

→ Josephus problem → Google ✓

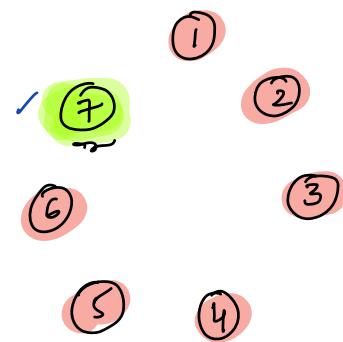
→ N^{th} Magical Number ✓

→ Majority element ↗ Google
→ Majority element ↘ Google }

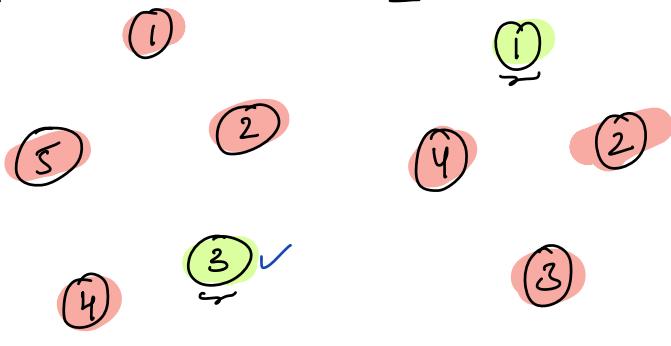
Josephus Problem:

N people standing in circle, person 1 kills next person to its right side & passes knife to the next available person to his/her right, find last person standing

$$\underline{N=7}$$



$$\underline{N=5}$$



$$\underline{N=4}$$



obs1 Ans is prime number ✗

obs2 Ans is odd number ✓

$N=1$



Observations

$$\underline{N}$$

$$1 = 2^0$$

Person standing

1

$N=2$



$$2 = 2^1$$

1



$$4 = 2^2$$

1

$$6$$

5

$$8 = 2^3$$

1

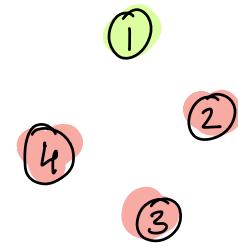
$$16 = 2^4$$

1

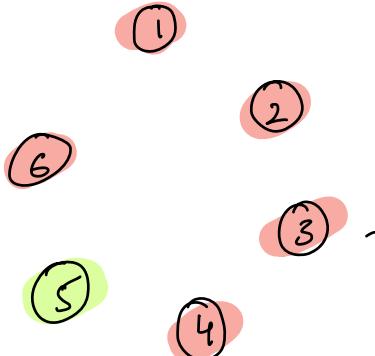
$$32 = 2^5$$

1

$\underline{N=4}$

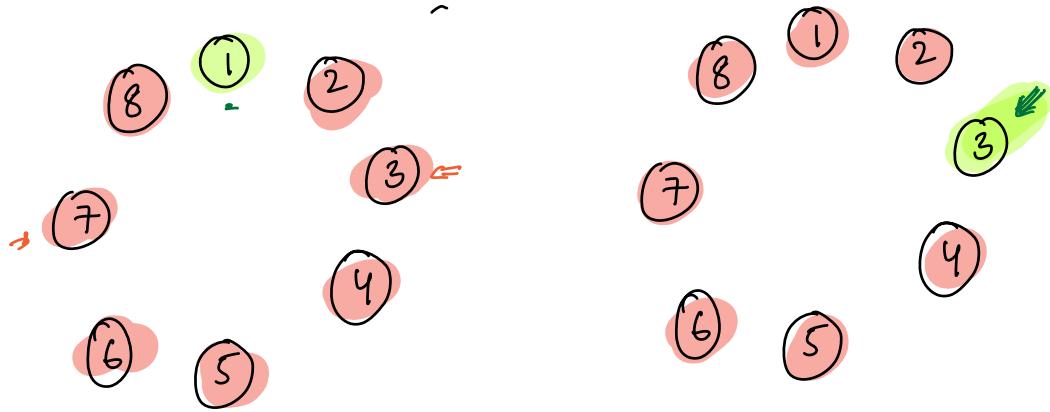


$\underline{N=6}$

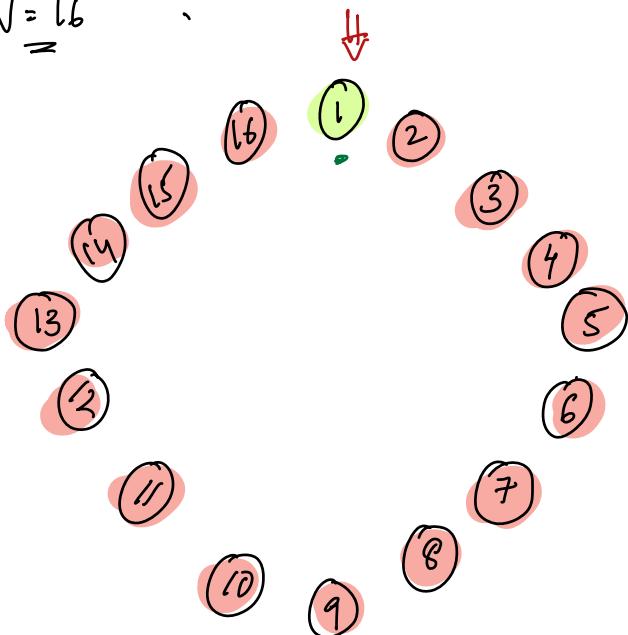


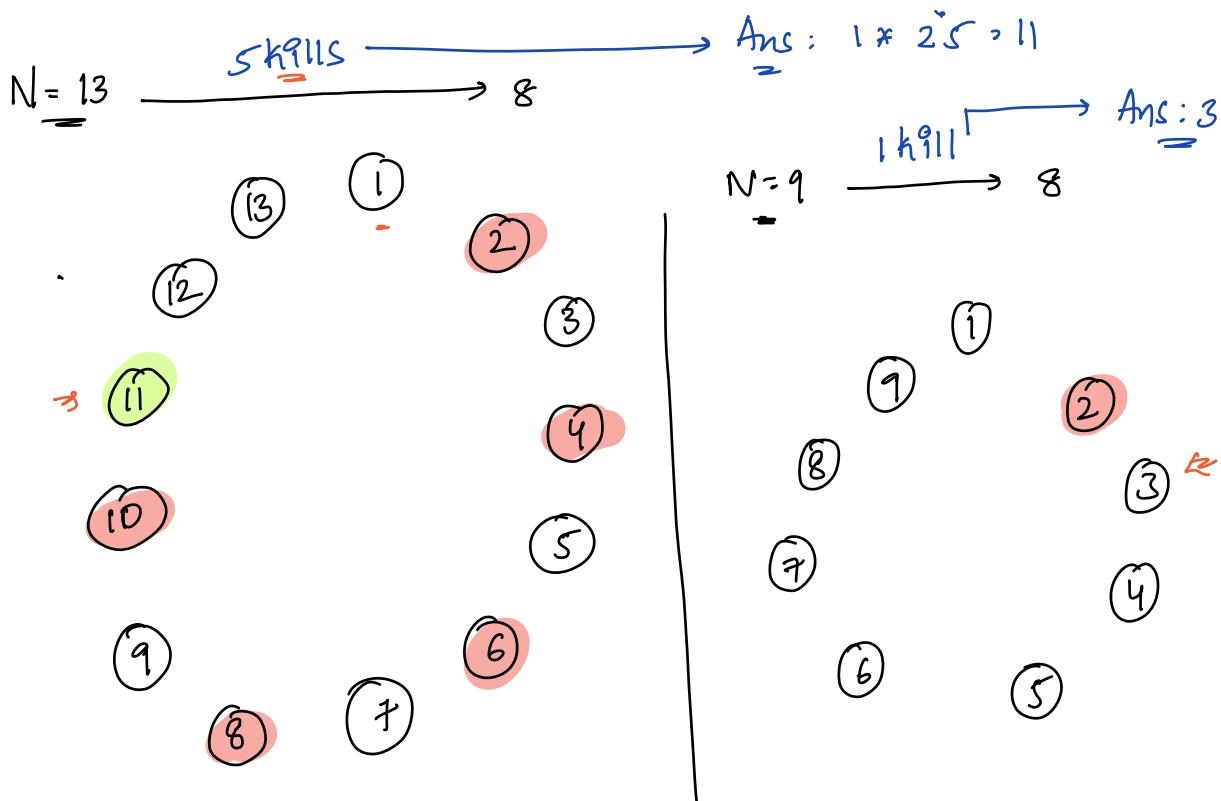
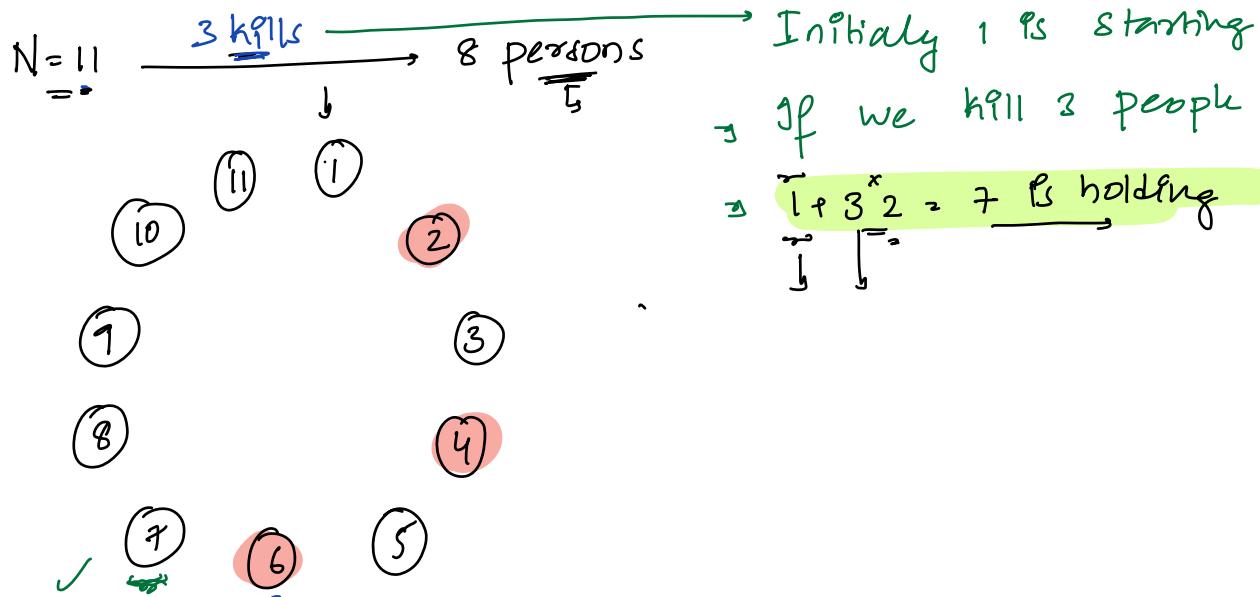
Obsl: If N is a power of 2
Who ever starts will win

N = 8



N = 16





→ $N = \underline{100}$ kills: 36, winner: $2 \times \underline{36} + 1 = 73$

$N = \underline{31}$ kills: 15, winner: 31

For a given N , starting at 1

$$i=0 \quad N=100$$

$$\therefore \underline{x} = (\text{Nearest power of } 2 \leq N) \quad \text{by TODO}$$

$$\therefore \text{kills} = N - x$$

$$\therefore \text{ans} = 2^{\underline{x}} \text{ kills} + 1$$

$$2^0 < 100 \checkmark$$

$$2^1 < 100 \checkmark$$

$$2^2 < 100$$

$$2^3 < 100$$

$$2^4 < 100$$

$$2^5 < 100$$

$$2^6 < 100 \quad \text{ans} = 6$$

$$2^7 > 100 \times$$

Majority Element 3

Q1) Given an arr[N], Return if there exists a no with

Note: $\xrightarrow{\text{No Extra Space}}$, $\xrightarrow{\text{TC: } O(N)}$ $\xrightarrow{\text{frequency } > \frac{N}{2}}$
 cut off element

Q1) $\xrightarrow{\substack{\uparrow \\ \downarrow}} \text{arr}[6] = \{1, 6, 1, 1, 2, 1\}$ freq $> \frac{6}{2} > 3$?
 freq(1) $>= 4$ freq $>= 4$?

Q2) Quiz1: 3 4 3 6 1 3 2 5 3 3 3
 freq(3) $>= 6$ freq $> \frac{11}{2} > 5$
 freq $>= 6$

Q3) Quiz2: 4 6 5 3 4 5 6 4 4 4
 freq(4) $>= 6$ freq $> \frac{10}{2} > 5$
 freq $>= 6$

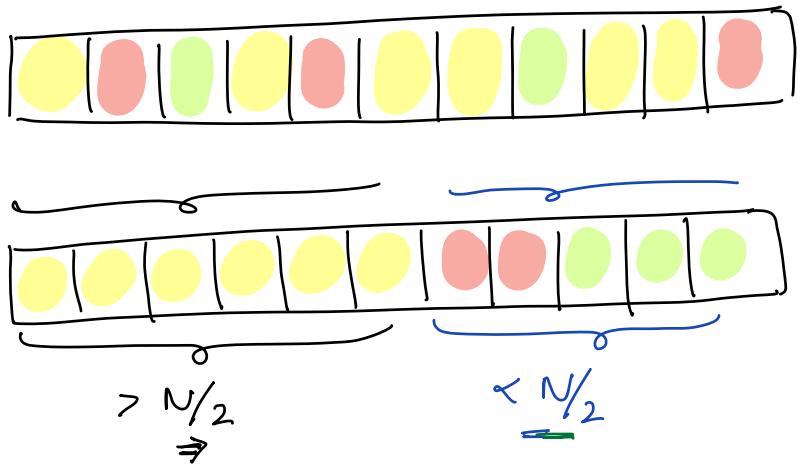
Solution

1) Using 2 nested loops
 Iterate & get frequency
 of each element $> \frac{N}{2}$
 $\underline{\text{TC: } O(N^2)}$ $\underline{\text{SC: } O(1)}$

2) Similar elements will
 come together.
 Sort + get frequency for
 every element in a single loop
 $\underline{\text{TC: } (N \log N + N)} \rightarrow O(N \log N)$

Observations

Ques 3:



obs 1: No. of majority Elements can be at $\underline{m_{\max}} = 1$

obs 2: If say we do have majority Element

- freq $> N/2$
- All other elements combined $< N/2$
- freq of majority elements $>$ All other elements combined

13 PLA Seats, 13 \rightarrow 11 \rightarrow 9

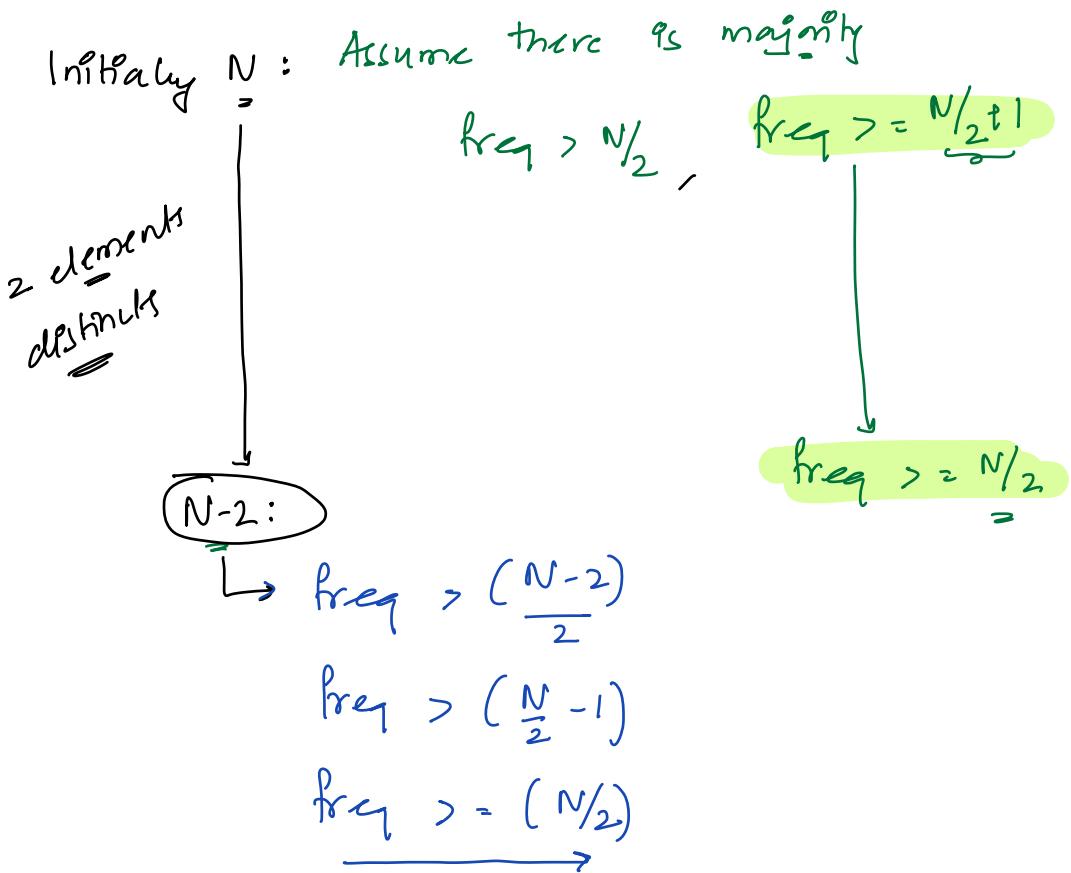
Satyajit:

Ajay:

Athira:

} Priyantha
Sandeep T
Kaspa

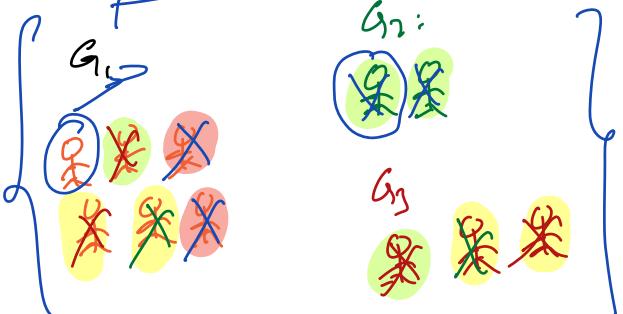
obs3: If we delete 2 distinct entries, majority element won't change.



Idea: If we delete 2 distinct, majority element

won't change. Majority gang will still win.

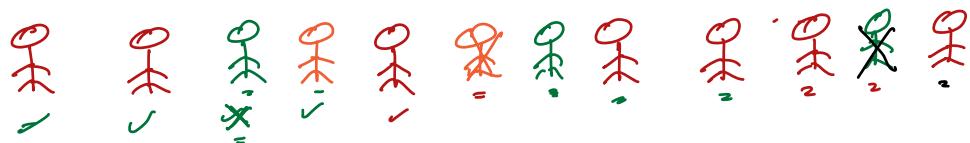
Manner:



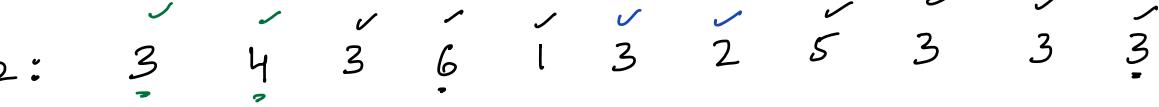
At every point 2 points
people from different
gangs will come &
kill each other

E_n1:

Below are 12 MLA election data



-  
- count it is not signifying
-  your frequency.

E_n2: 

Elc: 

Freq: 

E_n3: 

Elc: 

Freq: 

Ex: 3 6 7 3 3 8 7 8 3 3 Input has no majority

Elc: * * * *

Freq: * * 0 0 * * * 2 Elc: 3 → Not A majority elem
Freq: 2

Note: Iterate & check if elc is majority element or not.

Note,

Idea:

① Initial State: Elc, freq

② Check if Elc is majority

Pseudo code →

II: O(NP)

① Elc = arr[0], freq = 1

i = 1; i < N; i++) {

```

if ( freq == 0 ) {
    Elc = arr[i], freq = 1
}
else if ( arr[i] == Elc ) {
    freq++
}
else {
    freq--
}

```

② Check if Elc is majority

int cut = 0

i = 0; i < N; i++) {

```

if ( arr[i] == Elc ) {
    cut++
}
if ( cut > N/2 ) {
    majority is Elc
}

```

TC: O(N+N) → O(N) SC: O(1)

MajORITY Element 2 } TODO 3

An element is said to be majority element if its frequency $> N/3$

$ar[8] : 4 \ 5 \ 3 \ 2 \ 5 \ 2 \ 4 \ 2 :$

Ex1:

$ar[10] : 10 \ 6 \ 6 \ 7 \ 6 \ 4 \ 10 \ 10 \ 4 \ 10$

$ar[10] : 10 \ 6 \ 8 \ 8 \ 6 \ 10 \ 6 \ 6 \ 10 \ 10$

Ex3:

$ar[7] \quad 3 \ 6 \ 3 \ 8 \ 6 \ 4 \ 4 \ 4 \ 3$

Problem 3:

$i = 1$ Problem }

Put $c = 0$ = No: of factors

$i = 1; i \times i \leq N; i++ \{$

To check No prime a Not

count no: of factors == 2

if ($N \% i == 0$) {

// i is factor N/i is also factor

if ($i == N/i$) { $i == N/i, i \times i == N, \boxed{i = \sqrt{N}}$

$c = c + 1$

} If N is perfect we will
go to if case

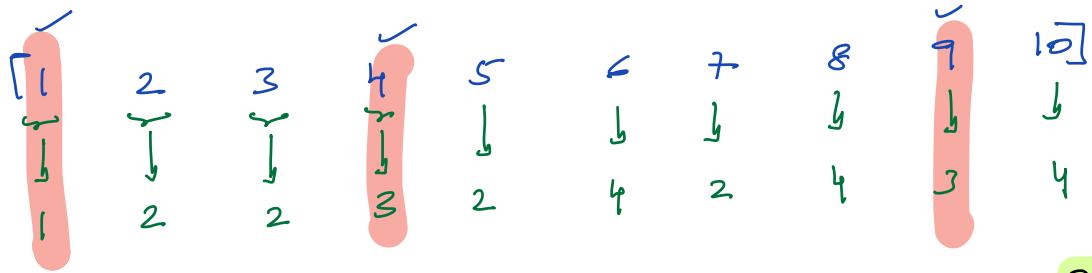
else {

$c = c + 2$

print(c); } = No: of factors

Q2) Given N , no: of elements from $[1-N]$ have odd no: of factors.

$$N = 10 \Rightarrow \underline{\underline{Ans = 3}}$$



\Rightarrow An Element will odd factors if element is Perfect Square

We need to get no: of perfect square from $1-N$?

So TODO

$$n = 100$$

$$n = 25$$

$$n = 50$$

Amazon: { N^{th} Magical Number }

Given a no: N , { Return N^{th} Magical Number }

Magical Number: { That it can be expressed sum of unique power of 5 or power to 5 }

$$50 = \underline{\underline{5^2 + 5^2 \times 1}}$$

$$\underline{\underline{5^1 + 5^1 \times 3}} \xrightarrow{\text{Inversed}} = 0$$

$$\begin{array}{cccc} 5^4 & 5^3 & 5^2 & 5^1 \\ 0 & 0 & 0 & 5 \times 1 \end{array}$$

1st Magical Number ::

2nd

3rd

4th

5th

6th

7th

8th

9th

	0	0	0	$\cancel{5 \times 1}$	=	5
	0	0	$5^2 \times 1$	0	=	25
	0	0	$5^2 \times 1$	5×1	=	30
	0	$5^3 \times 1$	0	0	=	125
	0	$5^3 \times 1$	5×1	5×1	=	130
	0	$5^3 \times 1$	$5^2 \times 1$	0	=	150
	0	$5^3 \times 1$	$5^2 \times 1$	5×1	=	155
	$5^4 \times 1$	0	0	0		
	$5^4 \times 1$	0	0	5×1		

= 630

//
13th Magical Number:

$$\begin{array}{r} 4 \\ \underline{\underline{5}} \quad \underline{\underline{5}} \quad \underline{\underline{5}}^2 \quad \underline{\underline{5}}^1 \\ | \quad | \quad 0 \quad | \Rightarrow \end{array} \left\{ \begin{array}{l} 625 + 125 + 5 \\ = 755 \end{array} \right.$$

→ 10th Magical Number:

$$\begin{array}{r} 4 \\ \underline{\underline{5}} \quad \underline{\underline{5}}^2 \quad \underline{\underline{5}}^2 \quad \underline{\underline{5}}^1 \\ | \quad | \quad | \quad | \Rightarrow \end{array} \left\{ \begin{array}{l} 625 + 25 \\ = 650 \end{array} \right.$$

If
0th bit = 5
1st bit = 5²
2nd bit = 5³
ith bit = 5ⁱ⁺¹

// PrandwCode (N)

TODD } Small Thg overflow

P = 5; ans = 0

P = 0; i < 32; P++ {

if (checkBft(N, T)) {

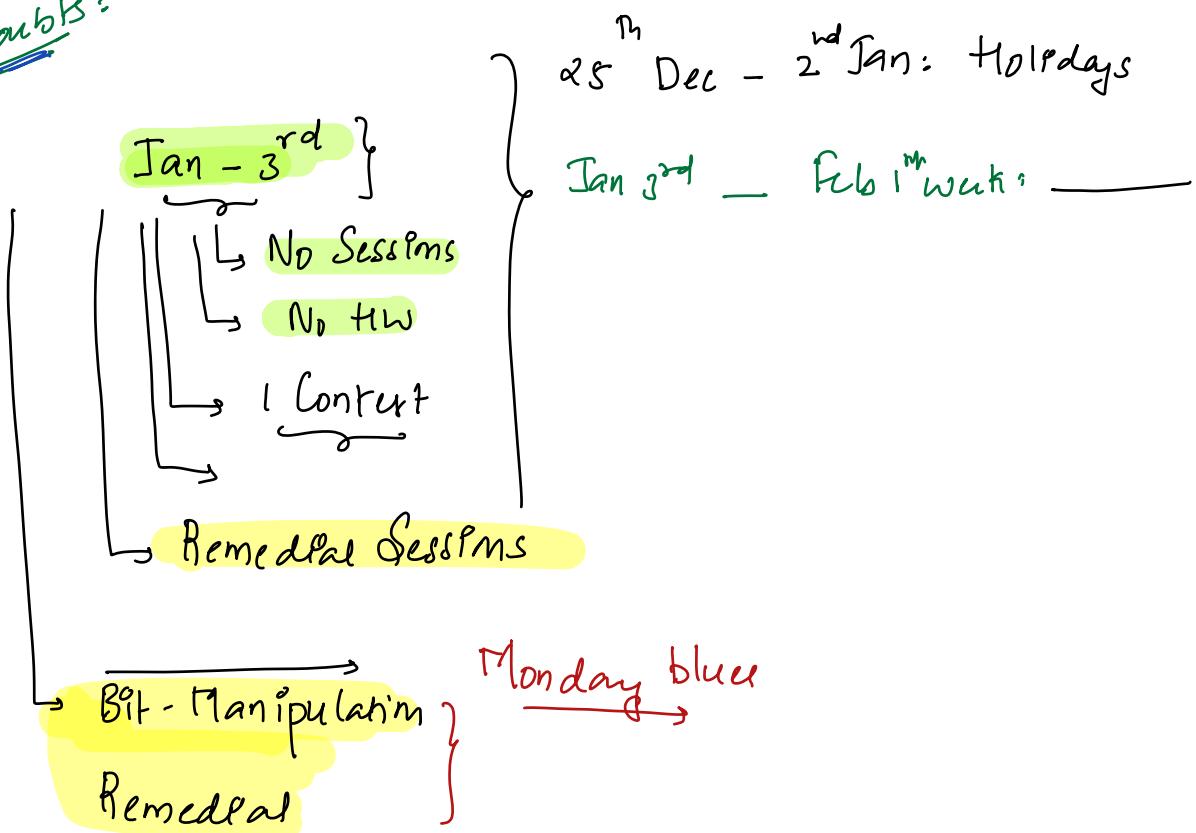
ans = ans + P;

P = P * 5

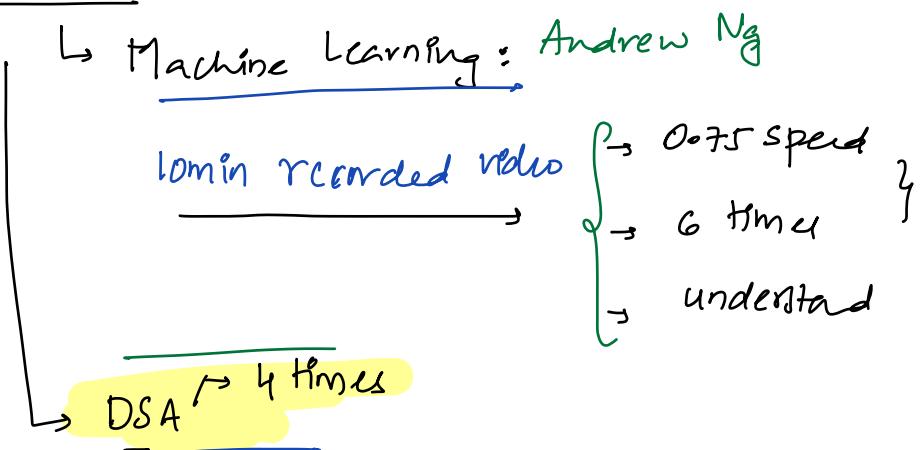
Will Add Thw/CW

by 12 AM

Doubts:

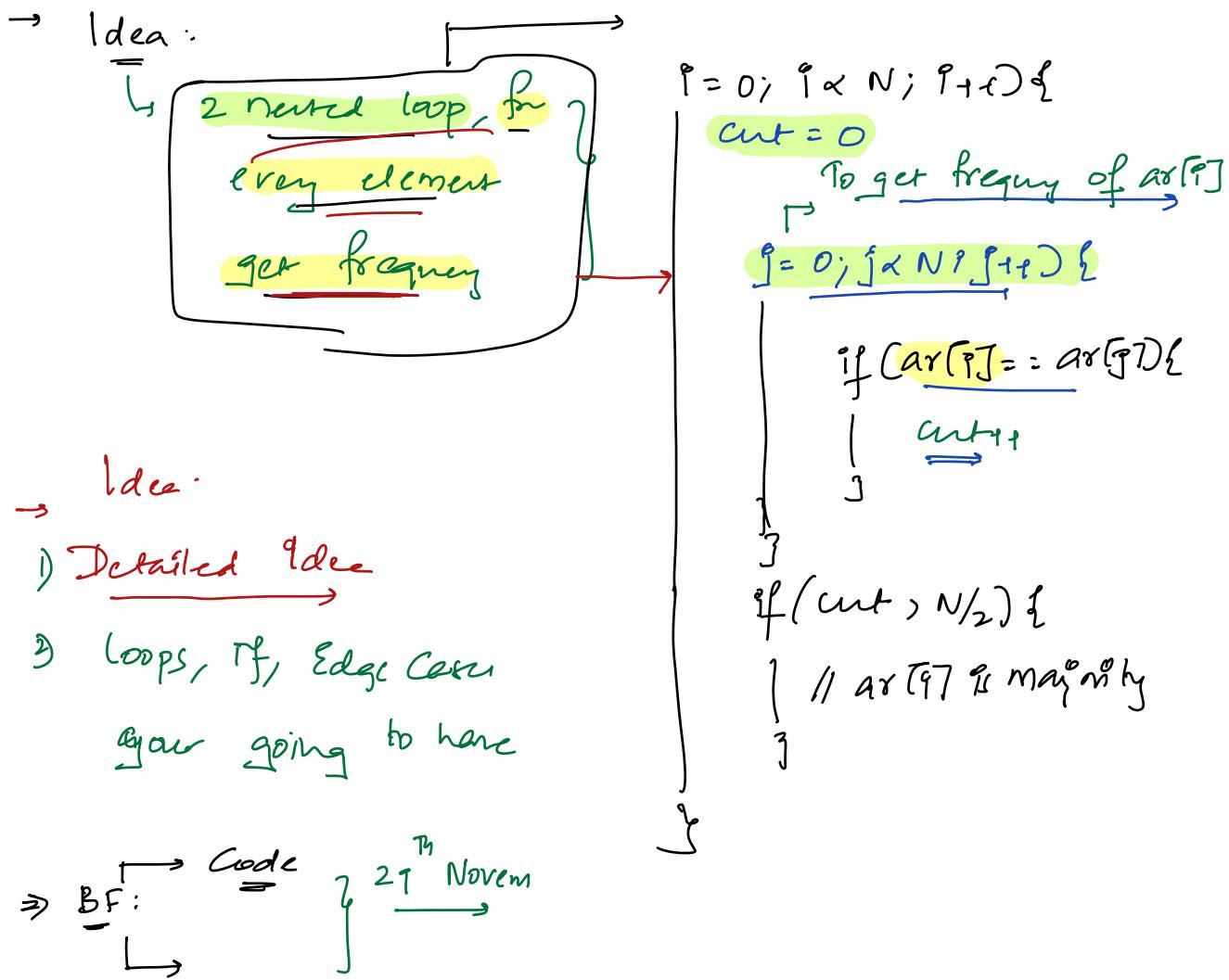


→ Coursera



→ — 13

{ Doubts Session



<

