Day - 137

Stack - 5

* Celebrity Problem:

⇒

$$
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 \\
2 & 0 & 1 & 0 & 1 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 \\
4 & 1 & 0 & 1 & 1 & 0
\end{array}
$$

⇒ Here, we have n-people in the party & this matrix shows that if they know each other or not.

⇒ 1 → know each other.
0 → Don't know each other.

⇒ Here, we have to return that index that don't know ~~each~~ other people but all other people know that person.

⇒ For brute force approach, we will find a row that all have zero in it & then check in that corresponding index column for all 1 except that index.

⇒ If the condition passed, our answer is that index.

⇒ For optimized approach, that can solve problem in O(N) time.

⇒ So, what will ~~d~~ we do that we will take two numbers & ask them, they know each other or not.

⇒ So, if

First	Second
× Yes	No ✓
× Yes	Yes ×
× No	No ×
✓ No	Yes ×

Note: '✓' → may be celebrity
'×' → not a celebrity

⇒ So, we do this with two no. then select one number from that after that we ~~d~~ will do with all numbers.

⇒ In the last, no. that we will get, we check ~~it~~ one more time by checking if with all other numbers the above same question.

⇒ If condition passed that means that no. is celebrity.

Date Page

⇒ So, we will use stack for storing elements.

Code

```
stack<int>st;
for(i=n-1; i>=0; i--){
        st.push(i);
}
while( st.size() > 1 ){
      int first= st.top();
      st.pop();
      int second= st.top();
      st.pop;

      if( m[first][second] && !m[second][first])
                st.push (second);
      else if (!m[first][second] && m[second][first])
                st.push  (first);
}
if( st.empty())
      return -1;
int num = st.top();
int now=0 ; col=0;
for(i=0; i<n; i++){
      now += m[num][i];
      col  += m[i][num];
}
return  now==0 && col== n-1 ? num : -1;
```