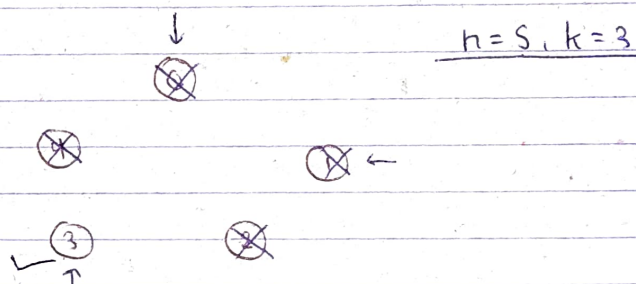


Day - 94

Recursion - 15

\* Josephus Problem (Predict the winner):

- ⇒ We have 'n' no. of people & a value of k.
- ⇒ Then start from any person and ~~counts~~ then jumps to next person upto k times.
- ⇒ And where the pointer stop, that person eliminates.
- ⇒ Now pointer goes to next person.
- ⇒ After that do the same thing.
- ⇒ And when ~~the~~ only one person left, that the winner.



n = 6, k = 4

0	1	2	3	4	5
01	01	01	01	0	01

$$\text{index} = (\text{index} + 1) \% n$$



Code

```

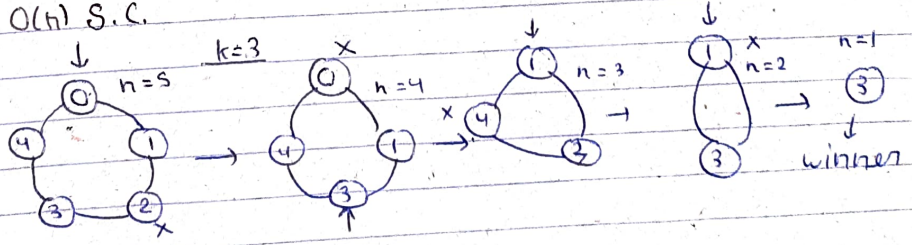
int winner (vector <bool> &person, int n, int
index, int person_left, int k) {
    if (person_left == 1) {
        for (int i=0; i<n; i++) {
            if (person[i] == 0)
                return i;
        }
    }
    int kill = (k-1) % person_left;
    while (kill != 0) {
        index = (index+1) % n;
        while (person[index] == 1)
            index = (index+1) % n;
    }
    person[index] = 1;
    while (person[index] == 1)
        index = (index+1) % n;
    return winner (person, n, index, person_left - 1, n);
}

```

T.C  $\rightarrow O(n^2)$

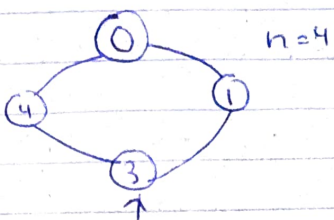
S.C  $\rightarrow O(n)$

$\Rightarrow$  Now, if we want to solve this question in  $O(n)$  T.C &  $O(1)$  S.C.

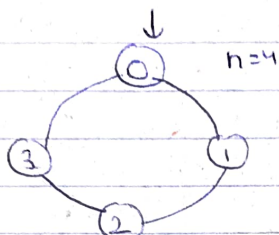




= Instead of



= Using above, we will use —



= We can make similarity b/w these two as —

$$0 \rightarrow 3$$

$$1 \rightarrow 4$$

$$2 \rightarrow 0$$

$$3 \rightarrow 1$$

=  $\text{winner}(n, k) = (\text{winner}(n-1, k) + k) \% n$

Code

```
int winner( int n, int k) {
    if(n==1)
        return 0;
    return (winner(n-1, k) + k) % n;
}
```