Today's Content:

- Majority Element
- Josephus Problem

10] Given N+ve elements find majority Element?

An element with freq > N/2 (size)

freq = occurrence of element

Ex1: ar[6] = {1 2 1 6 1 1} : freq(1) = 4 > 6/2

↳ ans = 1, majority element

Ex2: ar[9] = {3 4 4 8 4 9 4 3 4} :

↳ ans = 4, majority element freq(4) = 5 > 9/2

Ex3: ar[11] = {3 3 4 6 1 3 2 5 3 3 3}

↳ ans = 3, majority Element freq(3) = 6 > 11/2

Ex4: ar[10] = {4 6 5 3 4 5 4 4 4 8}

↳ no majority element

freq(4) = 5 > 10/2

↳ not majority

Solutions:

1) For every element iterate & get frequency

TC: $O(N^2)$ SC: $O(1)$ } $N > \log N$ → { for large values of N

2) Eg: ar[10] → { 3 2 4 6 6 3 4 4 2 3 }

Sort → {N Elements} → TC: $O(N \log_2 N)$

ar[10] = { 2 2 3 3 3 4 4 4 6 6 }

↑ 2 ↑ 3 ↑ 3 ↑ 2

Iterate & get frequency of every element : $O(N)$

→ Overall TC: $N \log_2 N + N$ → TC: $O(N \log_2 N)$

3) Given N array, Say 2 majority are present ele1, ele2,

freq(ele1) > N/2

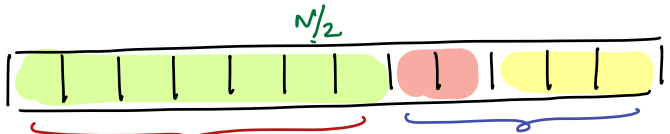
freq(ele2) > N/2

freq(ele1) + freq(ele2) > N → not possible? Because total elements are N

→ At max we can only have 1 single majority

// Say we bring all majority to left

Ex: ar[ ] :



→ { We cannot have any majority Element }

Elections :

Say 15 MLA Seats

Saurabh : 😀 😀 😀 😀 😀 😀 😀 ~~😀~~ ~~😀~~

Ayush : ~~😀~~ ~~😀~~

Cheran : 😀 😀 ~~😀~~ ~~😀~~

// Saurabh's > 15/2 : ✓

→ We remove 2 tickets

8' MLA > 13/2 : ✓

→ We remove 2 tickets

7' MLA > 11/2 : ✓

→ We remove 2 tickets

7' MLA > 9/2 : ✓

Observations :

If we remove 2 different elements, majority element will still remain same

Ex2 : 7 MLA :

Naiya : 😀 😀 ~~😀~~ ~~😀~~

Satish : 😀

Kulay : 😀 😀

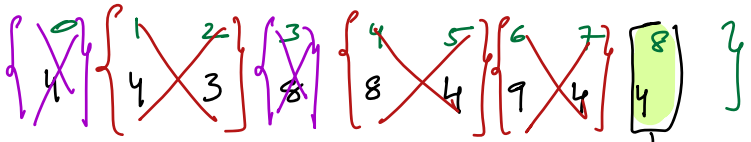
Naiya MLA →

4 > 7/2 : ✓

↓

2 > 5/2 *

Eg1:



majority = 4

freq(4)

5 > 9/2 :

↓

4 > 7/2 :

↓

3 > 5/2 :

↓

2 > 3/2 :

↓

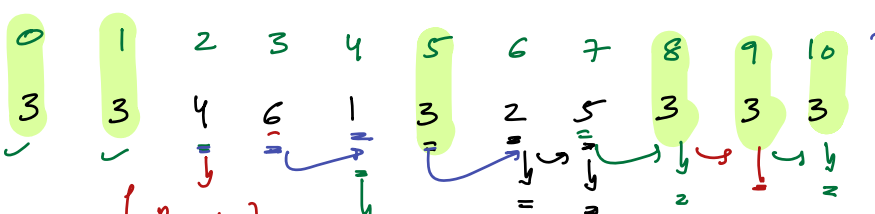
1 > 1/2 : finally we are leftout with majority element

→ check 2 distinct elements?

→

Moore's Voting Algorithm

Eg2:



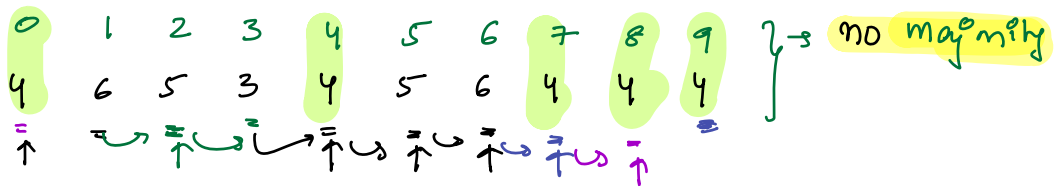
→ majority - 3

Etc: 3 x x 3 → {majority element}

freq: x x x x x x x x x x 3 → {This is not correct frequency of 3}

Private q get freq(3) > 11/2 ✓

Eg3:

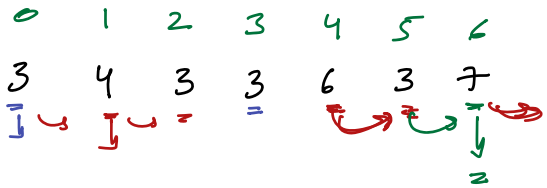


dic: ~~4~~ ~~6~~ ~~5~~ ~~3~~ 4 ~~5~~ ~~6~~ 4 ~~4~~ ~~4~~ → majority element

freq: 1 ~~0~~ ~~1~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ 2

check if 4 is majority : { Iterate & freq of 4 > N/2 }

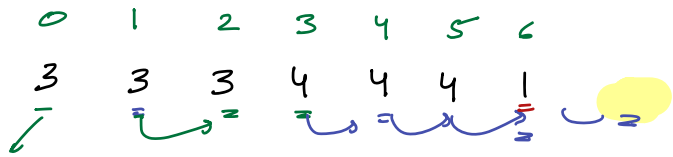
Eg4:



dic: ~~3~~ 3 → { check if 3 is majority
Iterate & get freq(3) > 7/2 }
> n/2 → n = 7 elems

freq: 1 ~~0~~ ~~1~~ ~~2~~ 1 ~~2~~ 1

Eg5:



dic = ~~3~~ 1 → if 1 is majority :

freq = 1 1 ~~2~~ ~~2~~ 1 ~~0~~ 1

Iterate & get freq(1) > 7/2 *

no majority element

Pseudocode:

```
int majority(int ar[]){
    int N = ar.size()
    int el = ar[0], freq = 1;
    for(i=1; i<N; i=i+1){
        if(freq==0){
            el = ar[i], freq = 1
        }
        else if(el == ar[i]){
            freq = freq+1
        }
        else{ // 2 different elements
            freq = freq-1
        }
    }
    int c=0;
    for(i=0; i<N; i++){
        if(ar[i] == el) { c=c+1 }
    }
    if(c > N/2) { return el }
    else { // no majority element is present
           return as per asked in Question }
}
```

TC: O(N)

SC: O(1)

→ longing

9:15 am

c — Cl & duets

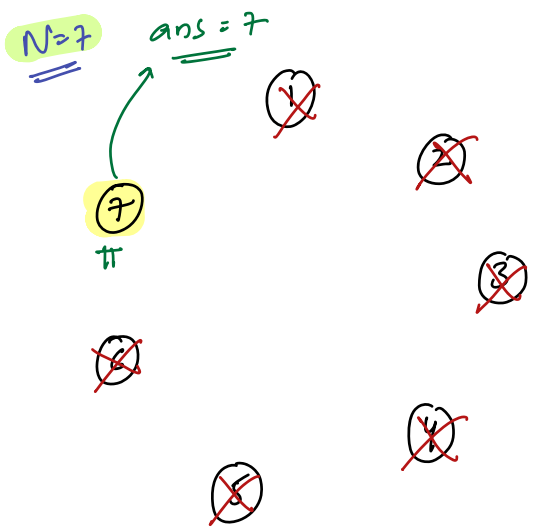
3 → 2

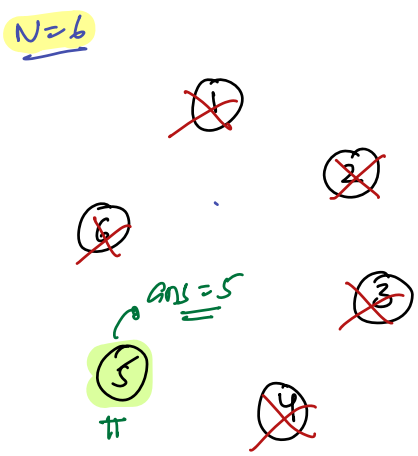
20) Josephus Problem:

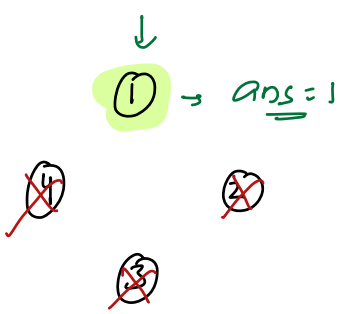
Initially N people are in circle, person 1 has knife, he kills his adjacent clockwise person, and pass the knife to next adjacent clockwise person.

Repeat untill only a single person stands,

who is last person standing ?







obs:

→ 1) If n is odd : n

If n is even : n-1

about worst work

$N = 1 : 2^0$ → ans = 1

$N = 2 : 2^1$

$N = 4 : 2^2$

$N = 8 : 2^3$ → ans = 1

obs: If N is power of 2
who ever starts the game
will win the game

$N = 16 : 2^4$

2^5

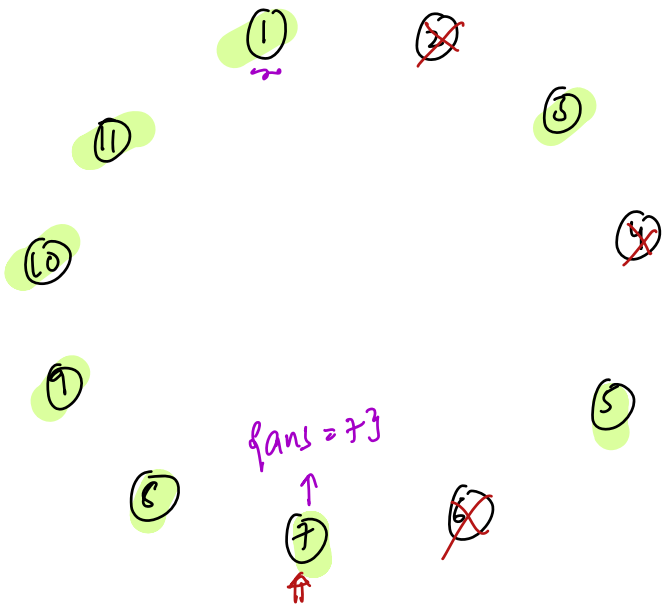
$N = 32$: 1 starts game → 1 wins

obs: If N is power of 2
1 is winner

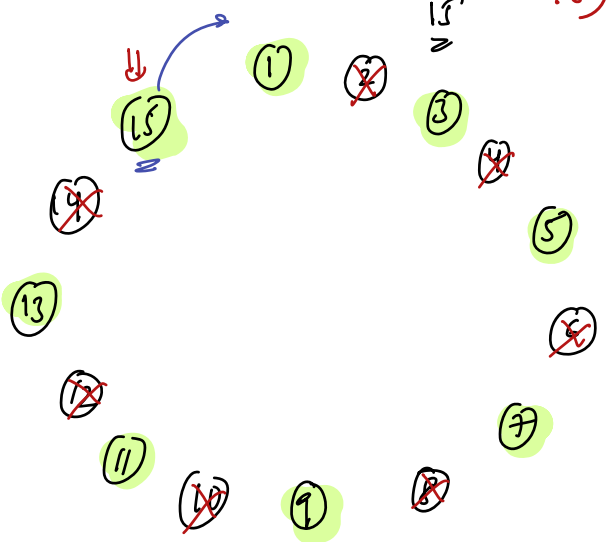
$N = 8$

ans = 3

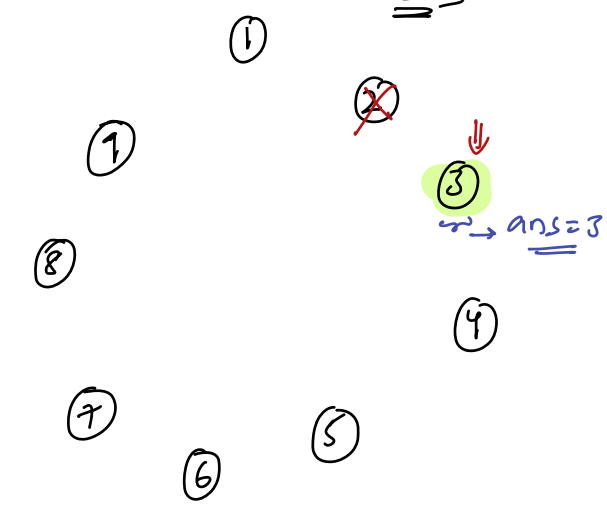
// N = 11 : {3 kills} : 1 + 2(3) → 8



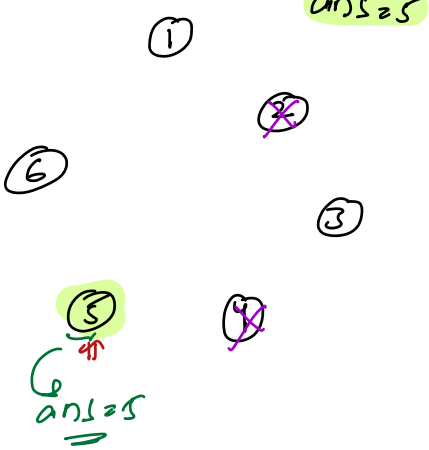
N = 15 : → {7 kills} : {1 + (2)(7)} → {8}



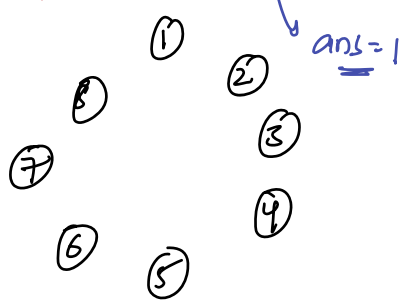
N = 9 : 1 kills → {1 + (2)(1)} → 8



N = 6 : 2 kills → {1 + (2)(2)} → 4



N = 8 okills : {1 + (2)(0)} → 8



N = 100 → {36 kills} → {1 + 36 × 2} → 64

1 starts

nearer power of 2 < 100

ans = 73

// generalize:

```
Put lastpers(N){
    y = nearpower(N)      → Nearest power of 2 <= N
    kills = N-y           → How to get nearest of 2
    return 2*(kills)+1

TODO: {Try with loops on your own}
}
```

→ {Done}

→ {Friday → Sorting}

// Ascii

↳ convert: { char → Put } '0' → 0 :
{ Put → char } '1' → '0' :

↳ Strings :

Concept

$(a+b)\%m = (a\%m + b\%m)\%m$

$(a \times b)\%m = (a\%m \times b\%m)\%m$

$-ve \%m :\rightarrow +ve$

$:\rightarrow -ve$ lag.

$$(a-b)\%m = (a\%m - b\%m + m)\%m$$

// $a\%m = b\%m$

$a\%m - b\%m = 0$

$a\%m - b\%m + m = m$

$(a\%m - b\%m + m)\%m = 0$

$(a-b)\%m = 0$

$\Rightarrow$ m should be a factor of a-b