Arrays and Maths

[1 → N]

Amazon
Adobe
GS

Q.1

There are N doors & a person is standing in front of every door. Initially all doors are closed.

1st person → 1st, 2nd, 3rd, ... Nth (open)

2nd person → 2nd, 4th, 6th, ... (close)

3rd person → 3rd, 6th, 9th, ... (toggle)

4th person → 4th, 8th, 12th, ...

⋮

Nth person → Nth

Return which all doors will be open finally.

eg: N=10.

1
2
3
4
:
10

. . . .

Door → 9 ⇒ All factors of 9 will toggle the door.

↳ 1, 3, 9

15 ⇒ 1, 3, 5, 15

18 ⇒ 1, 2, 3, 6, 9, 18

obs:-1

> Every door will be toggled by its factor

1st 2nd 3rd 4th 5th 6th

If the no. of factors is odd
then final state ⇒ Open.

24 ⇒ 1, 2, 3, 4, 6, 8, 12, 24

N = 100

1 ⟷ 100

2 ⟷ 50

4 ⟷ 25

16 → 1, 2, 4, 8, 16

4 ⟷ 4 → perfect square

N

$\sqrt{N}$ → integer

20

1 → 20

2 → 10

4 → 5

10 → 2

20 → 1

4.5

9 → 1, 3, 9

↳ c —1→ 0 —3→ c —9→ 0

Sqrt( )

N <= 50 . → 7

√50 → 7.

1 → 7

```
for( i=1 ; (i*i) <= N ; i++)
{
;
cout i-1 ;
```

T.C. $O(\sqrt{N})$

$1 \le N \le 10^{18}$

Amzon.

Q.2. N^{th} Magical Number

Given a no. N, return N^{th} magical number.

Magical :: A no. that can be expressed as a unique power of 5.

Magic Number

sum of unique

$5^1 + 5^2 + 5^4 + 5^{10}$

N = 1 $\Rightarrow$ 5^1

N = 2 $\Rightarrow$ 10 $\Rightarrow 5^1 + 5^1$

25 $\Rightarrow 5^2$

N = 3 $\Rightarrow$ $5^1 + 5^2 \Rightarrow 30$

N = 4 $\Rightarrow$ $5^3 \Rightarrow 125$

N = 5 $\Rightarrow$ $5^3 + 5^1 \Rightarrow 130$

N = 6 $\Rightarrow$ $5^3 + 5^2 \Rightarrow 150$

N = 7 $5^3 + 5^2 + 5^1 \Rightarrow 155$

N	Ans	B
1	5^1	0 0 0 1
2	5^2	0 0 1 0

1						
2	25	5^2	0	0	5^2	5^1
3	30	5^2+5^1	0	5^3	0	0
4	125	5^3				
5	130	5^3+5^1	0	5^3	0	5^1
6	150	5^3+5^2	0	5^3	5^2	0
7	155	$5^3+5^2+5^1$	0	5^3	5^2	5^1
8	625	5^4	5^4	0	0	0
⋮	⋮	⋮				

0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0

11th $\Rightarrow$ 1 0 1 1 (4 3 2 1)

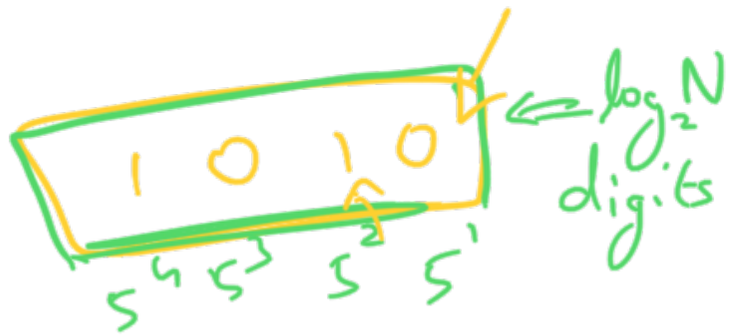
$5^4 + 0 + 5^2 + 5^1 \Rightarrow 625 + 30 \Rightarrow$ **655**

steps:

- Convert N into binary representation $\Rightarrow O(\log N)$

① ⟹ ⇒ O(log N)

② ⟹ Convert representation into power of 5.



$N \xrightarrow[1]{/2} N/2 \xrightarrow[2]{} N/4 \xrightarrow[3]{} N/8 \cdots \xrightarrow{K} 1$

$$\frac{N}{2^K} = 1$$

$$N = 2^K$$

$$\log_2 N = \log_2(2^K)$$

$$\log_2 N = K$$

Google

Q.3 Majority Element

Given array of size N. (positive number)

Return if there exists a no. with frequency > N/2, else return -1!

[without extra space]
S.C ⇒ O(1)

eg. [1, 6, 1, 1, 2, 1]

N = 6.

N/2 = 3.

freq[1] = 4.

Quiz : [3, 4, 3, 6, 1, 3, 2, 5, 3, 3, 3]

N = 11

N/2 = 5.

freq[3] = 6

Ans = 3

Quiz : [4, 6, 5, 3, 4, 5, 6, 4, 4, 4]

N = 10

N/2 = 5.

freq[4] -->

Ans -

No M.E:

Brute force : 2 for loops.

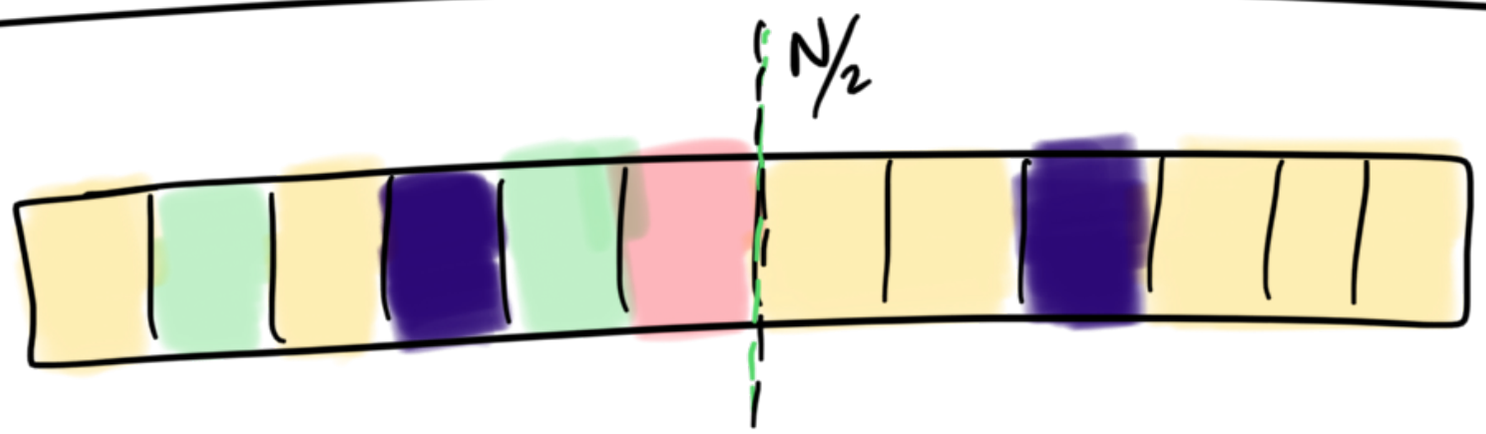
```
for(i = 0 → N-1)
    cnt = 0;
    for(j = 0 → N-1)
    {
        check (a[i] == a[j])
            cnt++
    }
if cnt > N/2 → return a[i]
```

T.C => $O(N^2)$

S.C => $O(1)$.

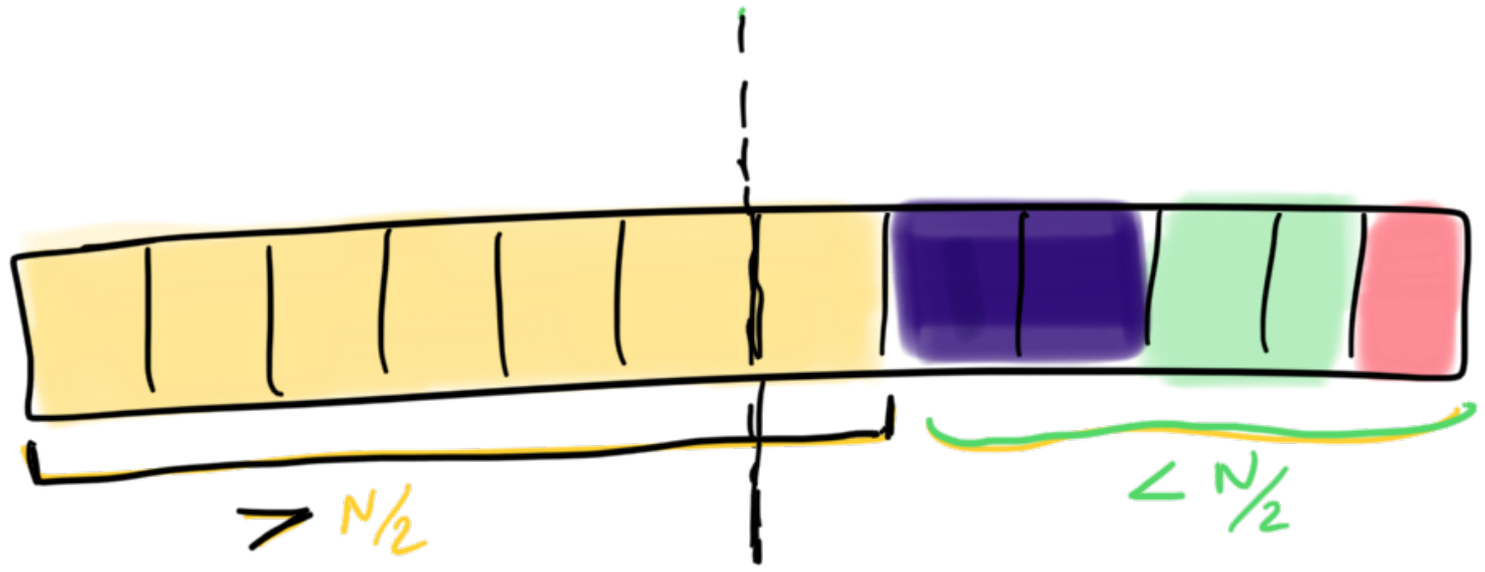
Sorting → space might not be const (haven't learnt yet)

=> T.C => $O(N \log N)$



N = 12

N/2 = 6



Obs: ① Only one Majority element can be there in the array.

② The count of M.E. will be greater than the combined counts of all other elements.

yellow party

[Green
Red
purple] gatbandhan

N.

12

Y

7

If one yellow & one other party

10	6
8	5
6	4
4	3

} is removed everytime

(3) If we remove 2 distinct elements, finall M.E. remains the same.

eg [3, 4, 3, 6, 1, 3, 2, 5, 3, 3, 3] N=11.

M.E => 3.

N	M.E
11	6

Moore's Voting Algo

M.E : (-1) (-1) (-1) (-1) (-1) 3

(0) (0) 0 0 0 0 3

9	5
7	4
5	3
3	2
2	1

count

At every step

① If current element is M.E.
then count ++

② Else if M.E. exist
count --

else if M.E. doesn't exist
make(current) == M.E
Count = 1

Next eg.

[3 3 2 3 1 5 3]

potential candidate

M.E.	(-1)	3	3	3	3	(-1)	3
Count	(0)	1	X	1	X	0	1

eg. [3 4 3 5 3 6 1] ⇒ No M.E is present.

M.E	3	1	3	-1	3	-1	1
Count	1	0	1	0	1	0	1

Finally iterate & check count of the remaining M.E. element.
$> N/2$

for → cnt++ → cnt > N/2

[1 2 3 4 5]

return -1

M.E	5	(-1)	5	(-1)	5
Count	1	0	1	0	1

cnt = 4. N = 6.

[4, 3, 4, 4, 4, 1, 5]

Eg:

M.E	✗	(-1)	✗	✗	✗	✗	4
Count	✗	0	✗	2	3	2	1

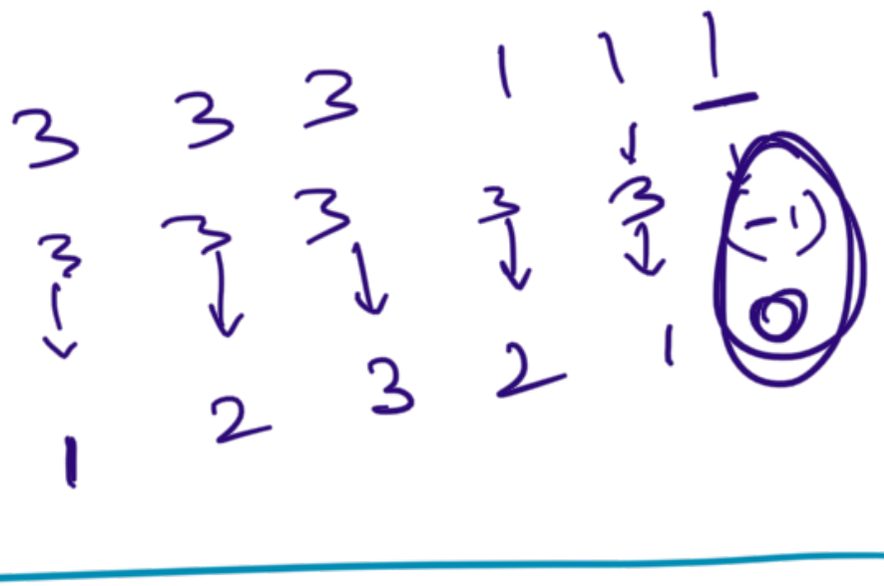
have to validate that it is M.E

M.E = 4

T.C ⇒ O(N)

S.C ⇒ O(1)

(extra)

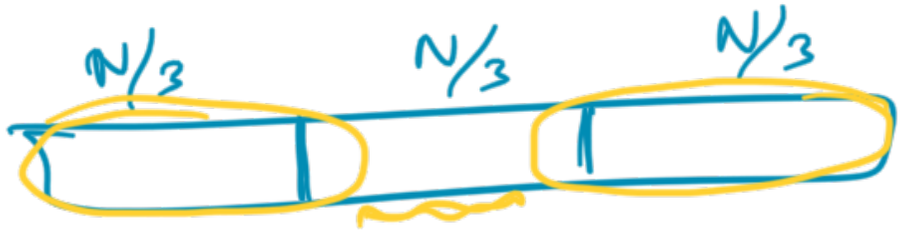


H.W. M.E ⇒ freq > N/3

How many M.E can exist? ⇒ atmax 2.

ME1
Count

ME2
Count



Break : 10:53

$N = 7$

$N/2 = 3$

[2, 3, 1, 3, 2, 3, 4]

4

validate ?

Imp

⇒ Ans = -1

⇒ 1

1 2 5 8 6 6 4 → -1

16 → validate ⇒ freq = 2

$N = 7$. $N/2 = 3$

Ans = -1

Validat^n

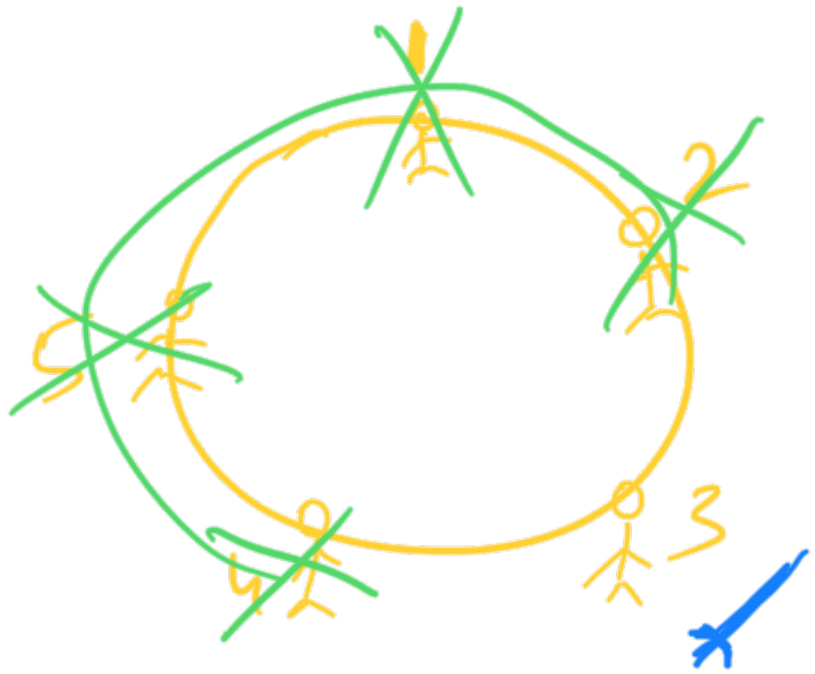
```
for(i=0 -> N)
  if(a[i] == m.E)
    cnt++
if(cnt > N/2)
  print(ans)
else -1
```

10 min

Toy out example

Q.4

Josephus (Jewish)

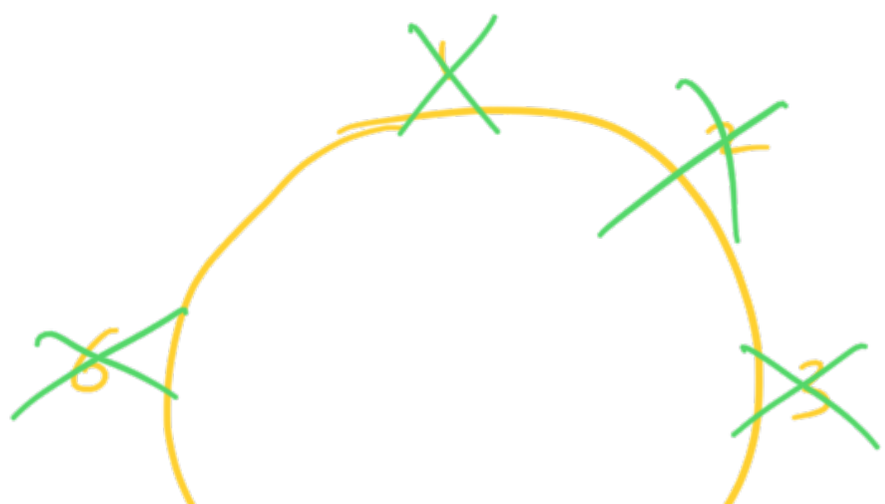


If there are N people in the circle, where Josephus should stand inorder to save his life?

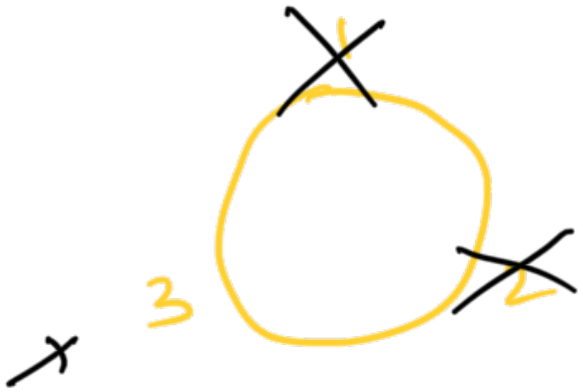
N = 1

N = 2

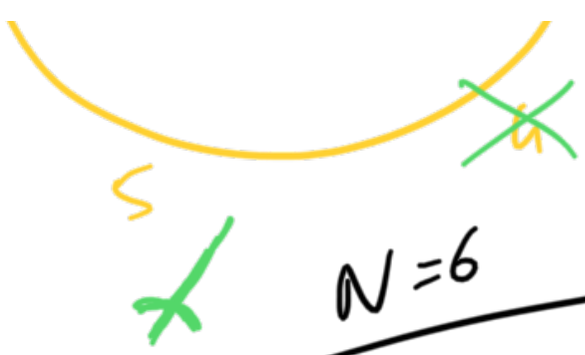
Ans = 1.



N = 3.



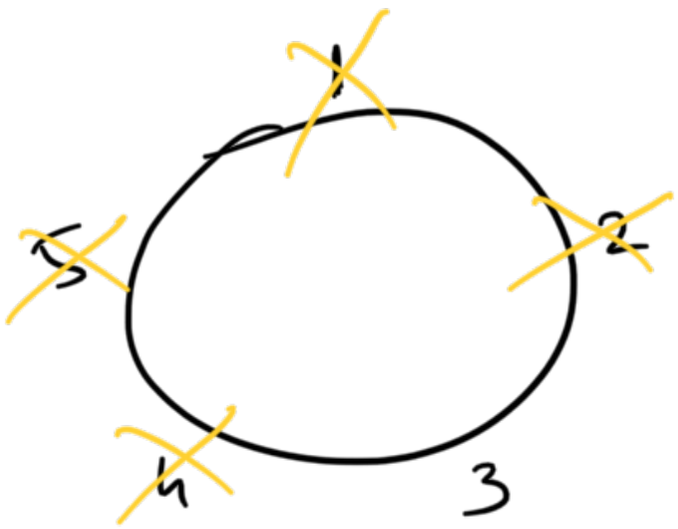
Ans = 3.



N = 6

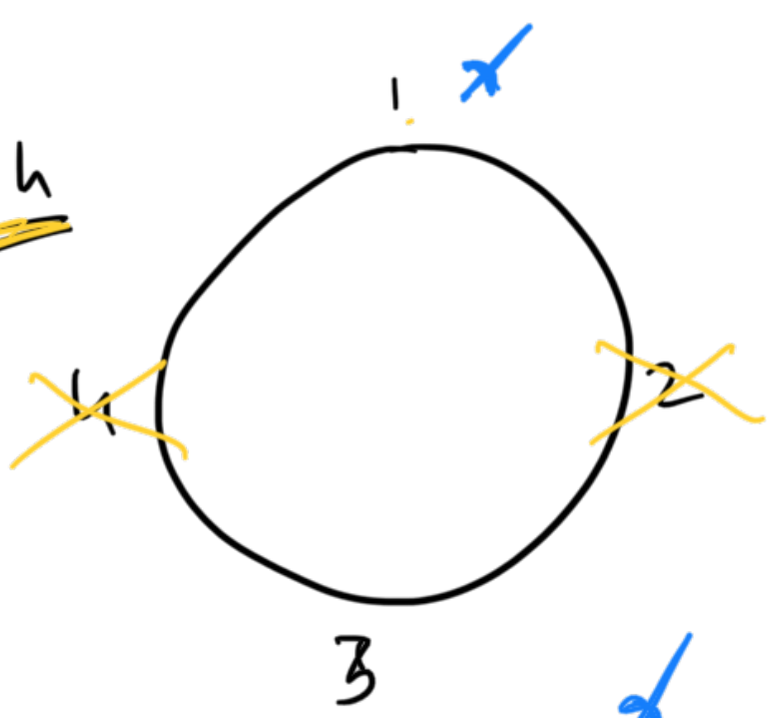
Ans = 5

N = 5



Ans = 3.

N = 4



starting pos = 1

N = 2

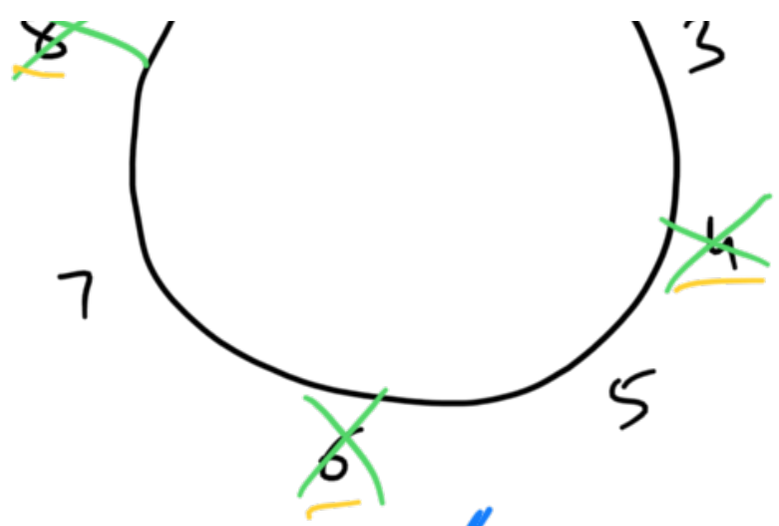
Ans = 1

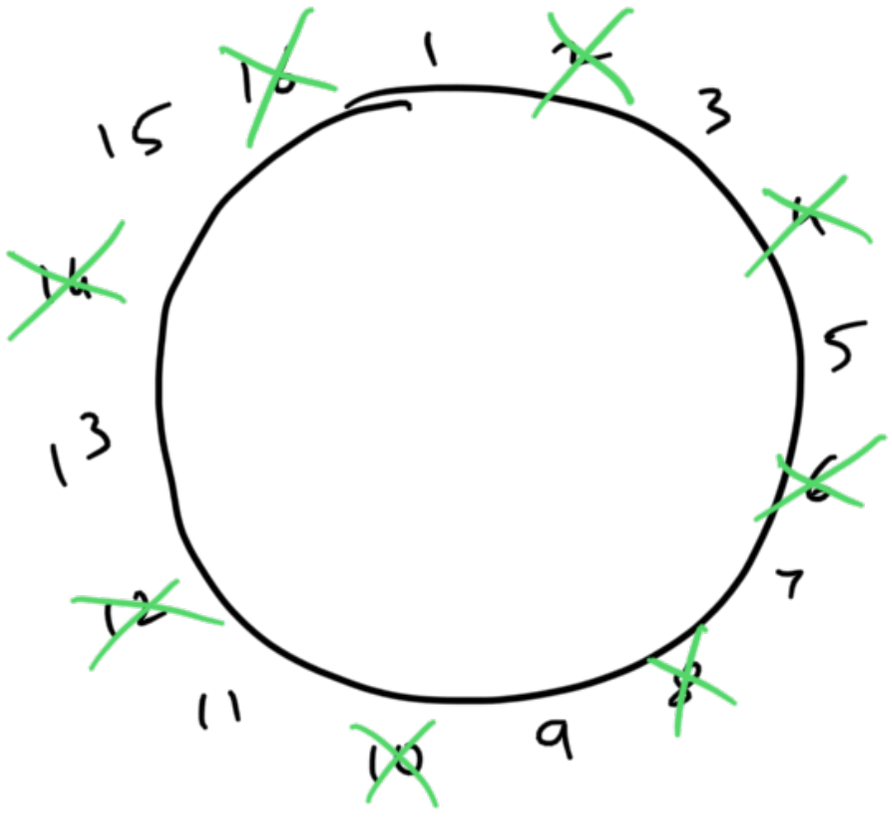
N = 8

starting pos = 1

N = 4

Ans = 1





N=16

starting pos = 1 → Ans = 1

N = 8

N	
1	
2	⇒ Ans = 1
4	

obs ①

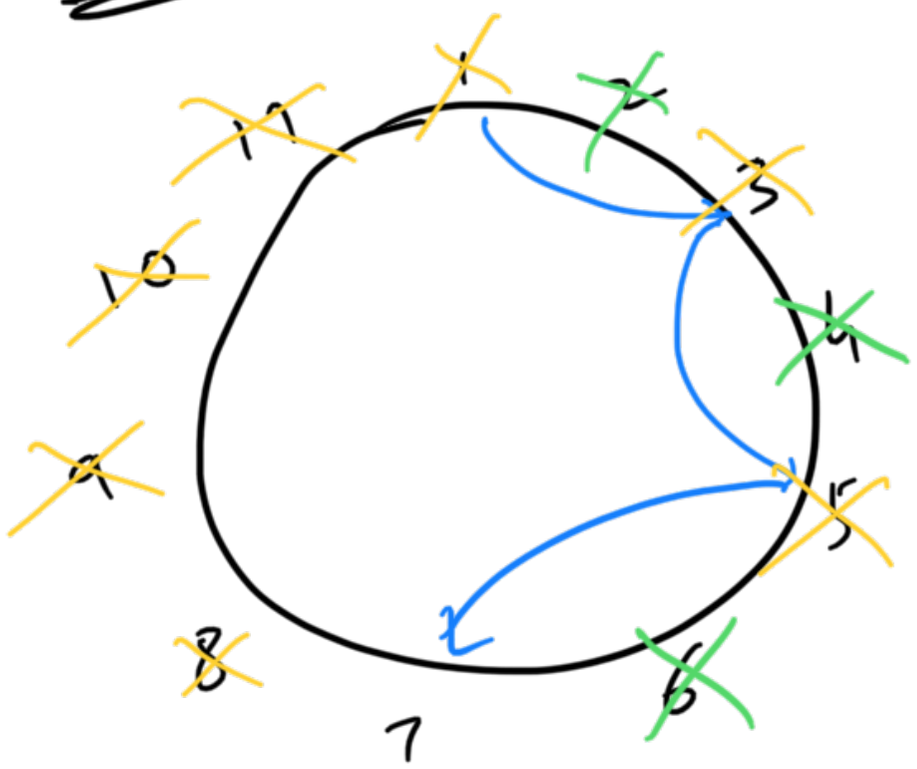
If no. of people is a power of 2

$N = 2^x$

then person who starts the Killing wins

8
16
32
64

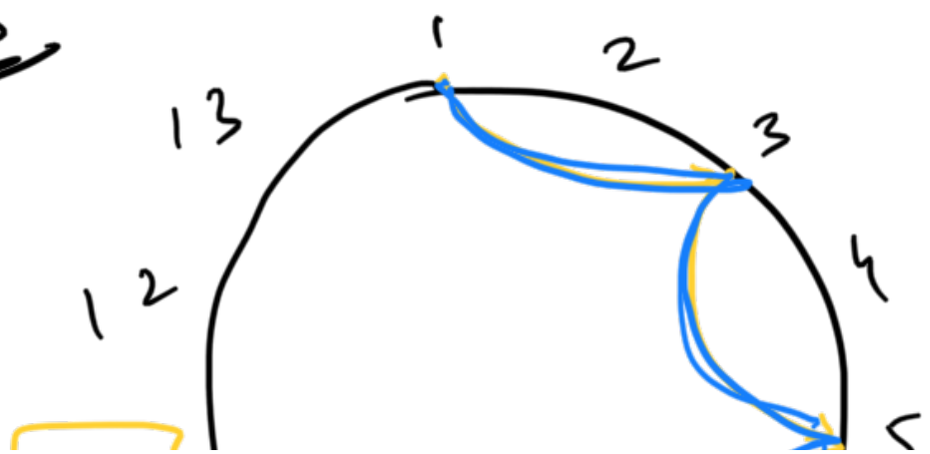
N = 11



N	sword pos
11	1
10	3
9	5
8	7

Winner ⇒ 7

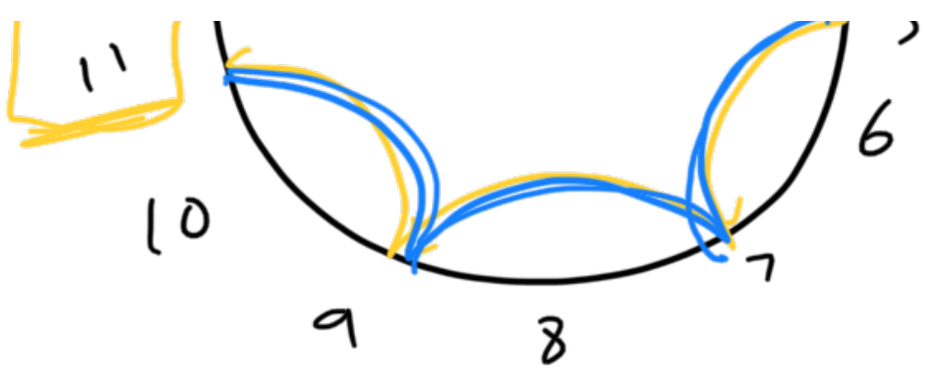
N = 13



Closest power of 2 ⇒ 8

13 − 8 = 5

Ans = 11



Ans = 11.

S^{th} position person

$5 * 2 + 1$

N =	sword
13	1
12	3
11	5
10	7
9	9
8	11

=> winner!

After killing x people,
position of sword $\Rightarrow 2x + 1$

N = 100

$x = 100 - $

$= 36$.

person who wins → $36 * 2 + 1$

$= \boxed{73}$

$2x + 1$

Doubts...

Find all squares below N?

N

(2) $x + 1 > N$?