PATH TO A MAANG ENGINEER
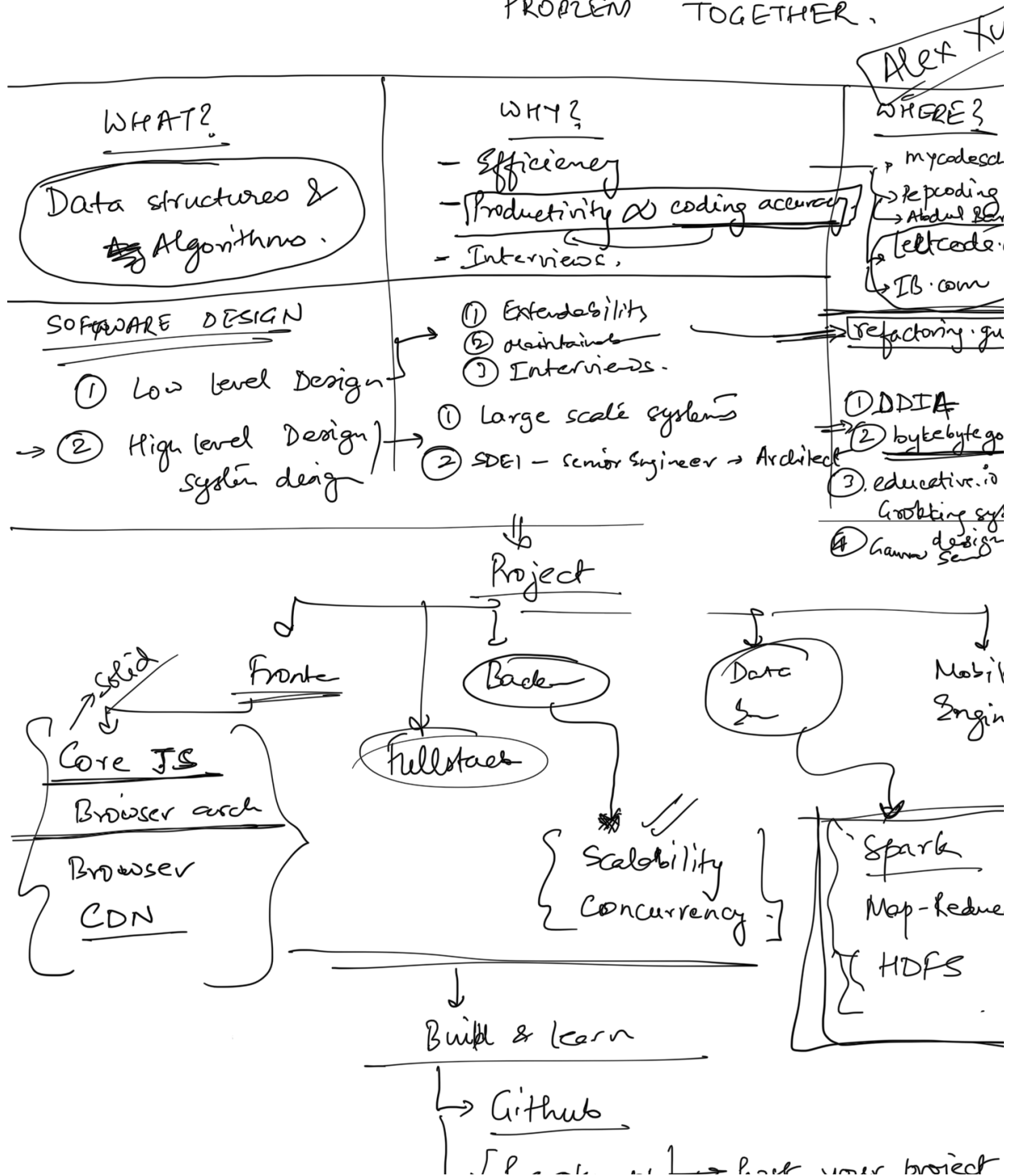
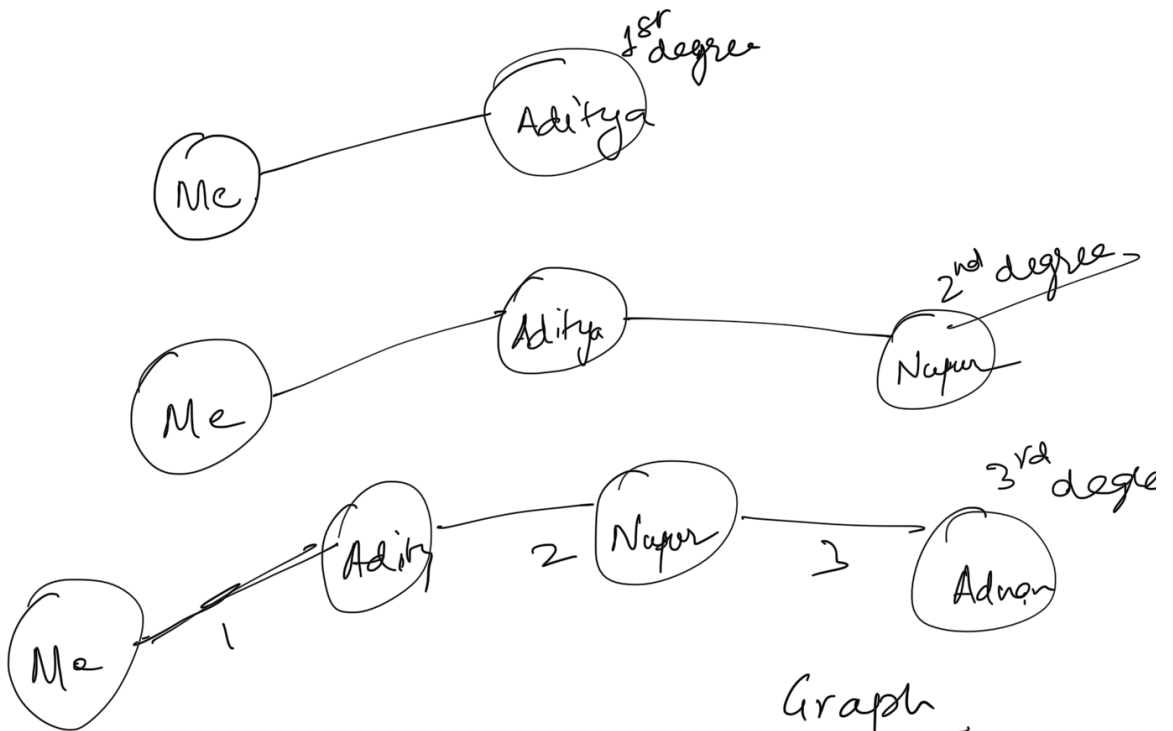## AGENDA

1. WHAT SHOULD YOU LEARN, WHY AN
                              HOW?

2. LETS SOLVE A GOOGLE INTERVIE
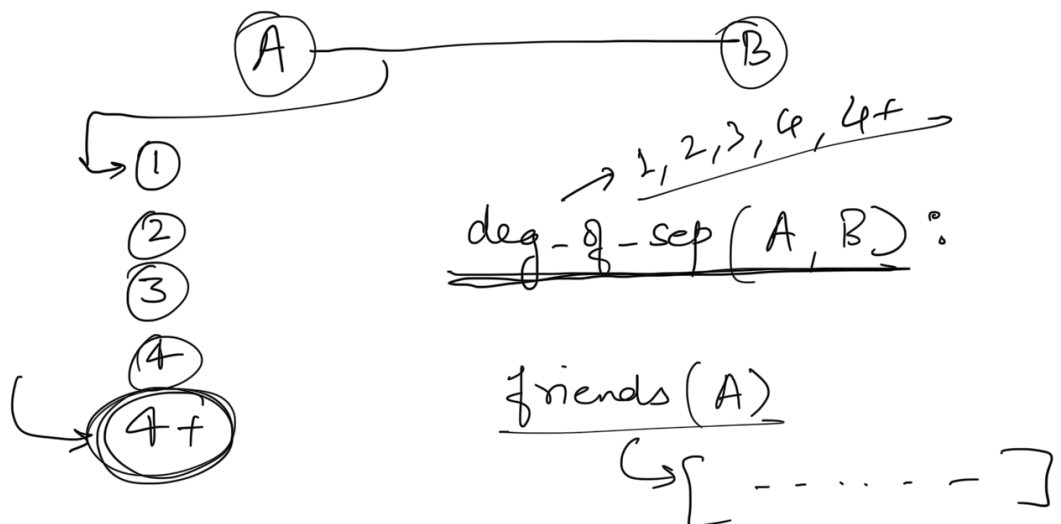   PROBLEM TOGETHER.

Alex Xu

WHAT?	WHY?	WHERE?
(Data structures & ~~Alg~~ Algorithms.)	— Efficiency — Productivity ∝ coding accuracy — Interviews.	• mycodesc → repcoding → Abdul Ba leetcode. → IB·com

SOFTWARE DESIGN
 	
1. Low level Design
→ 2. High level Design)
      System design
	1. Extendability
	2. maintains
	3. Interviews.
	
	1. Large scale systems
	2. SDEI — Senior Engineer → Architect
		→ refactoring.gu
		
		1. DDIA
		2. bytebytego
		3. educative.io
		   Grokking sys
		4. Gauriav design se

⇓

Project

```
        Solid        Fronte      Back         Data        Mobil
                                                          Engin
```

(Back)

(Fullstack)

- Core JS
- Browser arch
- Browser
- CDN

{ Scalability
  Concurrency }

{ Spark
  Map-Reduce
  HDFS }

↓

Build & learn

↳ Github

... ↳ host your project

→ heroenapp { → host your proj...
　　{ AWS free tier }　　　　for free

---

① Kritika' advice
② Solve a DSA Google problem
　　　together.

Degree of separation

1st degree

Me —— Aditya

Me —— Aditya —— Nupur　　2nd degree →

Me —— Aditya —— 2 —— Nupur —— 2 —— Adnan　　3rd degree

1　　

Graph

A —— B

→ 1, 2, 3, 4, 4+ →

deg-of-sep (A, B) :

friends (A)

↪ [- - - · · · - -]

A → ①
　　②
　　③
　　④
　　④+

friends ∘ (friends(A)) →

F(A) F(F(A)) F(F(F(A)))

F(F(F(A)))

1000

1000 * 1000
= 1 million

1000 * 1000
* 1000
= 1 Billion

F(F(A))

4 km
2 km Delhi
2 km X F(F(A)) B
2 km
A N 2 steps: F(F(B)) 2 steps

A

A

1, 2, 3, 4, 4+

deg-of-sep(A, B):

1
2
3
4
4+

= if B in friends(A):
      return 1

else if intersect(F(A), F(B)):
      return 2

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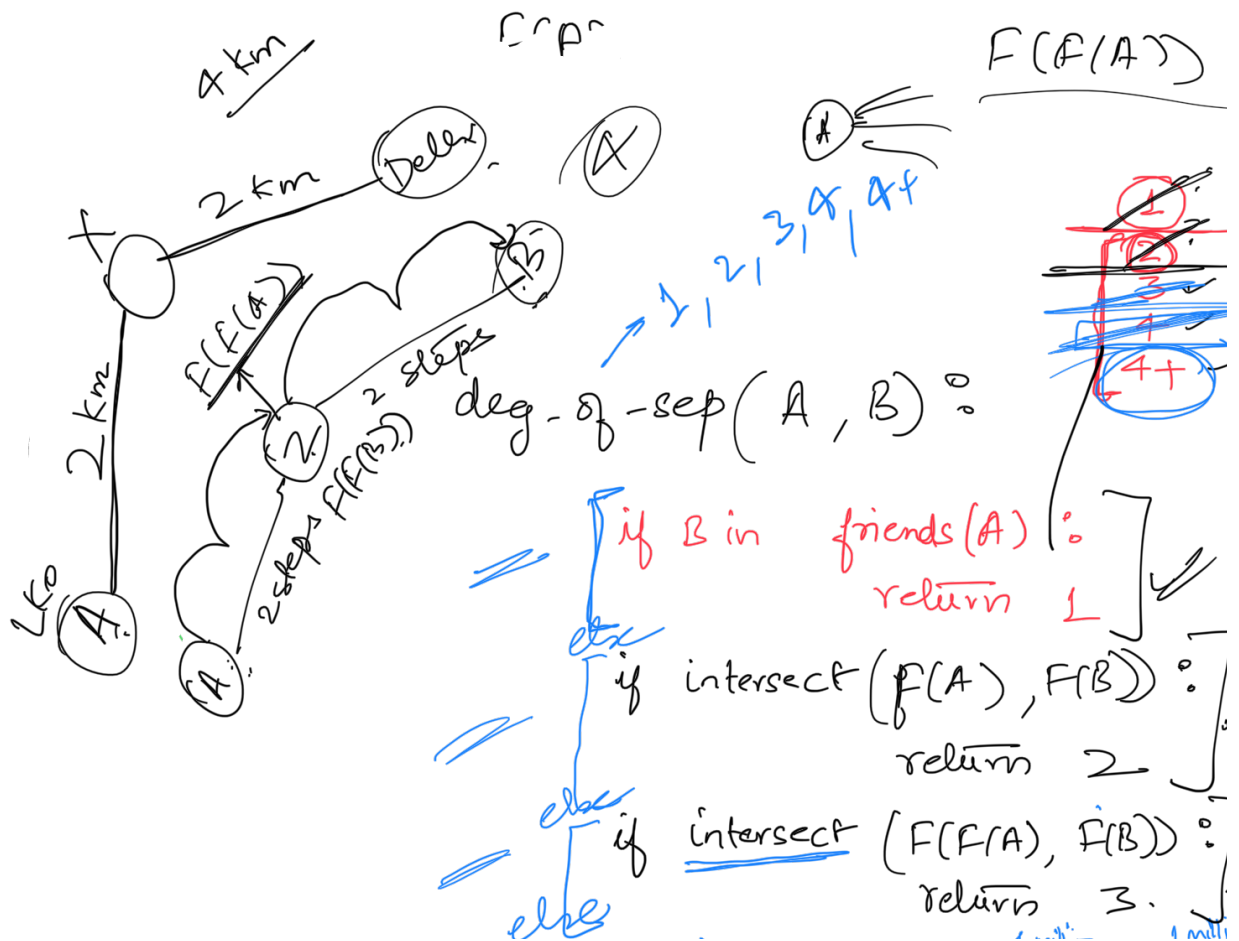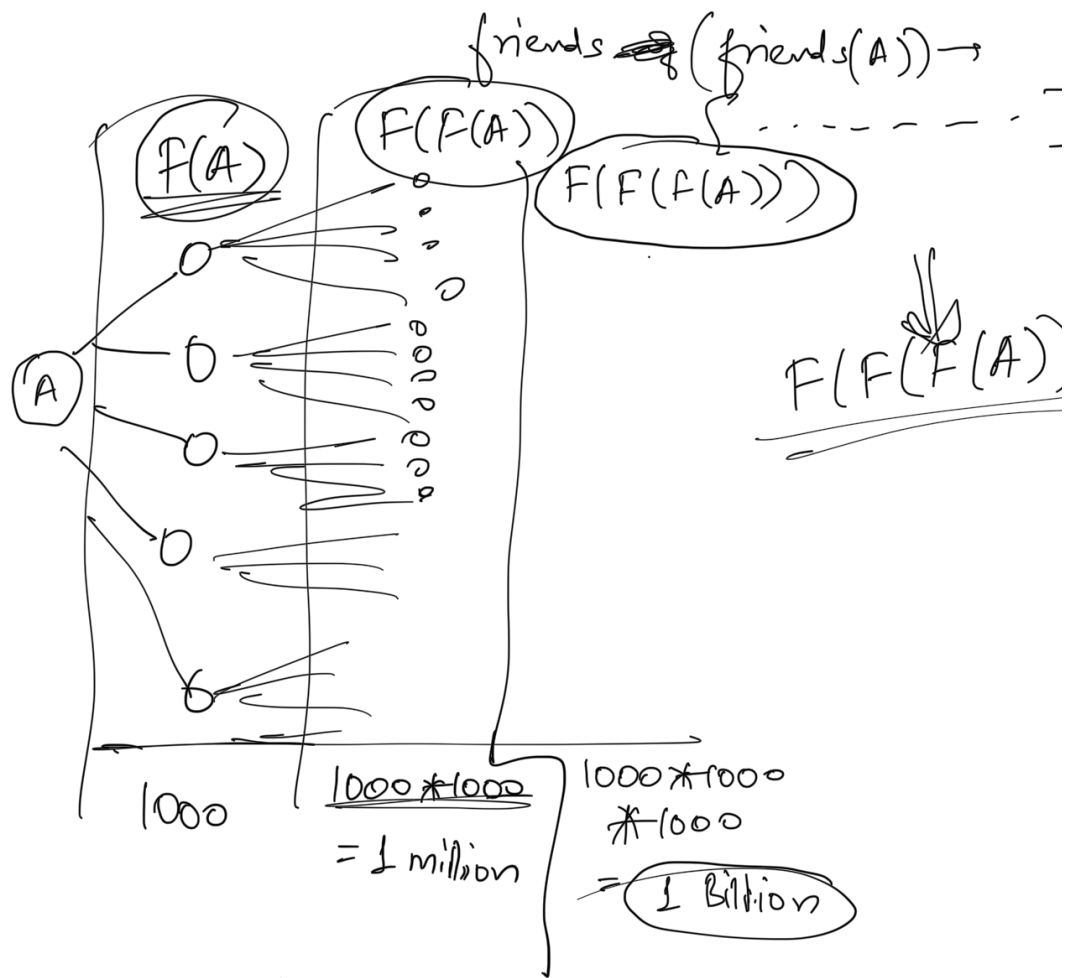
else

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

T > 2

Rust   intersect (list1, list 2) :
                                        1 million   1 million
                                                        sort (list 2)
1000 → N   M
                             for X in list 1 :

log N   log N ≈ search (X, list 2)

500

           M log N + N log N