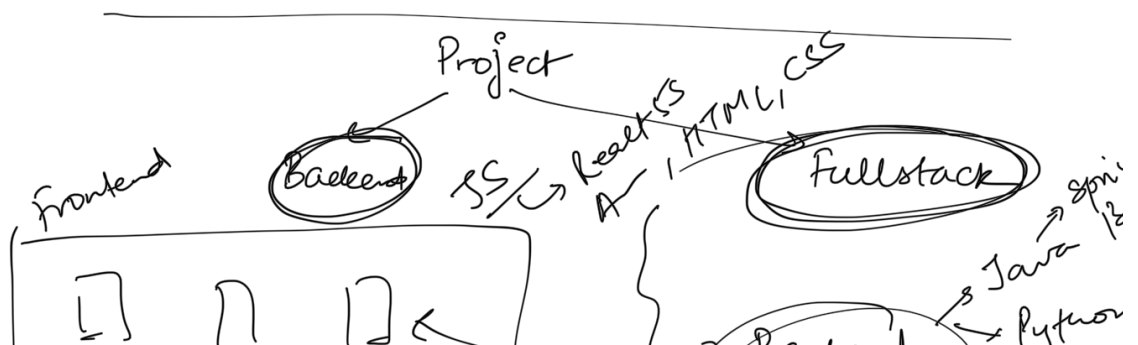
LTB

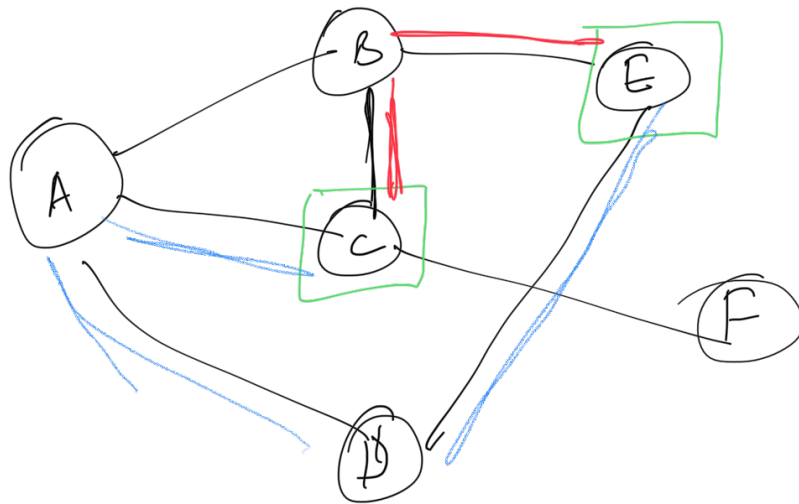
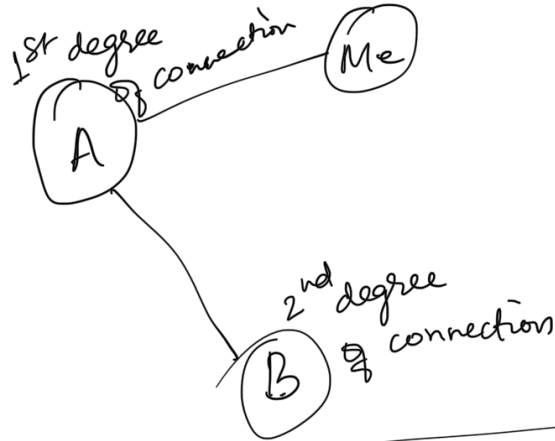
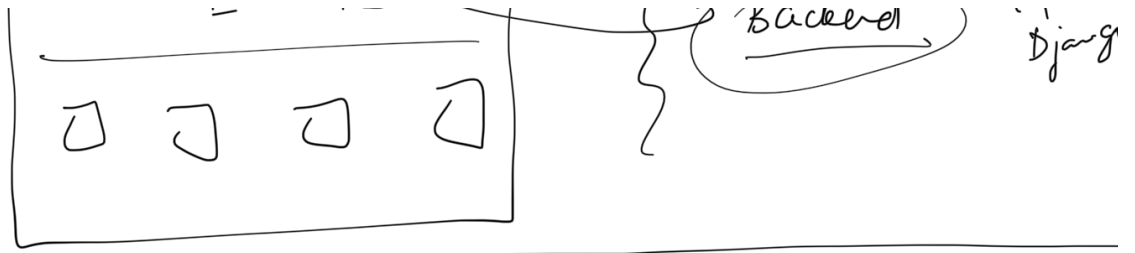
$$\Rightarrow \underline{20 * 10^9 * 200 \text{ byt}}$$

→  $4000 * 10^9$  bytes

1 GB

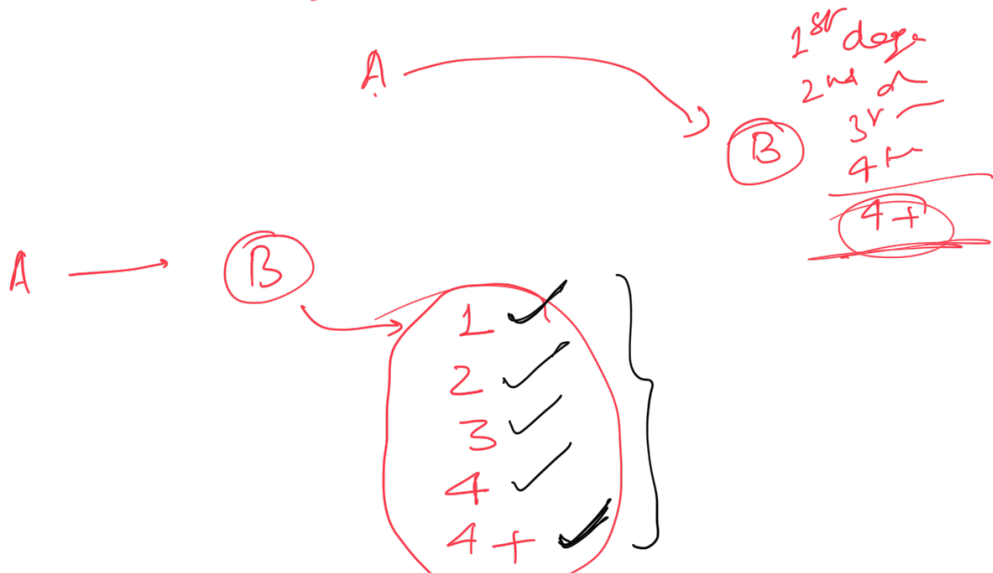
→ 4000 GB ⇒ 4 TB





2  
3

LinkedIn



degree-of-conn (A, B) :

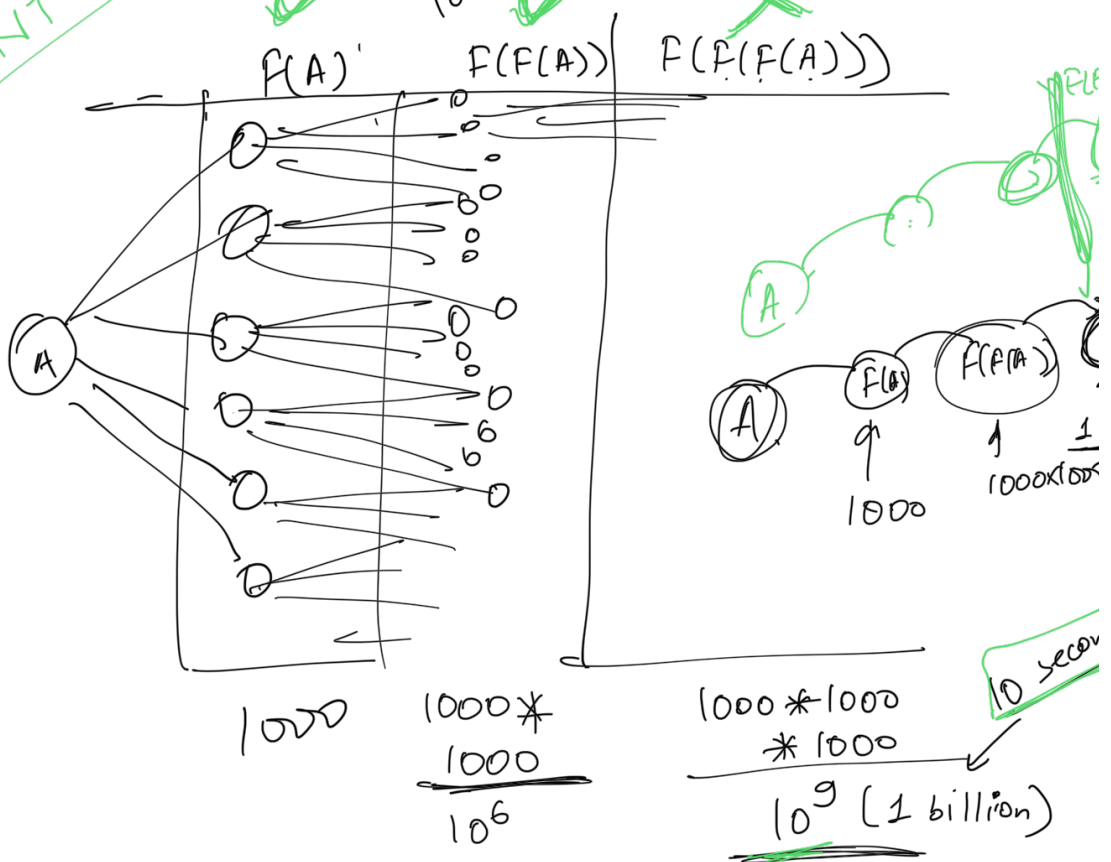
$O(\sqrt{N})$   
 $2 \times 10^8$

friend(A)  
friends(friends(A))

HINT

$\frac{10^8}{10^9} \rightarrow 1 \text{ sec}$   
 $\frac{10^8}{10^9} \rightarrow 10 \text{ second}$

$10^8 \rightarrow 1$



$i \rightarrow 1$  to 1 billion :  
 # do something

0 or 1  
 3  
 011

2 GHz CPU  
 1.6 GHz CPU

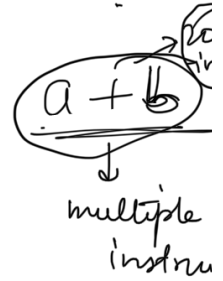
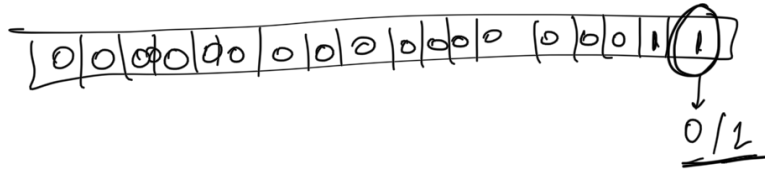
1 ... 1

int float bool . . .  
 ↓  
 4 bytes → 32 bits

8 bit = 1 byte

2 GHz CPU  
 ↓  
 $2 \times 10^9$  instructions per second

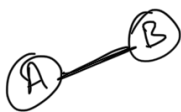
int a = 3;



1 second →  $2 \times 10^9$  instructions  
 1 ~~instruction~~ <sup>operation</sup> → 20 instructions

→  $\frac{2 \times 10^9}{20}$  operations per sec

→  $10^8$  operations per second



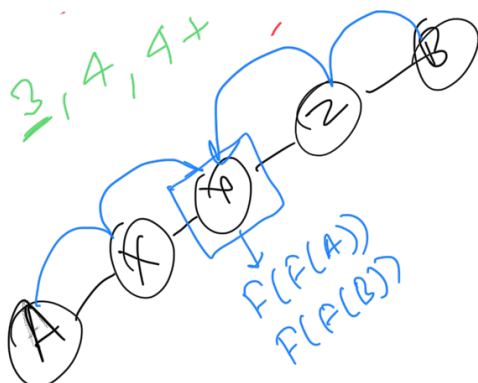
deg-of-sep(A, B) : if (A == B) return  
 ① if B in friends(A) : return 1

② if intersect(F(A), F(B)) return 2

if intersect(F(F(A)), F(B)) return 3

if intersect(F(F(A)), F(F(B))) return 4

return 4



In & .)

7



friends (10<sup>6</sup>)  
[...]

(10<sup>9</sup> ops) → 1 sec

intersect (L1, L2) :

for x in L2 :

← x exists in L1

~~sorting & search~~

fast

OR  
Hashmap/HashSet  
avg O(1)

$N * M$

→ O(N + M)

10<sup>6</sup> + 10<sup>6</sup> → 2 \* 10<sup>6</sup>

Large problem

