```
public class solve {
    int foo(int x){
        if (x <= 2) { return x }
        return foo(x-1) + foo(x-2) + foo(x-3)
    }
}
```

foo(5)

last intermediate cut.

> 66.6%

63 → 90%

70

Advance level↑

< 2 questions { reattempt }

ensure to solve >= 2 questions

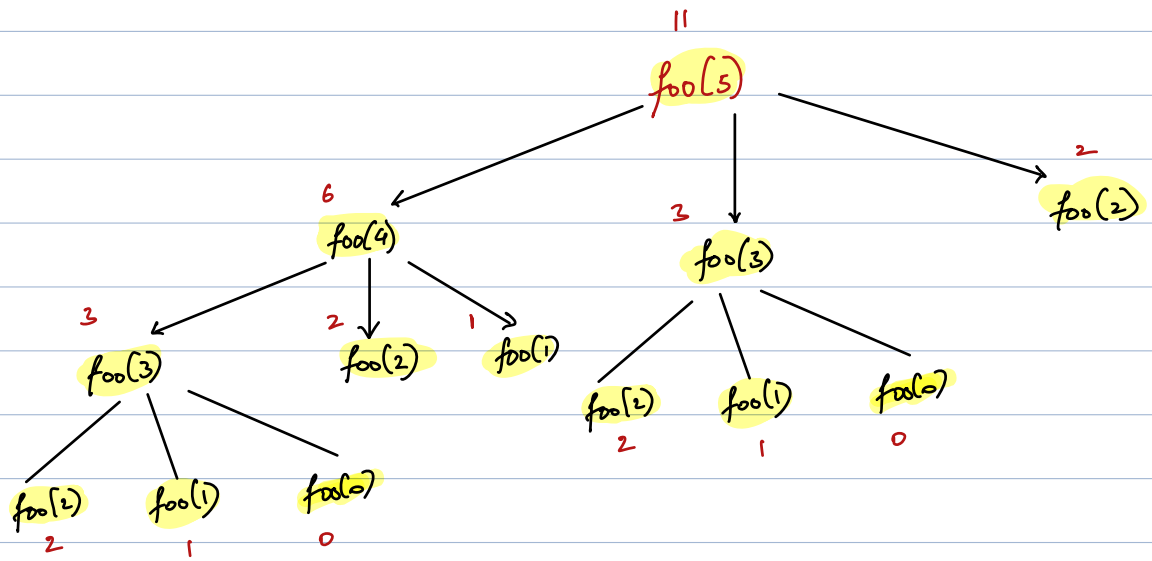
100% complete backlog

1) Reach out to peers on WA group
2) Ping me on WA
   Friday, Sat, Sund, Mon
   15mins every
3) Reach out to TA

set a goal to complete 100% of question.

```
public class solve {
    int foo(int x){
        1  if(x <= 2) { return x }
        2  return foo(x-1) + foo(x-2) + foo(x-3)
    }
}
```

Ans : 11



Q2) Write a function that takes in the string A representing the story and returns the first letter that repeats in a string.

NR R

0 1 2 3 4 5

a d b c d b

d → 2
b → 2
c → 1
a → 1

0 1 2 3

c g c g

0

x

a → repeating first

a b c c a b

Ans: c

HM

{}

a | b | c | c | a | b

{a:1} | {a:1, b:1} | {a:1, b:1, c:1}

Repeating

HM

< Key , Count >

↓

char

Hash set

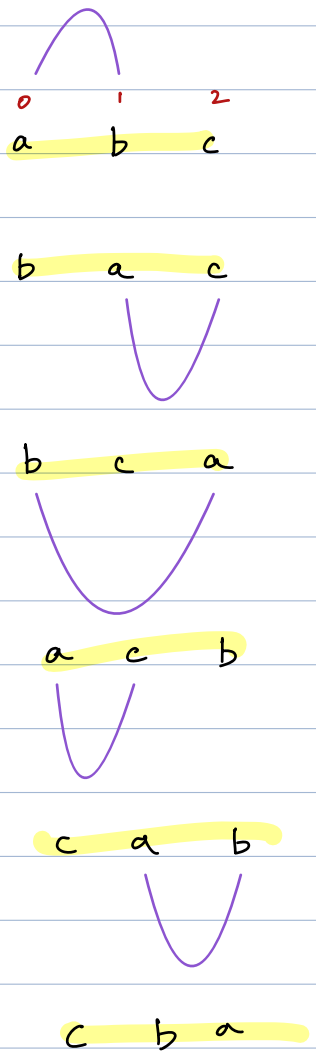
TC: ~~O(N)~~ → O(1)

SC: O(1)

```
hs = {}
for (i=0; i<N; i++) {
    if (hs.contains(A[i])){
        return A[i]    # convert to string
    }
    hs.insert(A[i])
}
return ""
```

a - z

(Q3) Given a string check if it can be converted to a palindrome or not by doing any no. of swaps.



obs 1) Using N no. of swaps you can permut the string in any order as you want

c a d ⟶ dac ✓

c a d ⟶ a d c ✓

c a d ⟶ cat ✗

you can only permute and not change

a b b a c ⟶ Palindrom: a b c b a | Yes
b a c a b

a b c b ⟶ Not possible

len = 1 x , y , z ⟶ True

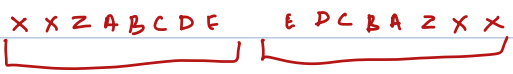
len = 2 x y ⟶ False

both chars have to be same

len 3 =>
- all are same
- 2 same elements

b * b
↑

N is even



all chars must occur even no. of times

N is odd

all chars must occur even no. of times

except 1 and only one which occur odd no. of times

Hashmap < char , int >

s[i] Count

1) Create HM

2) Iterate over HM and check no. of s[i] having odd count

3) check if odd occuring elements ≤ 1