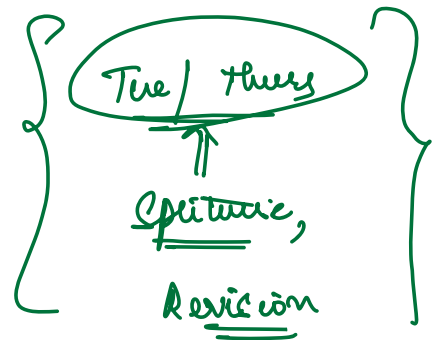


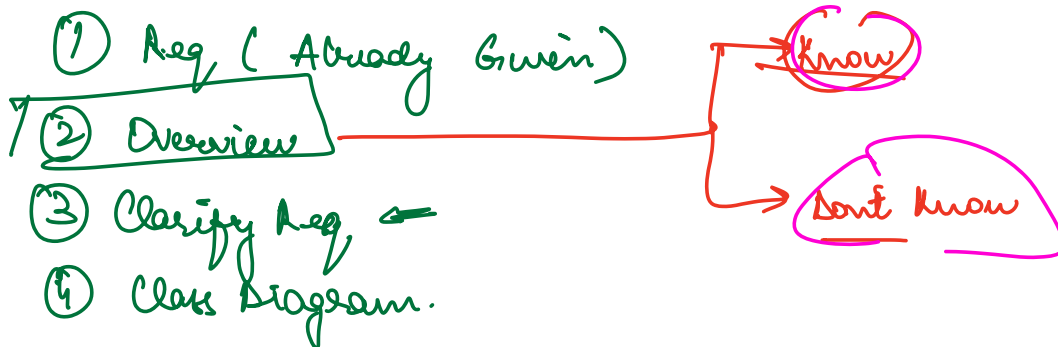
→ How to settle expenses

- ① Overview
- ② Gather & Clarify Req
- ③ How to settle expenses.



-- a Machine Coding Round

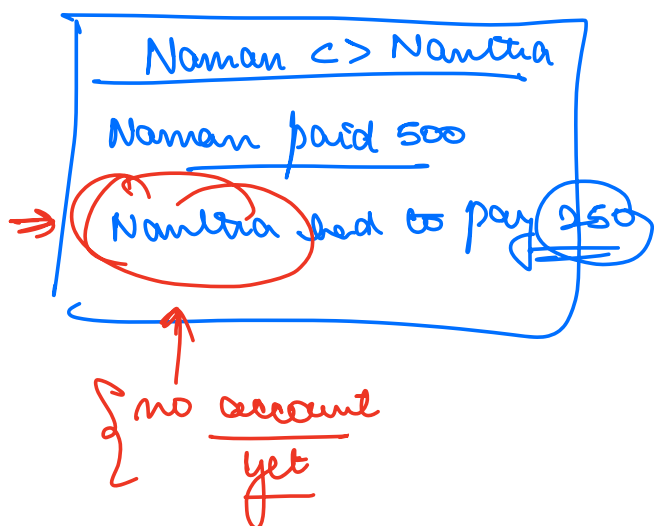
MC Round (Design an Expense Tracking Appⁿ)



- ⇒ {
- ① what you know about System Ask Interview
to give you idea
 - ② is it entity or C/V
 - ③ how will we persist data
 - ④ how will it be taken

- ① Read Req
- ② Overview
- ③ Clarify Req

→ Can include a person who doesn't even have an account it



Yes. Allow adding a user that doesn't have an account yet. If their phone # is not present in users table, create a new user with status INACTIVE

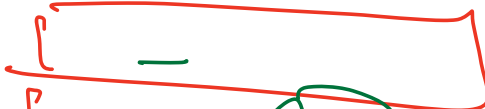
Expense

Amount (C) 2000

Dee



members:



// went to Amc

paid :

Naman : 1000
Ashish : 1000

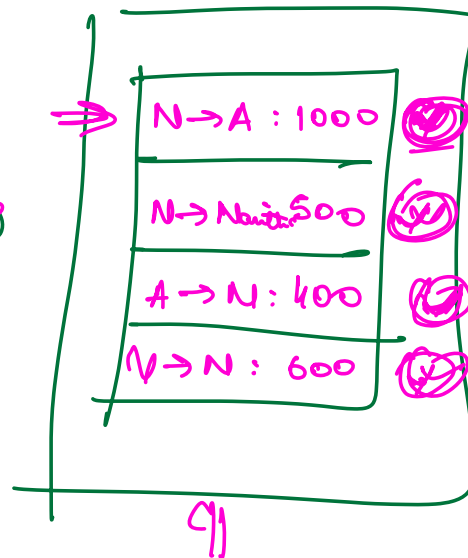
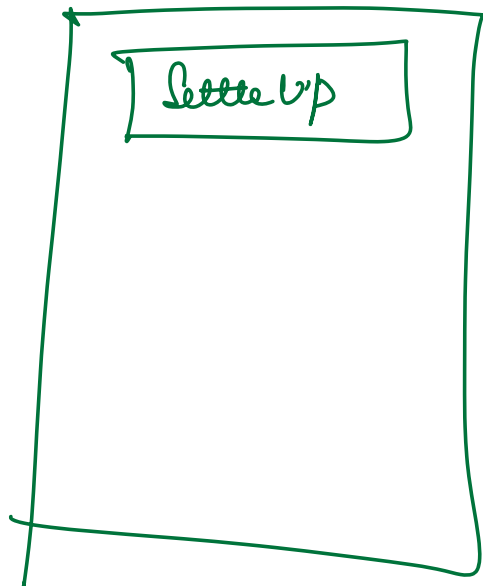
2000

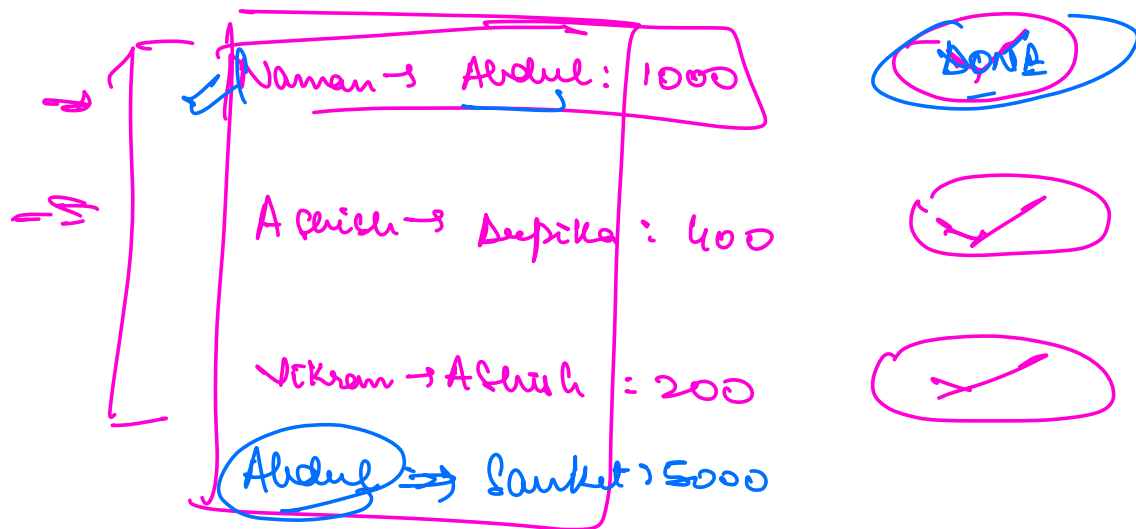
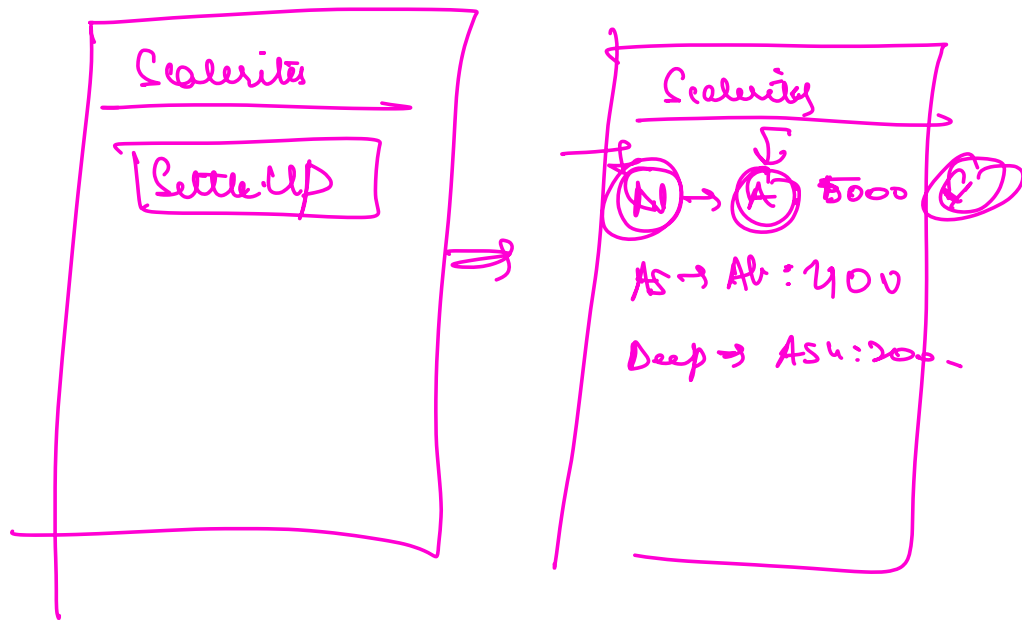
had To Pay :

=

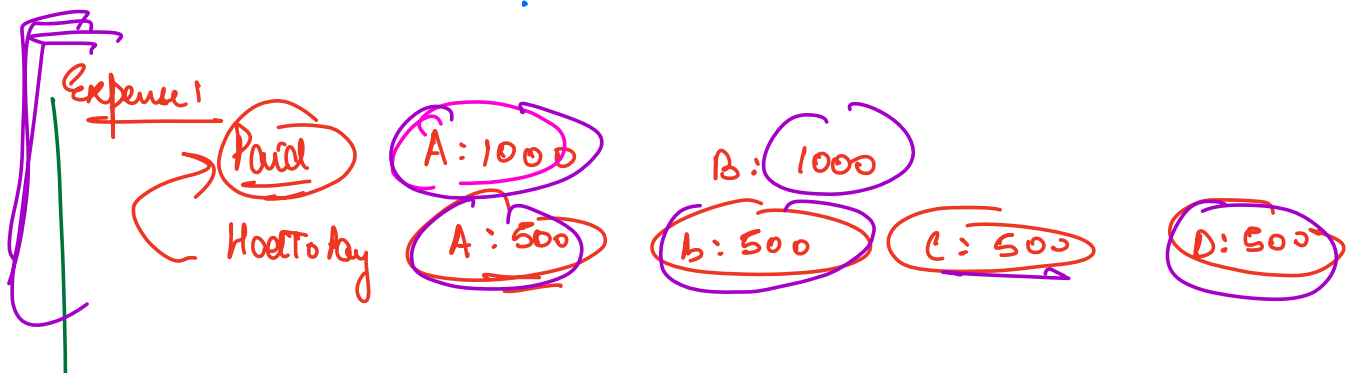
Naman : 400
Ashish : 400
Ujjwal : 800
Kiran : 200
Vikram : 200

2000





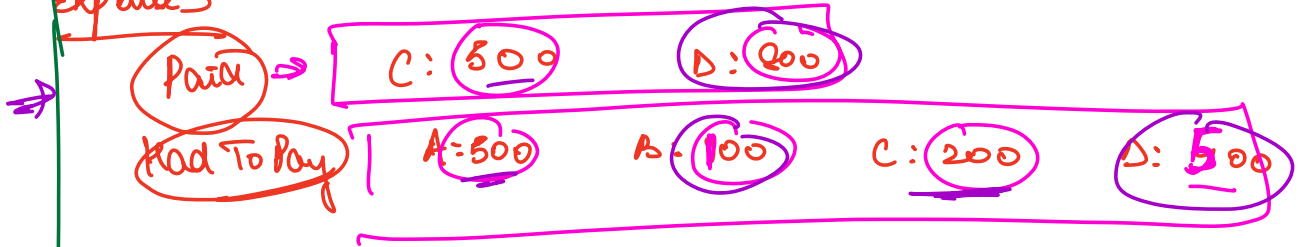
How to handle Settle up



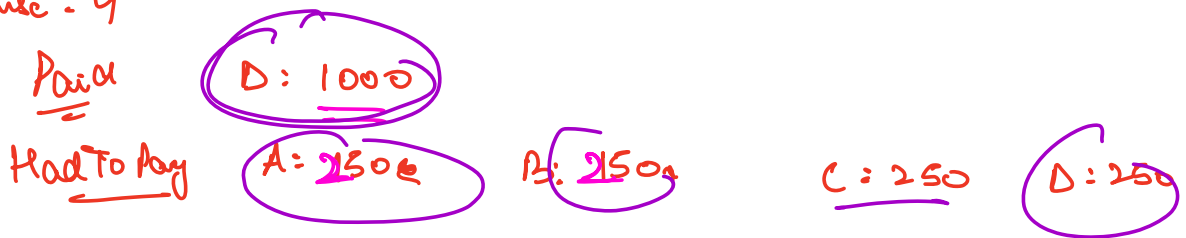
Expense 2



Expense 3



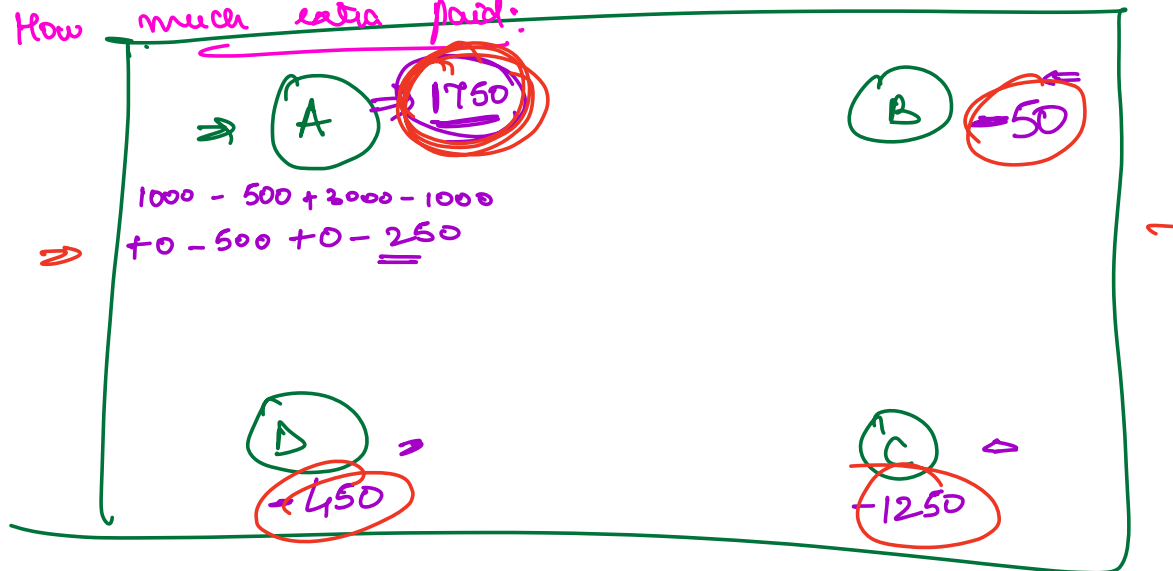
Expense 4



Settle up

list of transⁿ that
shall settle up everyone

How much extra paid:



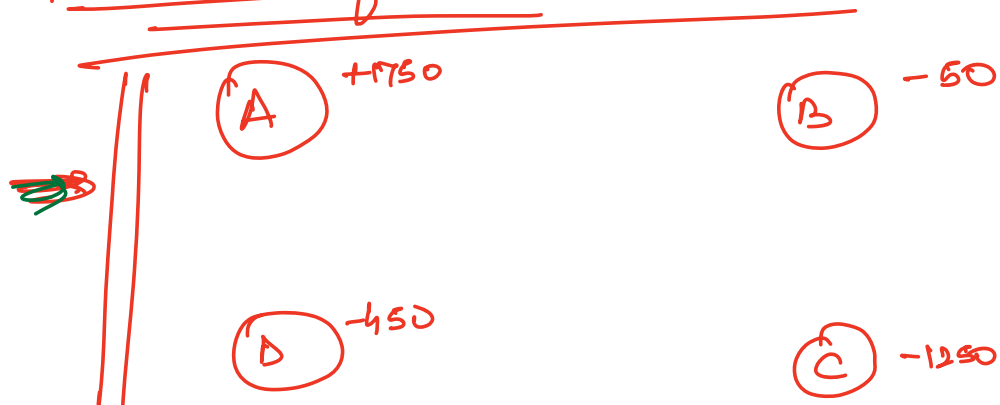
$$\text{Overall } \Sigma = 0$$

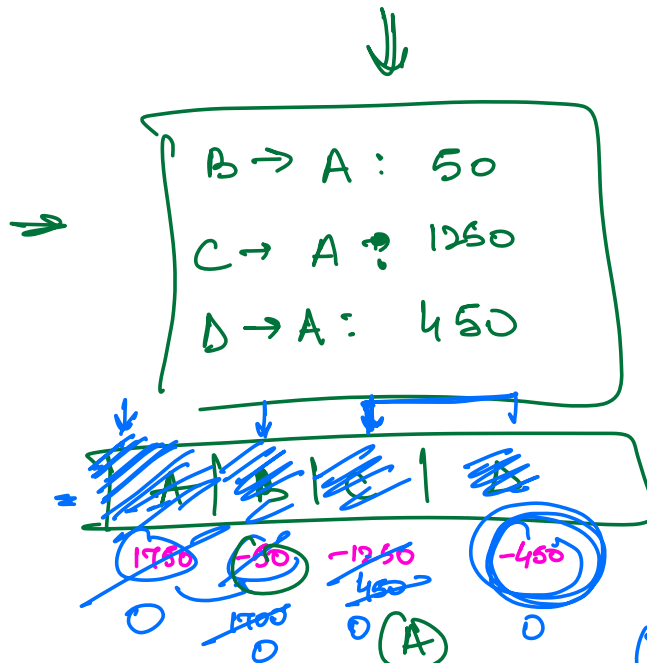
for each user:

for each exp:

$$\Sigma \text{ paid} - \Sigma \text{ had To Pay}$$

- ① We have list of expenses
- ② We comp how much extra or less has everyone paid
- ③ Compute list of transⁿ





A	B	C	D
-250	-450	700	-30

Algo 2 → Only someone who has paid less should be asked to pay money
 → people who have paid more should only get money

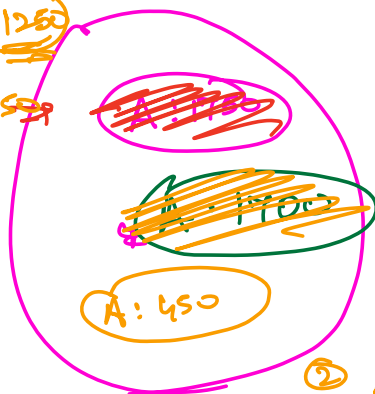
- ① divide people in 2 buckets
 (i) People who have paid extra
 (ii) less

A	B	C	D
1750	-50	-1250	-450

① B → A: 50

② C → A: 1250

③ D → A: 450



Paid Extra



Paid Less

② A ← C
 1700
 450
 ③ A (450)

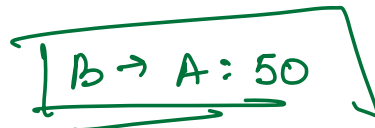
D (-450)

- ② while both buckets are not empty:
- a.) take out any one from both buckets



- b.) make a transⁿ of $\min(A, |A|)$
 $\Rightarrow 50$

from lesser to more



- c.) put the non settled case back



~~C: 300~~ ~~D: 100~~

①

C: 100

D: -200

$$\min(100, |-200|) = 100$$

D → C: 100

⇒ at every step the person with lower abs value will sell

②

A: 200

D: -200

D → A: 200

③

A: 200

B: -200

Why Practical

① Will anyone who had paid extra every have to pay? NO

② Will anyone be asked to pay more than they have to? NO

N-1 transⁿ

Can we do better?

Yes

But doing better is an NP-Hard Problem

→ the most hardest problem of CS

→ No polynomial time solⁿ possible

→ do brute force

A → B
C → D
A → E

A → D
C → B
E → F

A → F
C → B
E → D

A → B
C → F
E → D

A →

O(N!)

① → N → O(N)

② → N → O(N)

③ Algo X → < N-1 trans

O(N log N) O(N)

H/W

① Do class diag And scheme design on
S/W

⇒ only 3 entities in CD / ④

② Dry Run the Algo's of settle up yourself.