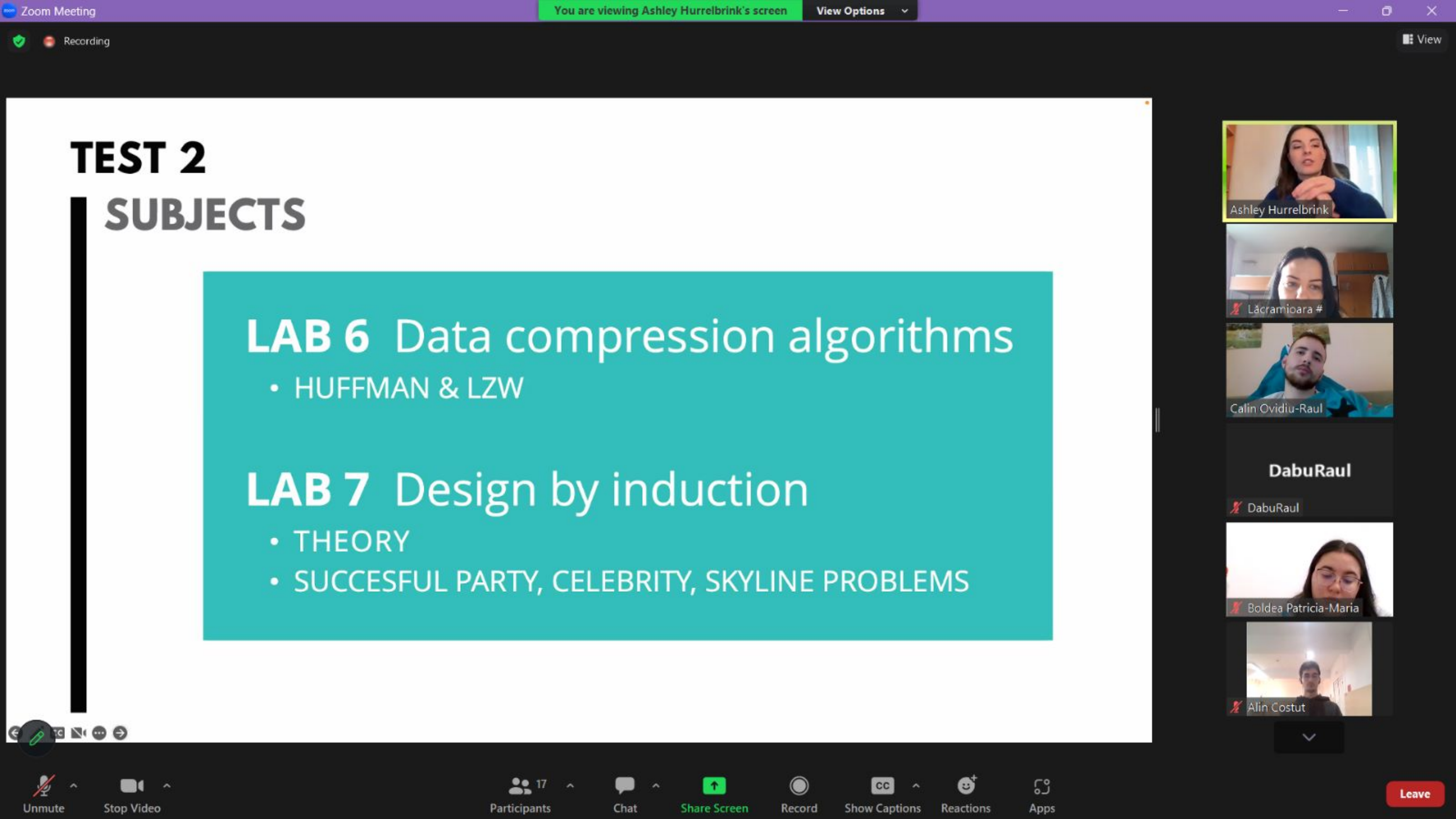


TEST 2

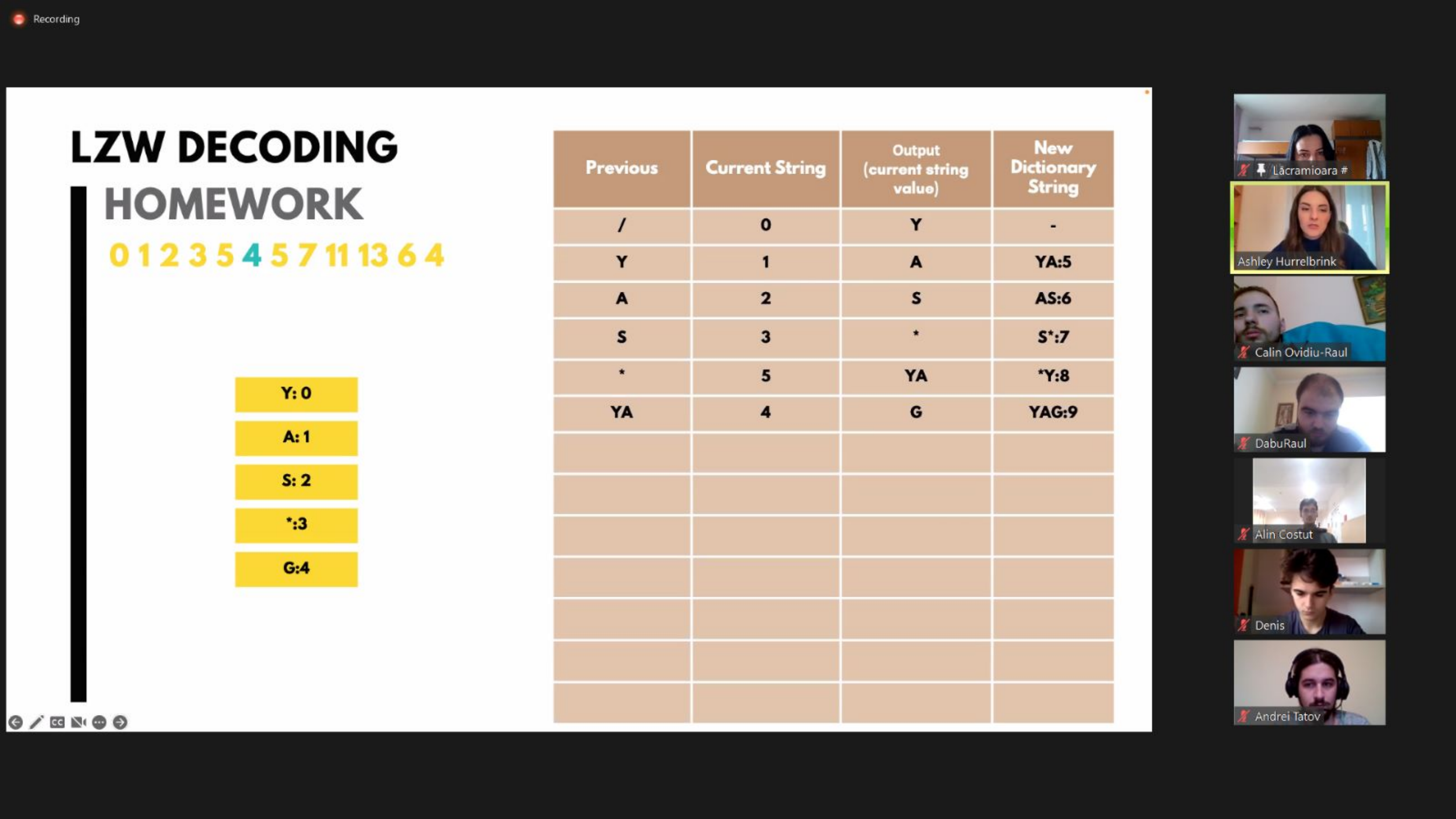
SUBJECTS

LAB 6 Data compression algorithms

- HUFFMAN & LZW

LAB 7 Design by induction

- THEORY
- SUCCESFUL PARTY, CELEBRITY, SKYLINE PROBLEMS



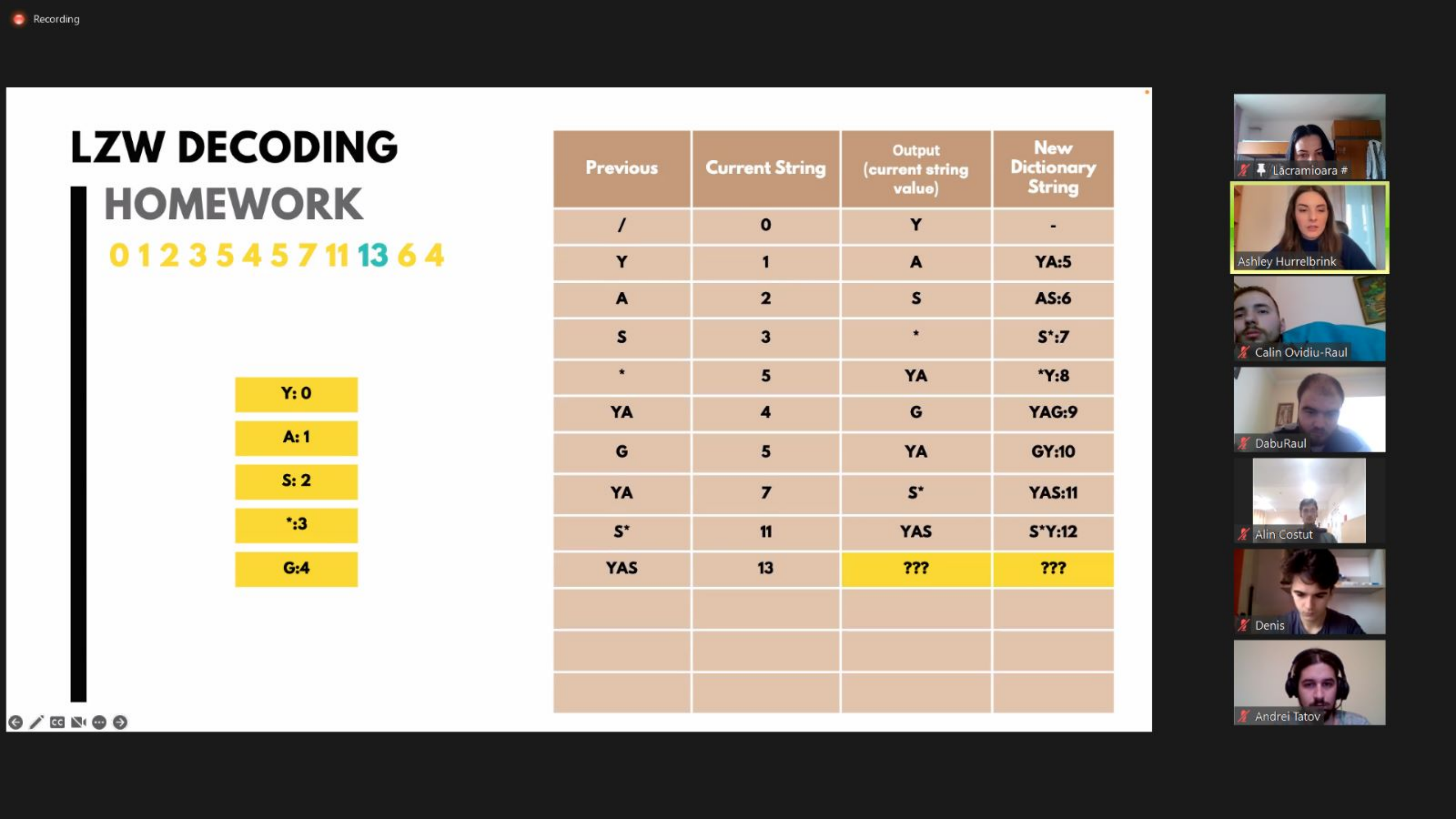
LZW DECODING

HOMEWORK

0 1 2 3 5 4 5 7 11 13 6 4

Y: 0
A: 1
S: 2
*:3
G:4

Previous	Current String	Output (current string value)	New Dictionary String
/	0	Y	-
Y	1	A	YA:5
A	2	S	AS:6
S	3	*	S*:7
*	5	YA	*Y:8
YA	4	G	YAG:9



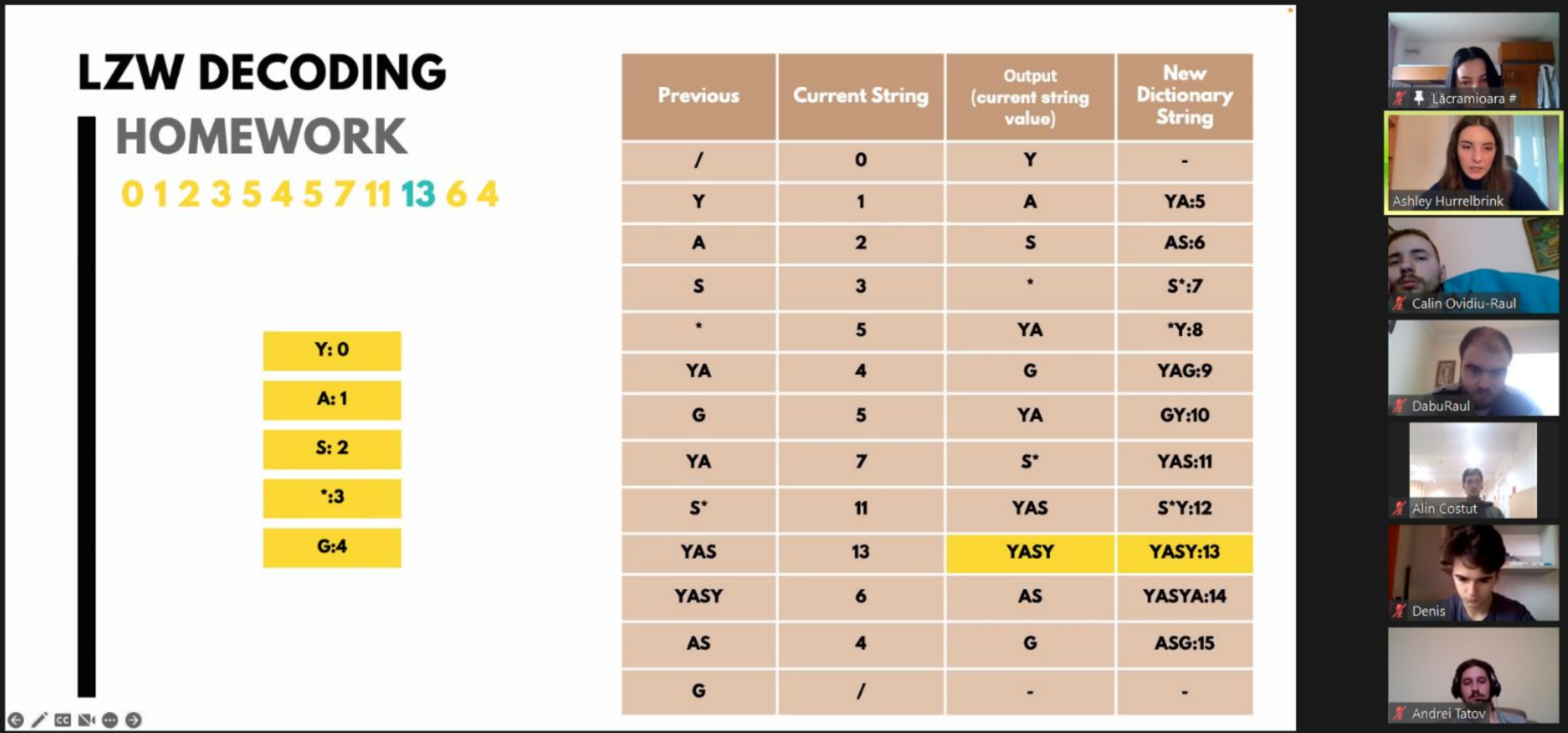
LZW DECODING

HOMEWORK

0 1 2 3 5 4 5 7 11 13 6 4

Y: 0
A: 1
S: 2
*:3
G:4

Previous	Current String	Output (current string value)	New Dictionary String
/	0	Y	-
Y	1	A	YA:5
A	2	S	AS:6
S	3	*	S*:7
*	5	YA	*Y:8
YA	4	G	YAG:9
G	5	YA	GY:10
YA	7	S*	YAS:11
S*	11	YAS	S*Y:12
YAS	13	???	???



LZW DECODING

HOMEWORK

0 1 2 3 5 4 5 7 11 13 6 4

Y: 0

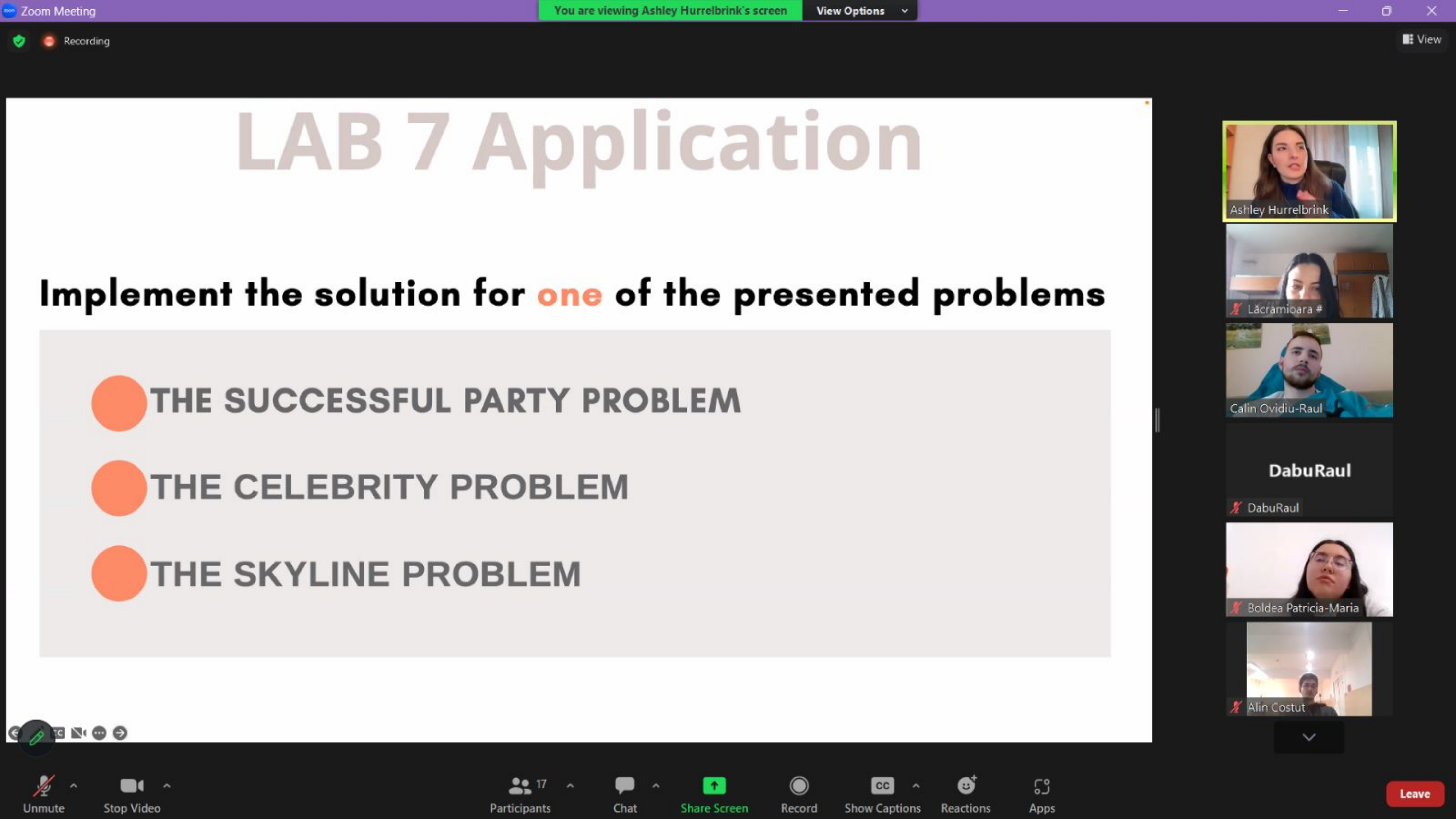
A: 1

S: 2

*:3

G:4

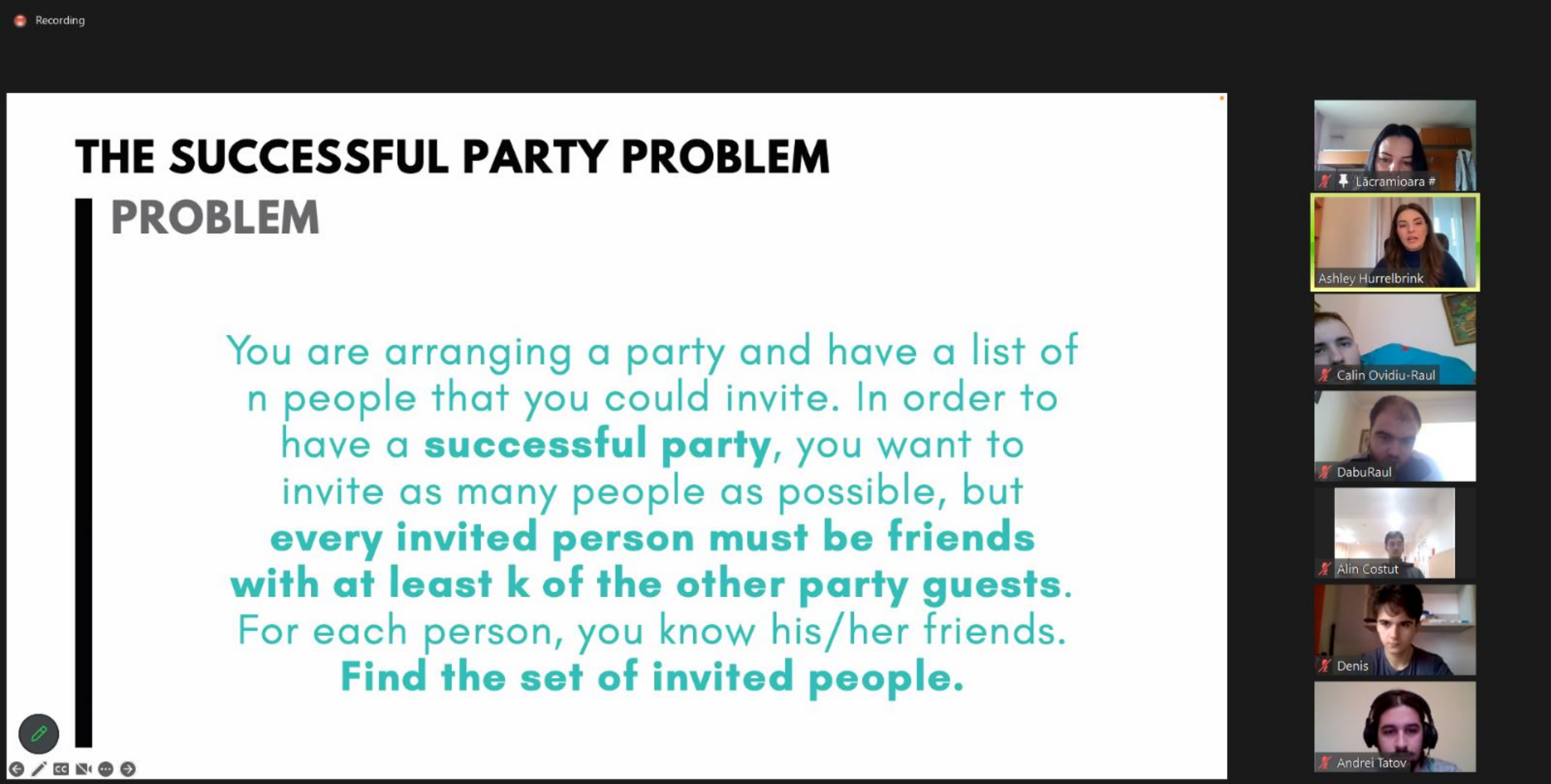
Previous	Current String	Output (current string value)	New Dictionary String
/	0	Y	-
Y	1	A	YA:5
A	2	S	AS:6
S	3	*	S*:7
*	5	YA	*Y:8
YA	4	G	YAG:9
G	5	YA	GY:10
YA	7	S*	YAS:11
S*	11	YAS	S*Y:12
YAS	13	YASY	YASY:13
YASY	6	AS	YASYA:14
AS	4	G	ASG:15
G	/	-	-



LAB 7 Application

Implement the solution for one of the presented problems

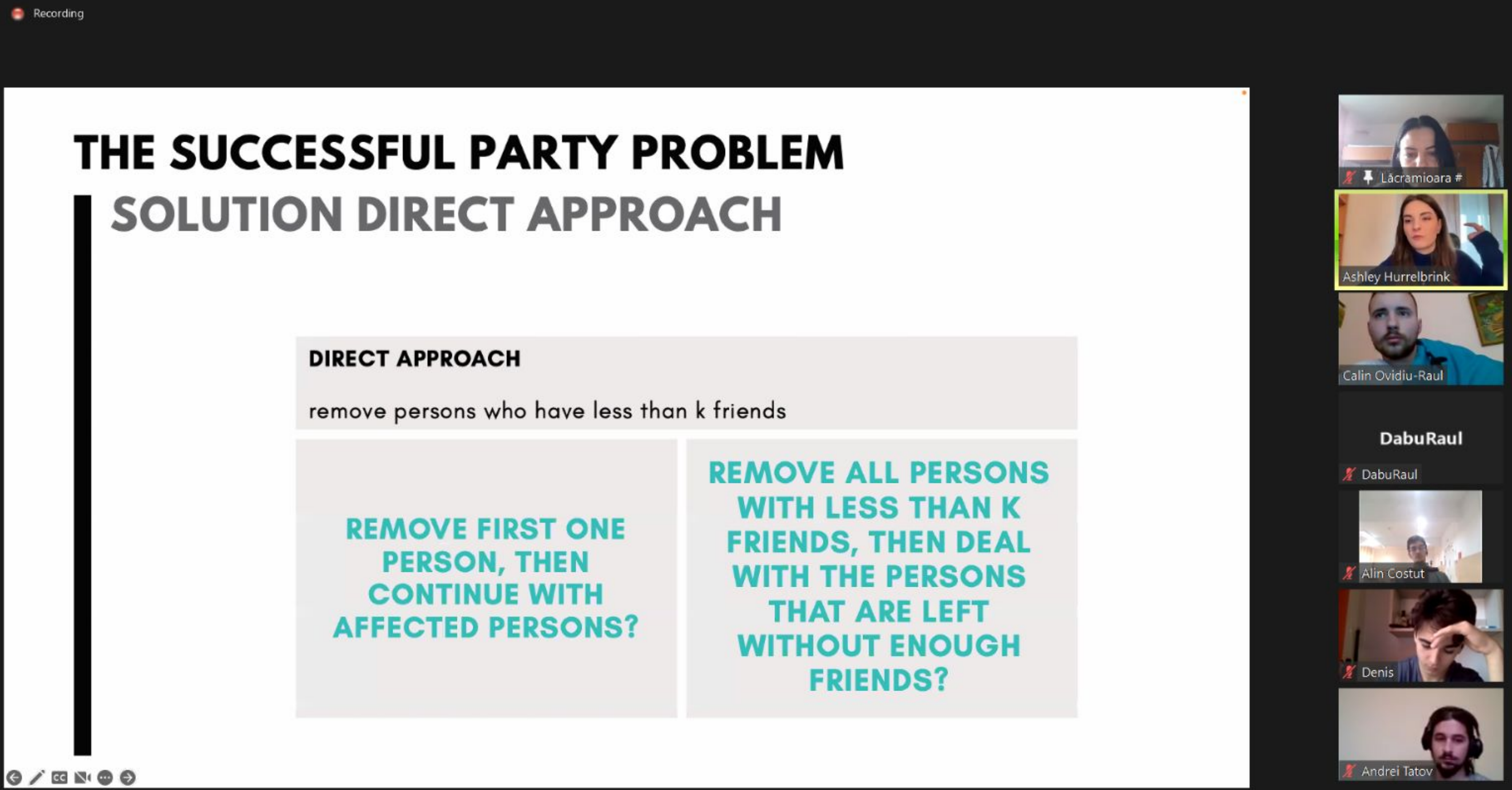
- THE SUCCESSFUL PARTY PROBLEM
- THE CELEBRITY PROBLEM
- THE SKYLINE PROBLEM



THE SUCCESSFUL PARTY PROBLEM

PROBLEM

You are arranging a party and have a list of n people that you could invite. In order to have a **successful party**, you want to invite as many people as possible, but **every invited person must be friends with at least k of the other party guests**. For each person, you know his/her friends. **Find the set of invited people.**



THE SUCCESSFUL PARTY PROBLEM

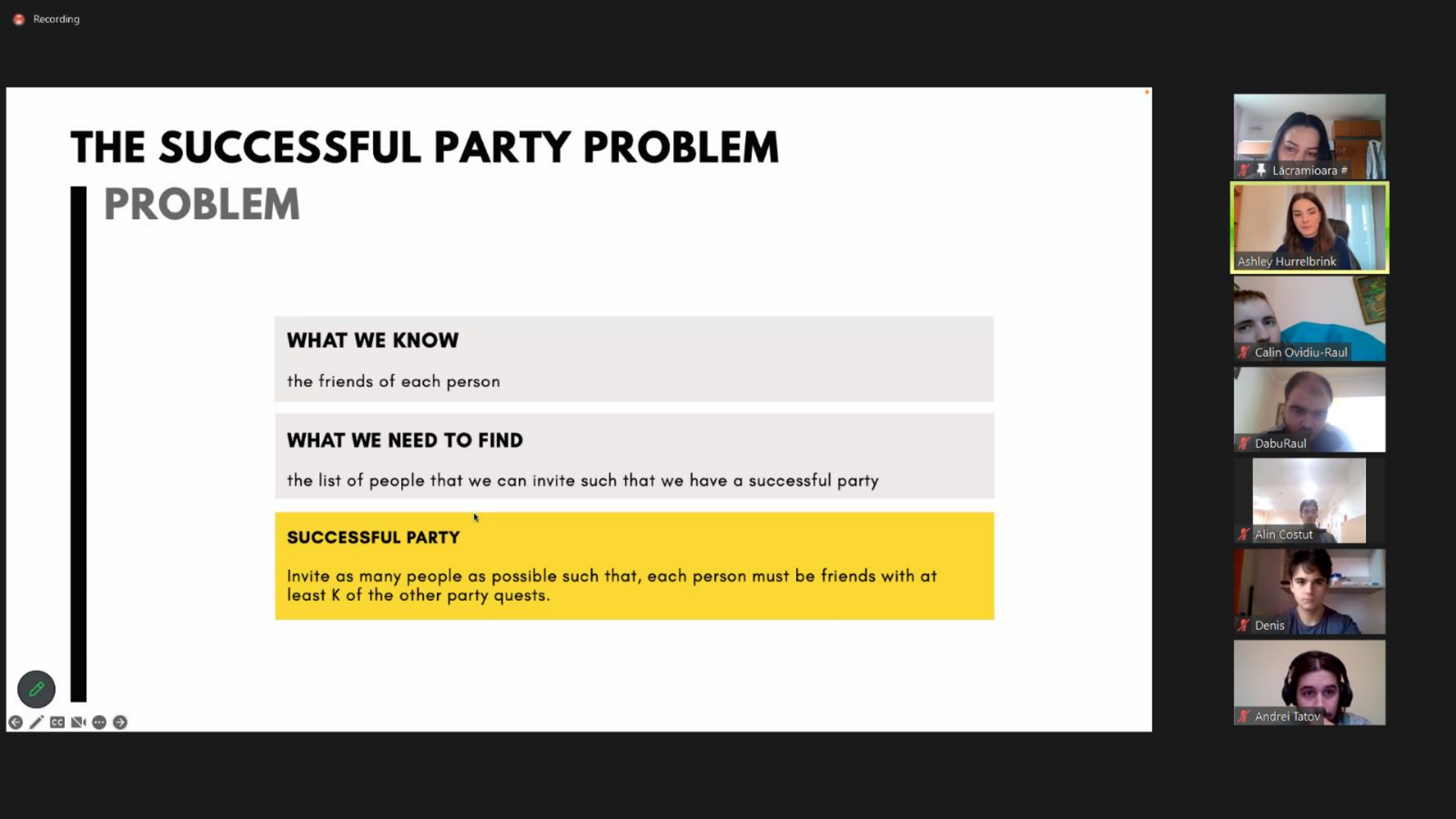
SOLUTION DIRECT APPROACH

DIRECT APPROACH

remove persons who have less than k friends

REMOVE FIRST ONE PERSON, THEN CONTINUE WITH AFFECTED PERSONS?

REMOVE ALL PERSONS WITH LESS THAN K FRIENDS, THEN DEAL WITH THE PERSONS THAT ARE LEFT WITHOUT ENOUGH FRIENDS?



THE SUCCESSFUL PARTY PROBLEM

PROBLEM

WHAT WE KNOW

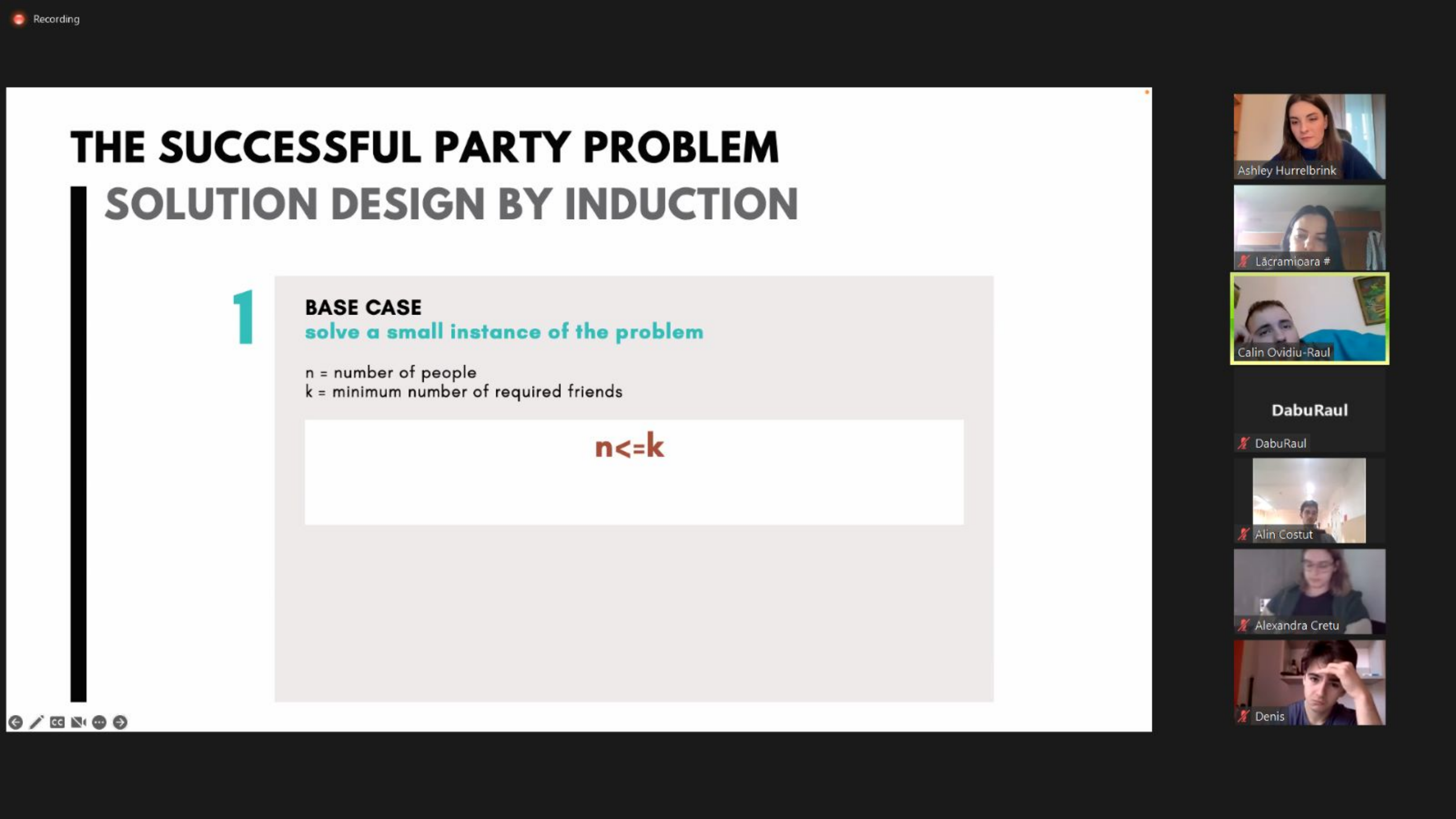
the friends of each person

WHAT WE NEED TO FIND

the list of people that we can invite such that we have a successful party

SUCCESSFUL PARTY

Invite as many people as possible such that, each person must be friends with at least K of the other party quests.



THE SUCCESSFUL PARTY PROBLEM

SOLUTION DESIGN BY INDUCTION

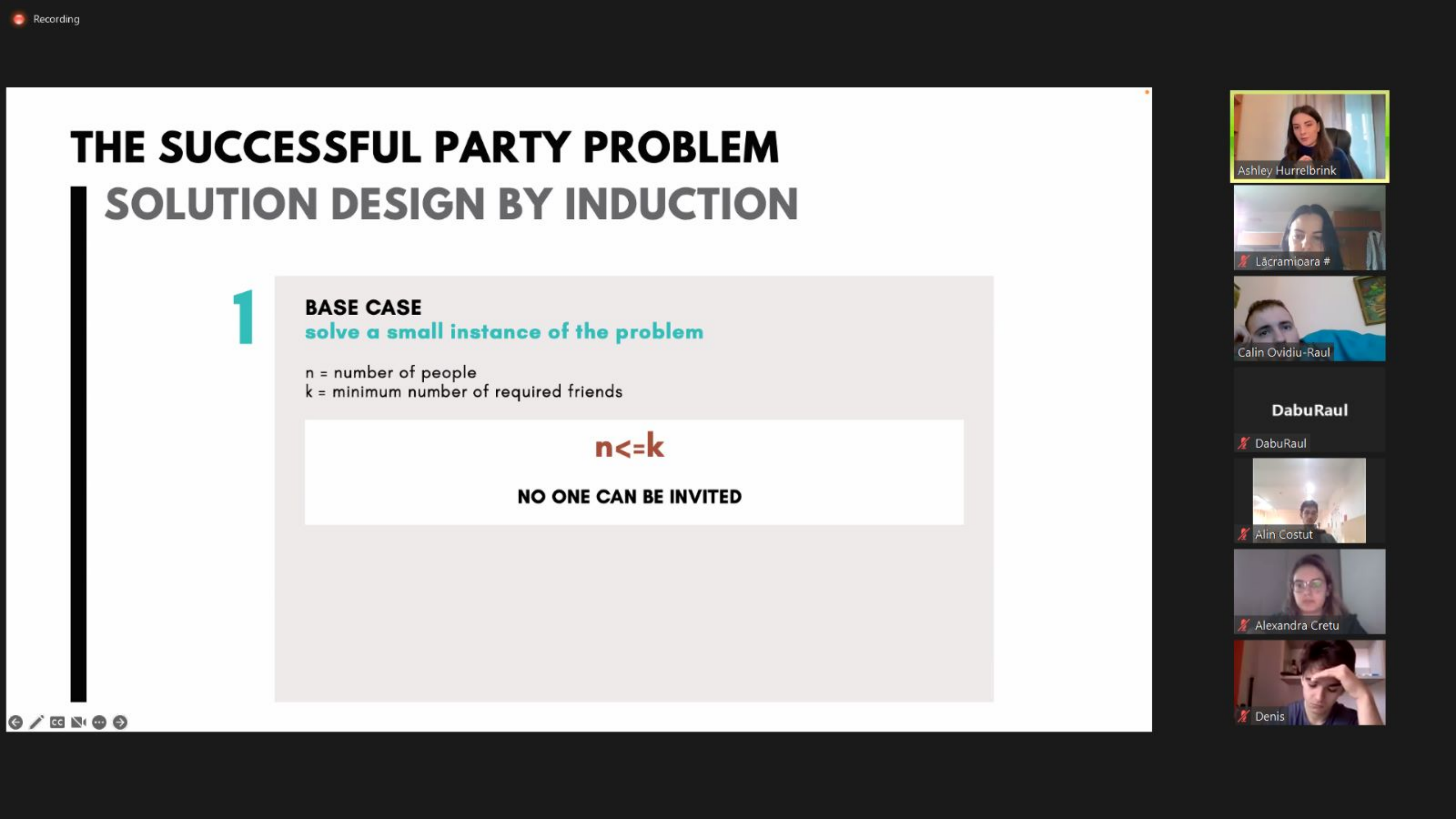
1

BASE CASE

solve a small instance of the problem

n = number of people
k = minimum number of required friends

n<=k



THE SUCCESSFUL PARTY PROBLEM

SOLUTION DESIGN BY INDUCTION

1

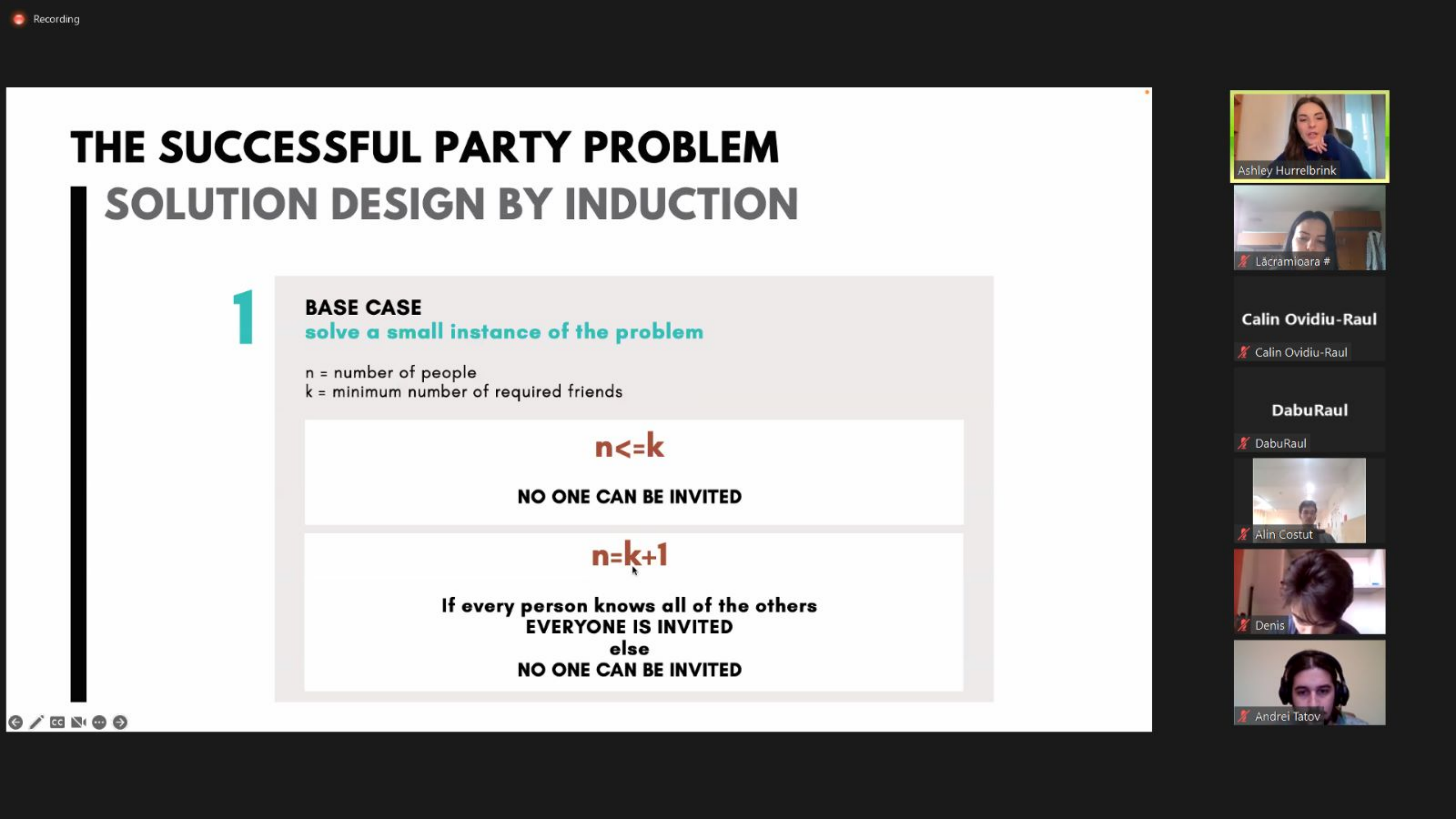
BASE CASE

solve a small instance of the problem

n = number of people
k = minimum number of required friends

n<=k

NO ONE CAN BE INVITED



THE SUCCESSFUL PARTY PROBLEM

SOLUTION DESIGN BY INDUCTION

1

BASE CASE

solve a small instance of the problem

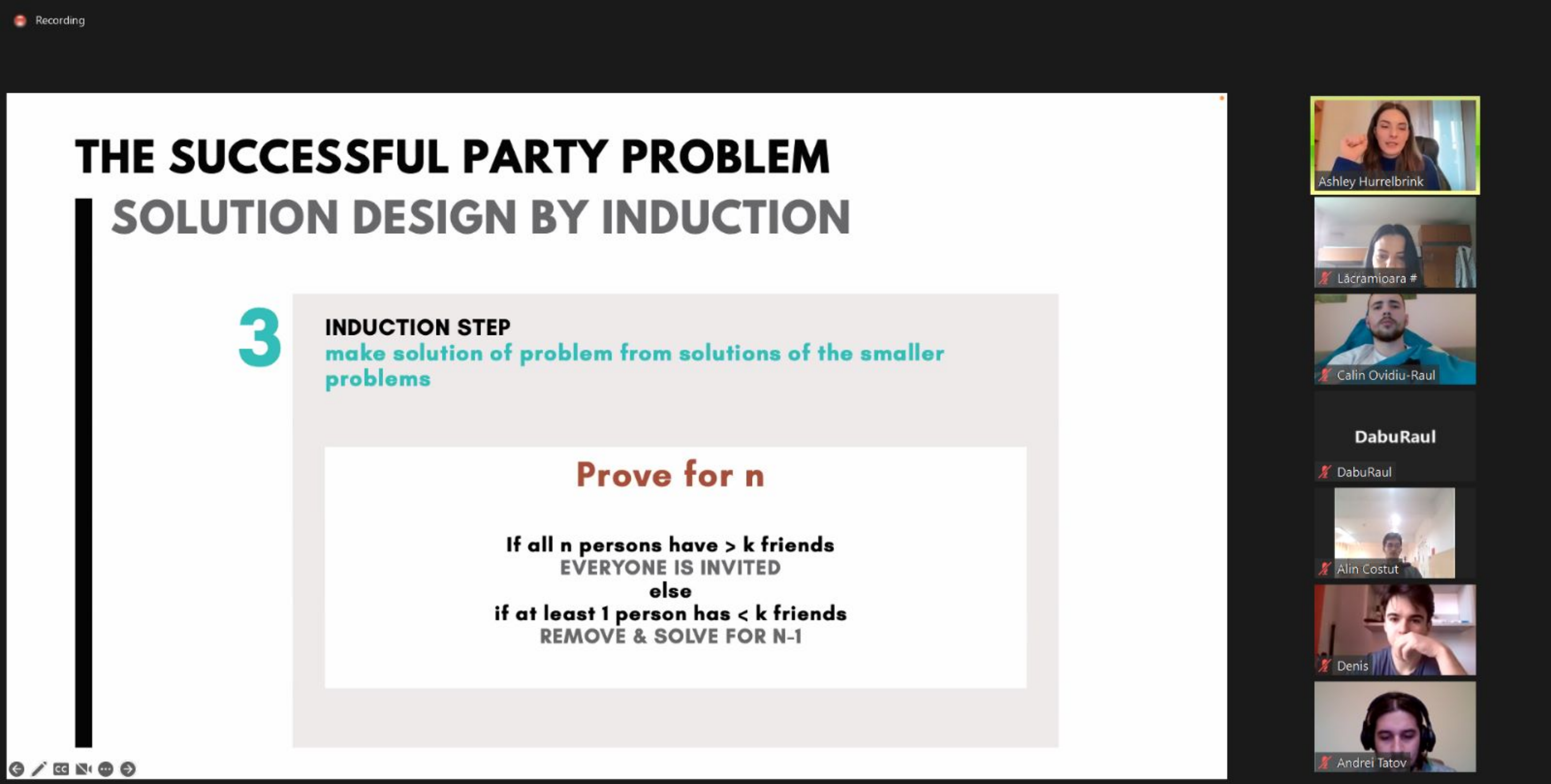
n = number of people
k = minimum number of required friends

n<=k

NO ONE CAN BE INVITED

n=k+1

If every person knows all of the others
EVERYONE IS INVITED
else
NO ONE CAN BE INVITED



THE SUCCESSFUL PARTY PROBLEM

SOLUTION DESIGN BY INDUCTION

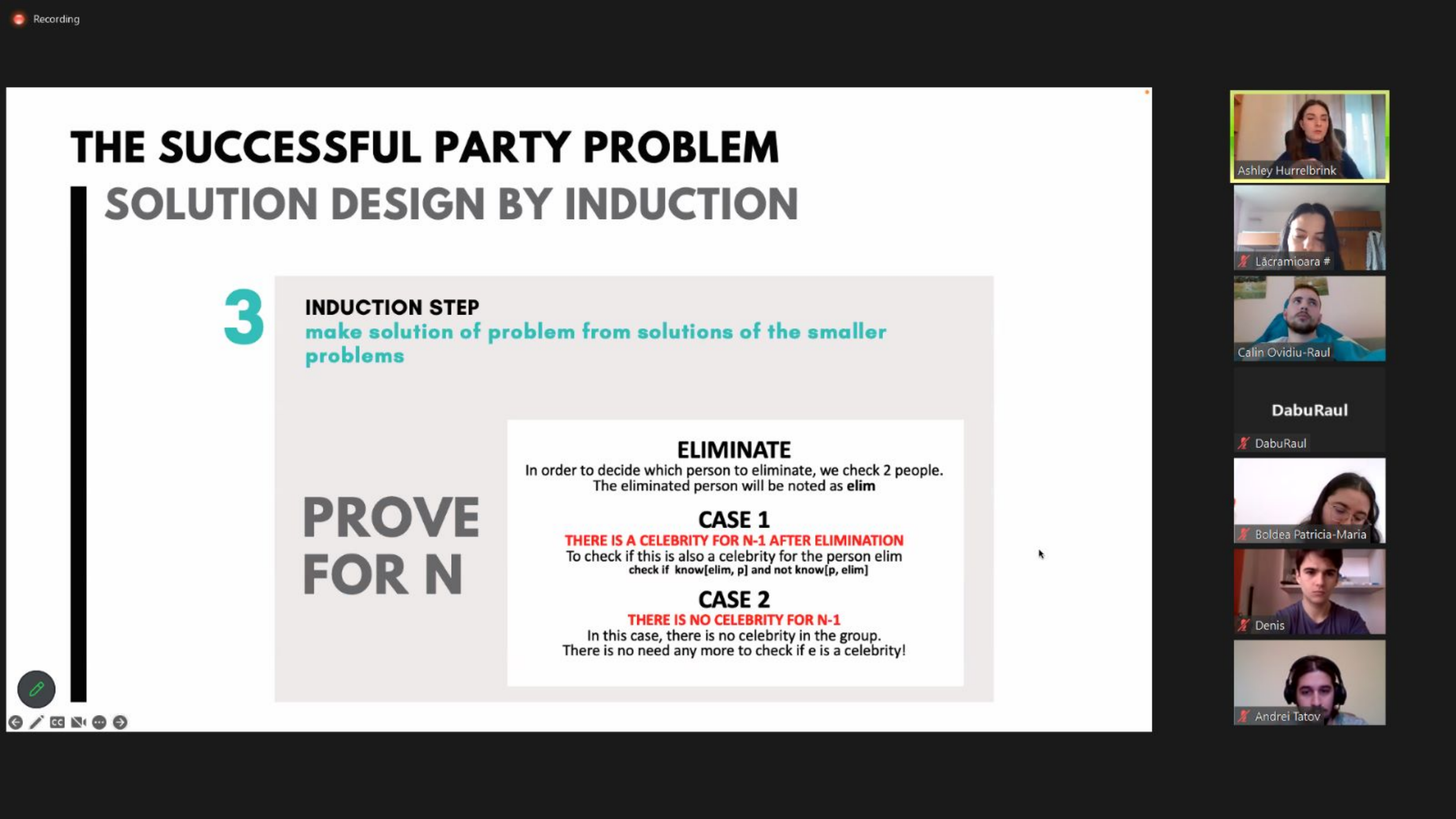
3

INDUCTION STEP

make solution of problem from solutions of the smaller problems

Prove for n

If all n persons have > k friends
EVERYONE IS INVITED
else
if at least 1 person has < k friends
REMOVE & SOLVE FOR N-1



THE SUCCESSFUL PARTY PROBLEM

SOLUTION DESIGN BY INDUCTION

3

INDUCTION STEP

make solution of problem from solutions of the smaller problems

PROVE FOR N

ELIMINATE

In order to decide which person to eliminate, we check 2 people.
The eliminated person will be noted as **elim**

CASE 1

THERE IS A CELEBRITY FOR N-1 AFTER ELIMINATION

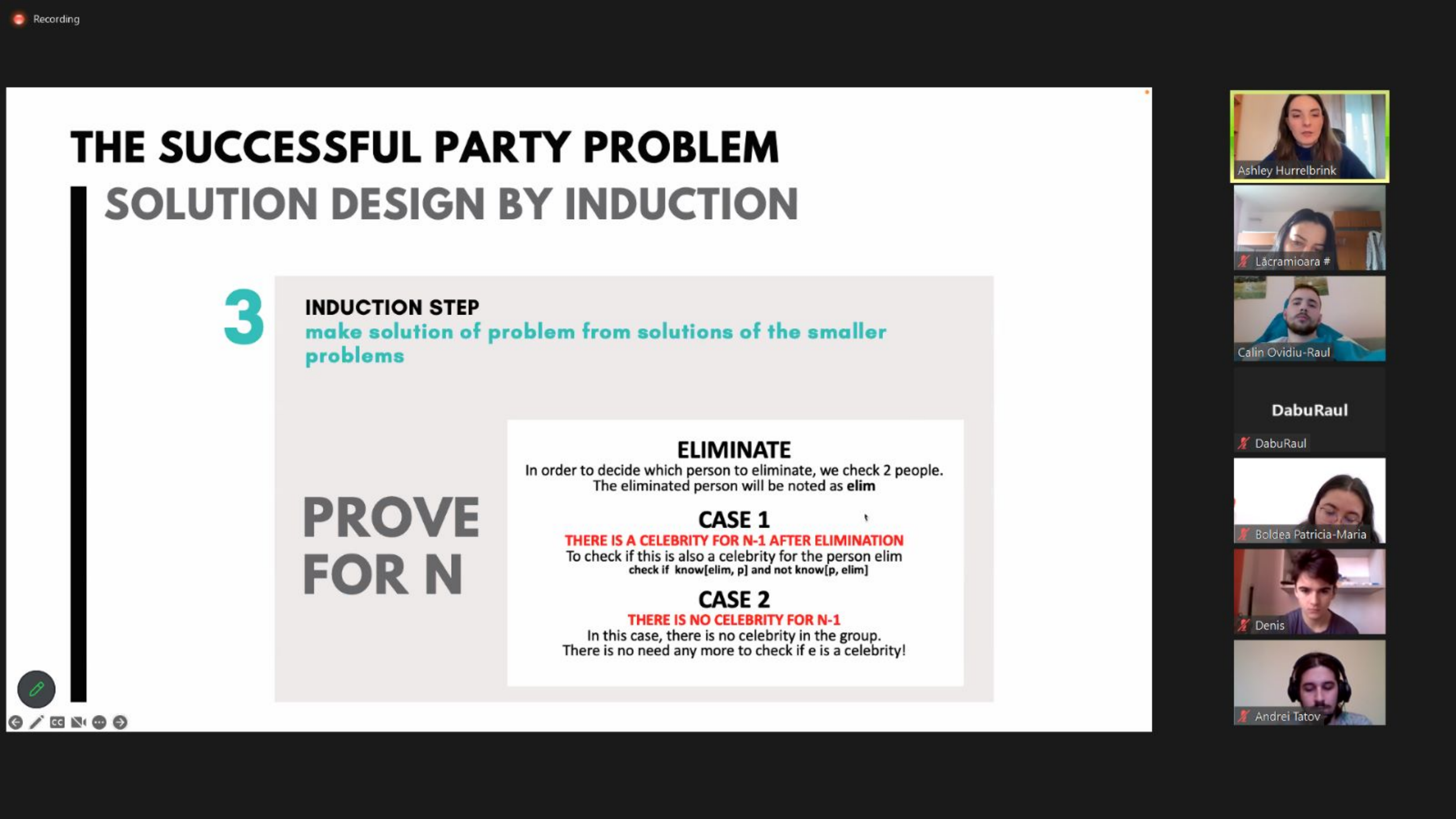
To check if this is also a celebrity for the person elim

check if know[elim, p] and not know[p, elim]

CASE 2

THERE IS NO CELEBRITY FOR N-1

In this case, there is no celebrity in the group.
There is no need any more to check if e is a celebrity!



THE SUCCESSFUL PARTY PROBLEM

SOLUTION DESIGN BY INDUCTION

3 **INDUCTION STEP**

make solution of problem from solutions of the smaller problems

PROVE FOR N

ELIMINATE

In order to decide which person to eliminate, we check 2 people.
The eliminated person will be noted as **elim**

CASE 1

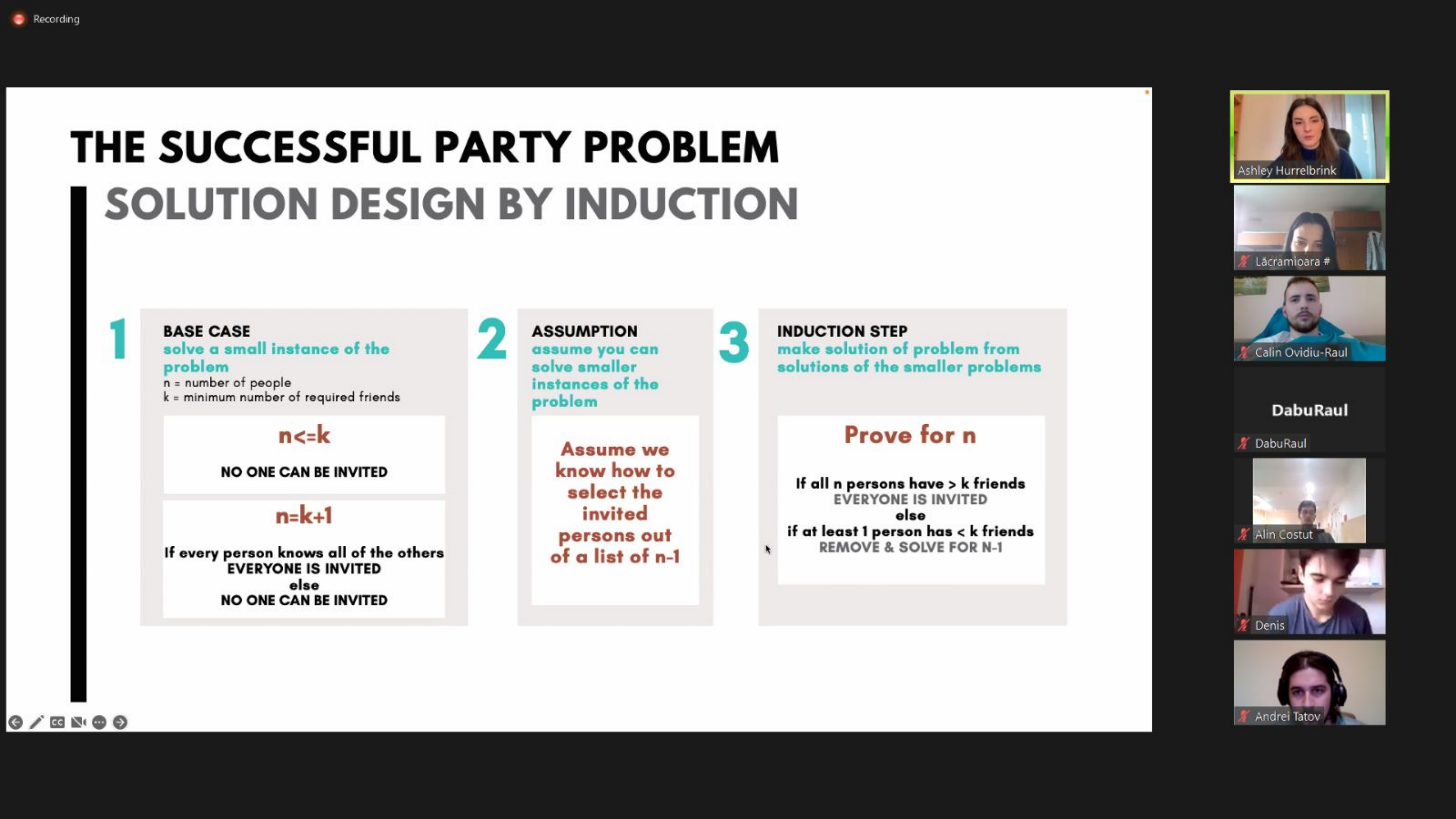
THERE IS A CELEBRITY FOR N-1 AFTER ELIMINATION

To check if this is also a celebrity for the person elim
check if know[elim, p] and not know[p, elim]

CASE 2

THERE IS NO CELEBRITY FOR N-1

In this case, there is no celebrity in the group.
There is no need any more to check if e is a celebrity!



THE SUCCESSFUL PARTY PROBLEM

SOLUTION DESIGN BY INDUCTION

1

BASE CASE

solve a small instance of the problem

n = number of people
k = minimum number of required friends

n<=k

NO ONE CAN BE INVITED

n=k+1

If every person knows all of the others
EVERYONE IS INVITED
else
NO ONE CAN BE INVITED

2

ASSUMPTION

assume you can solve smaller instances of the problem

Assume we know how to select the invited persons out of a list of n-1

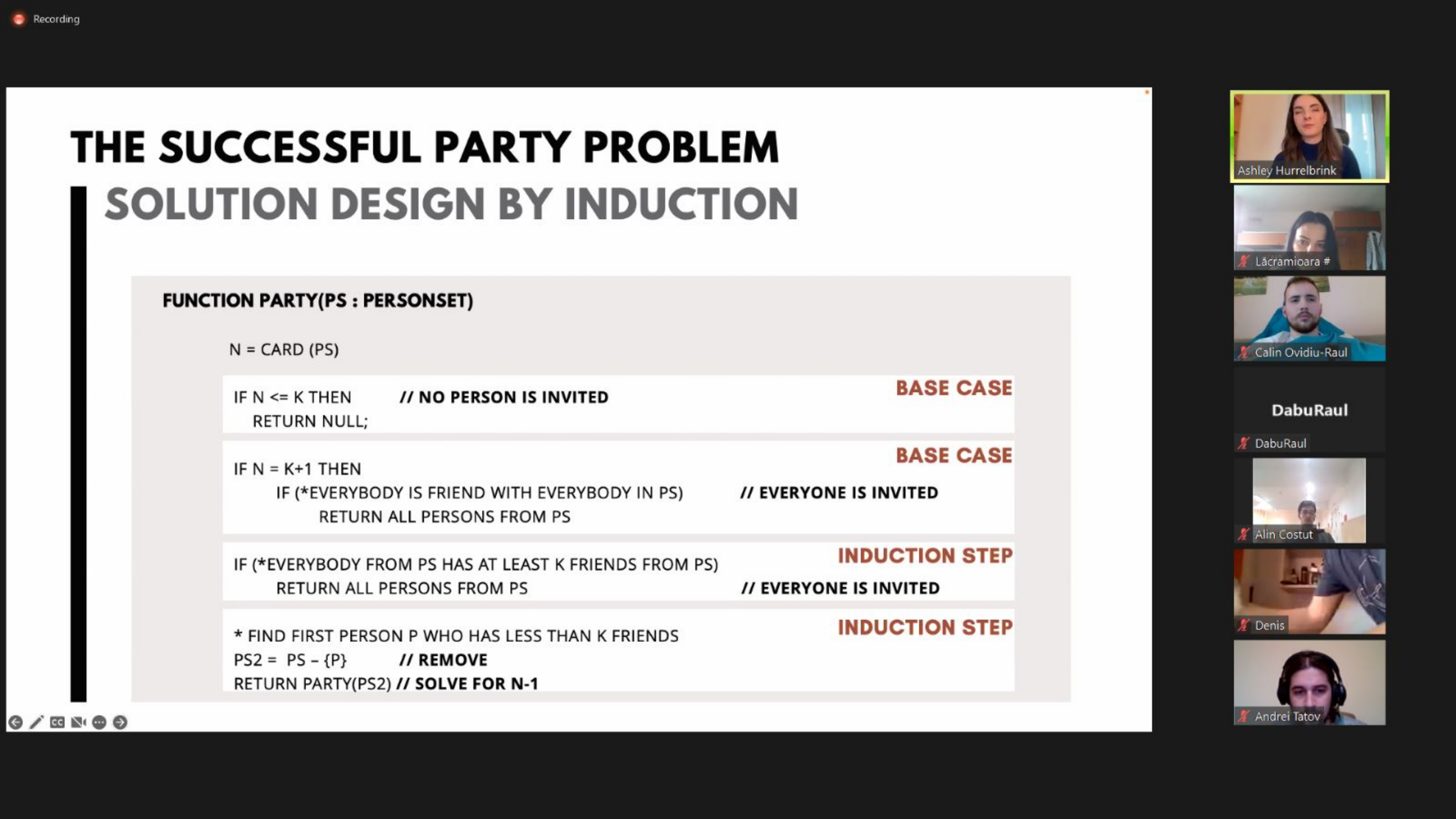
3

INDUCTION STEP

make solution of problem from solutions of the smaller problems

Prove for n

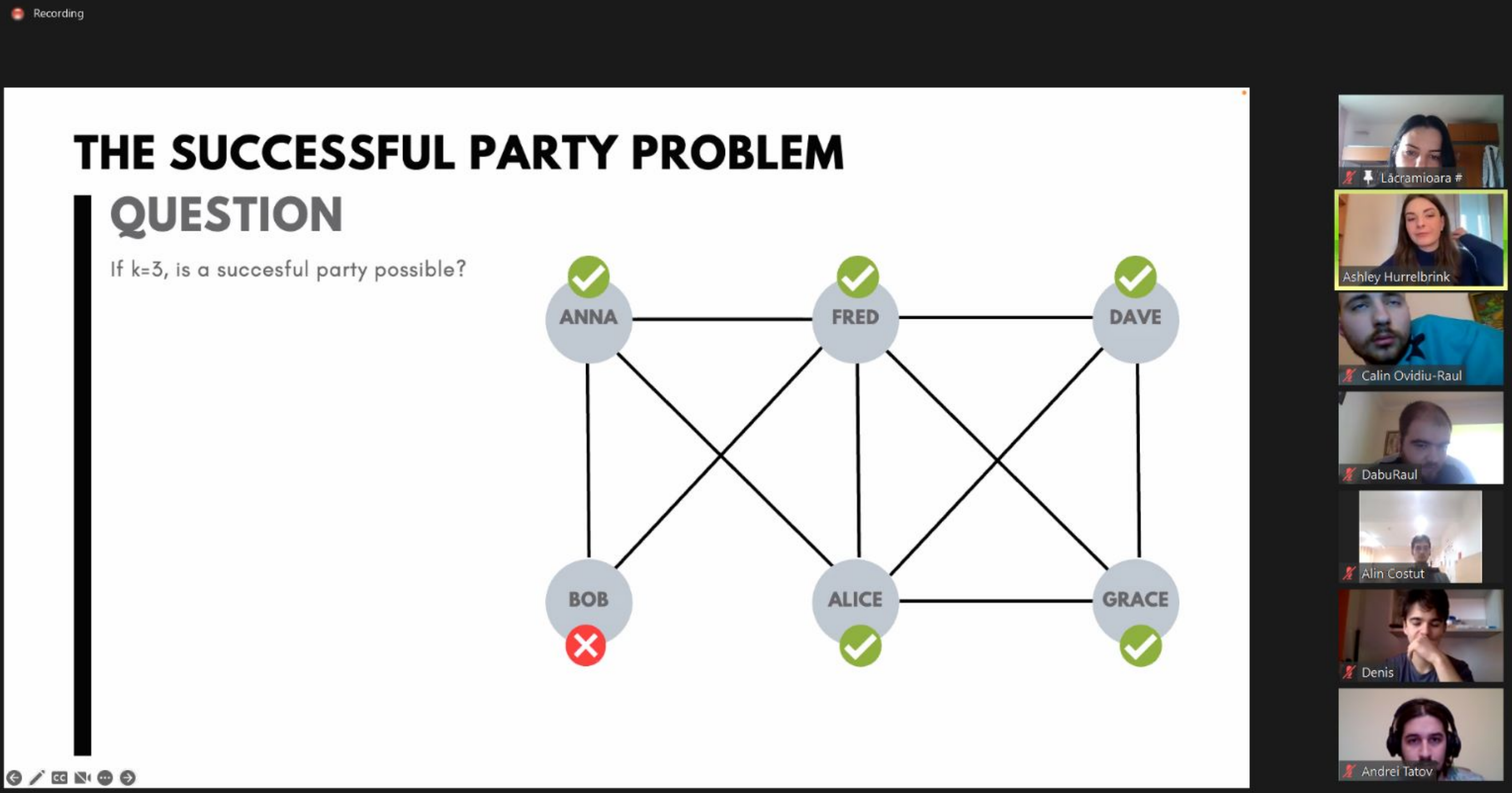
If all n persons have > k friends
EVERYONE IS INVITED
else
if at least 1 person has < k friends
REMOVE & SOLVE FOR N-1



THE SUCCESSFUL PARTY PROBLEM

SOLUTION DESIGN BY INDUCTION

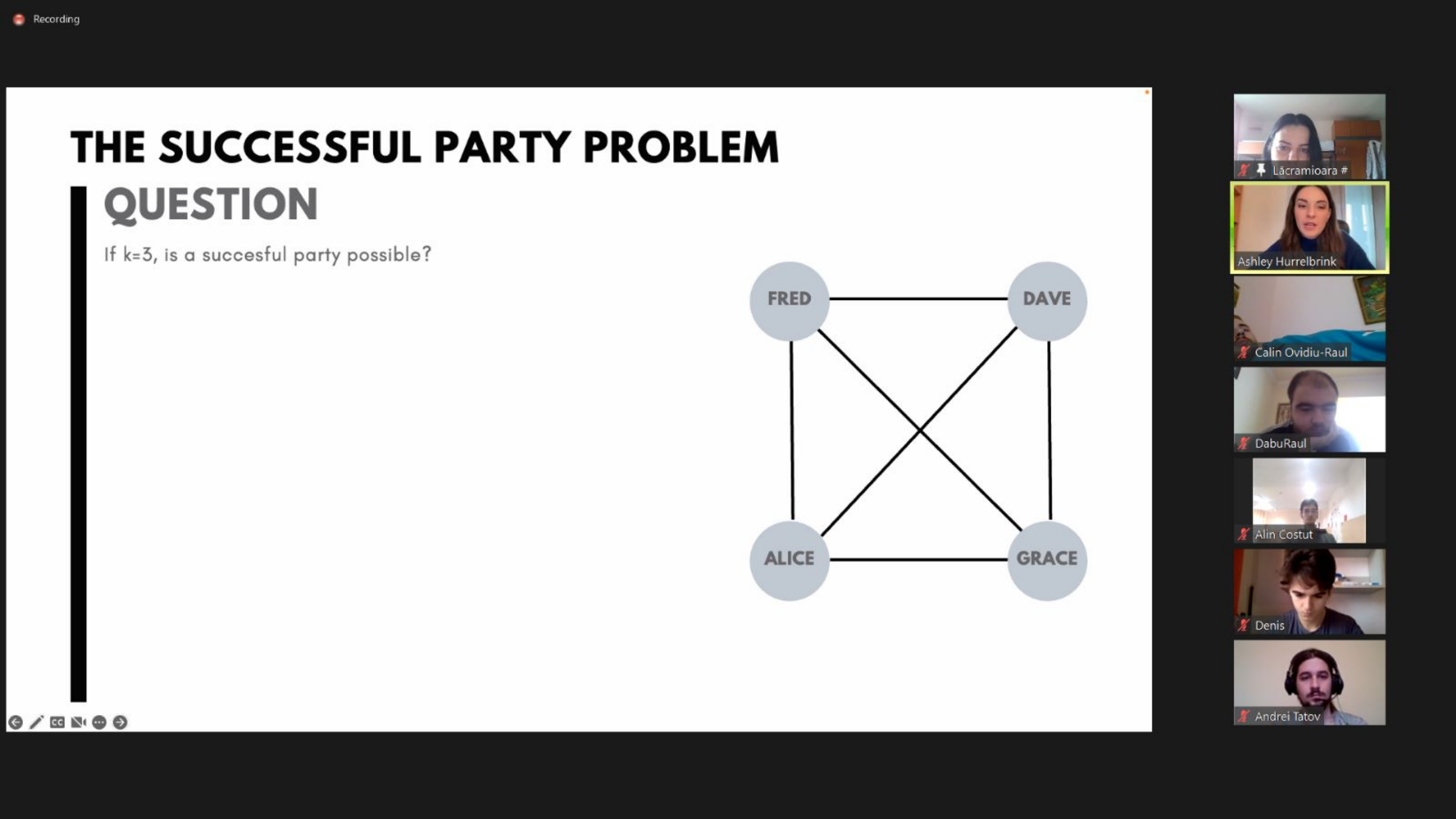
```
FUNCTION PARTY(PS : PERSONSET)

        N = CARD (PS)

    IF N <= K THEN          // NO PERSON IS INVITED                              BASE CASE
       RETURN NULL;

    IF N = K+1 THEN                                                             BASE CASE
           IF (*EVERYBODY IS FRIEND WITH EVERYBODY IN PS)       // EVERYONE IS INVITED
                RETURN ALL PERSONS FROM PS

    IF (*EVERYBODY FROM PS HAS AT LEAST K FRIENDS FROM PS)                 INDUCTION STEP
           RETURN ALL PERSONS FROM PS                           // EVERYONE IS INVITED

    * FIND FIRST PERSON P WHO HAS LESS THAN K FRIENDS                      INDUCTION STEP
    PS2 =  PS – {P}         // REMOVE
    RETURN PARTY(PS2) // SOLVE FOR N-1
```



THE SUCCESSFUL PARTY PROBLEM

QUESTION

If k=3, is a succesful party possible?

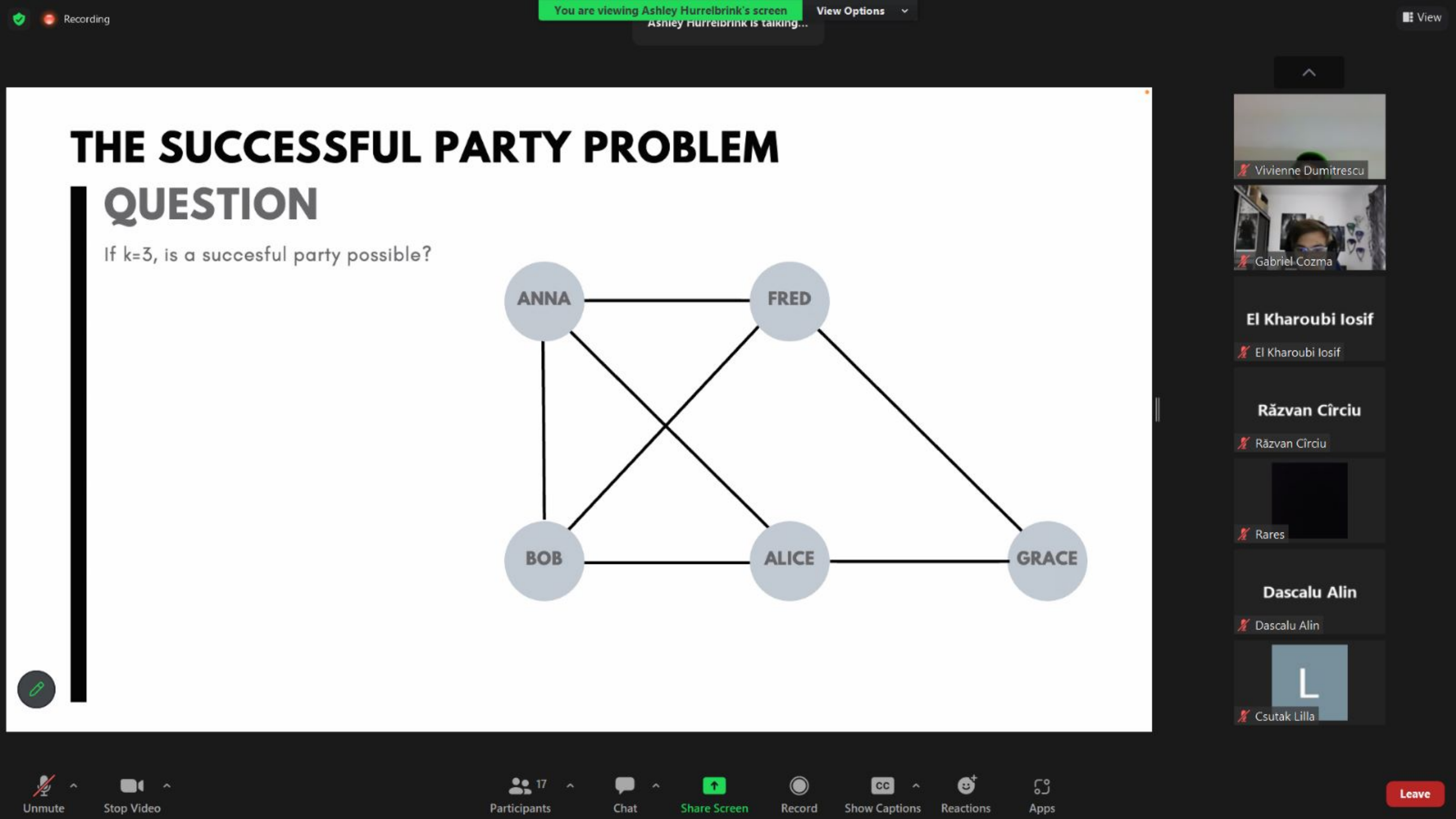


Recording

THE SUCCESSFUL PARTY PROBLEM

QUESTION

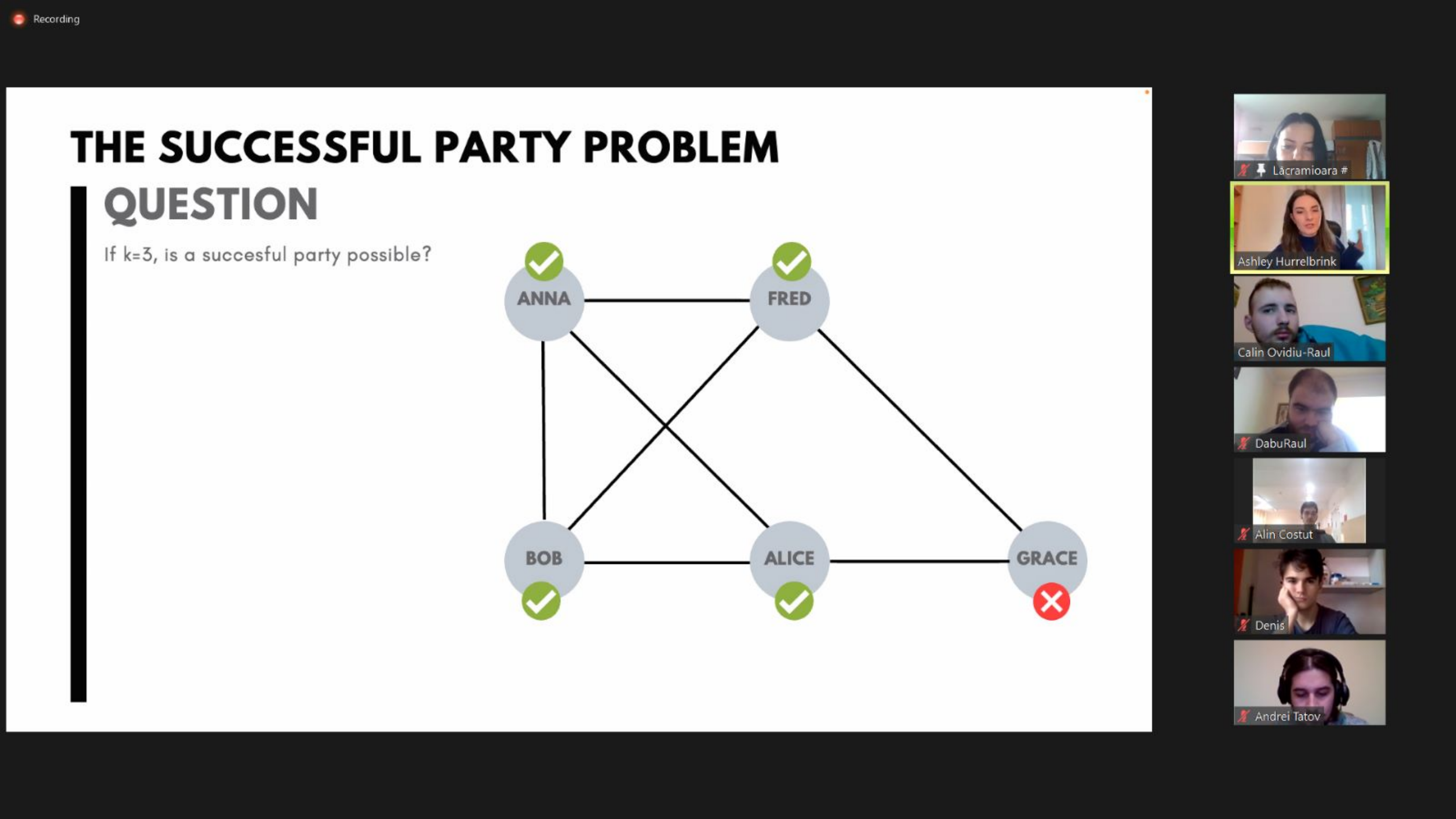
If k=3, is a succesful party possible?



THE SUCCESSFUL PARTY PROBLEM

QUESTION

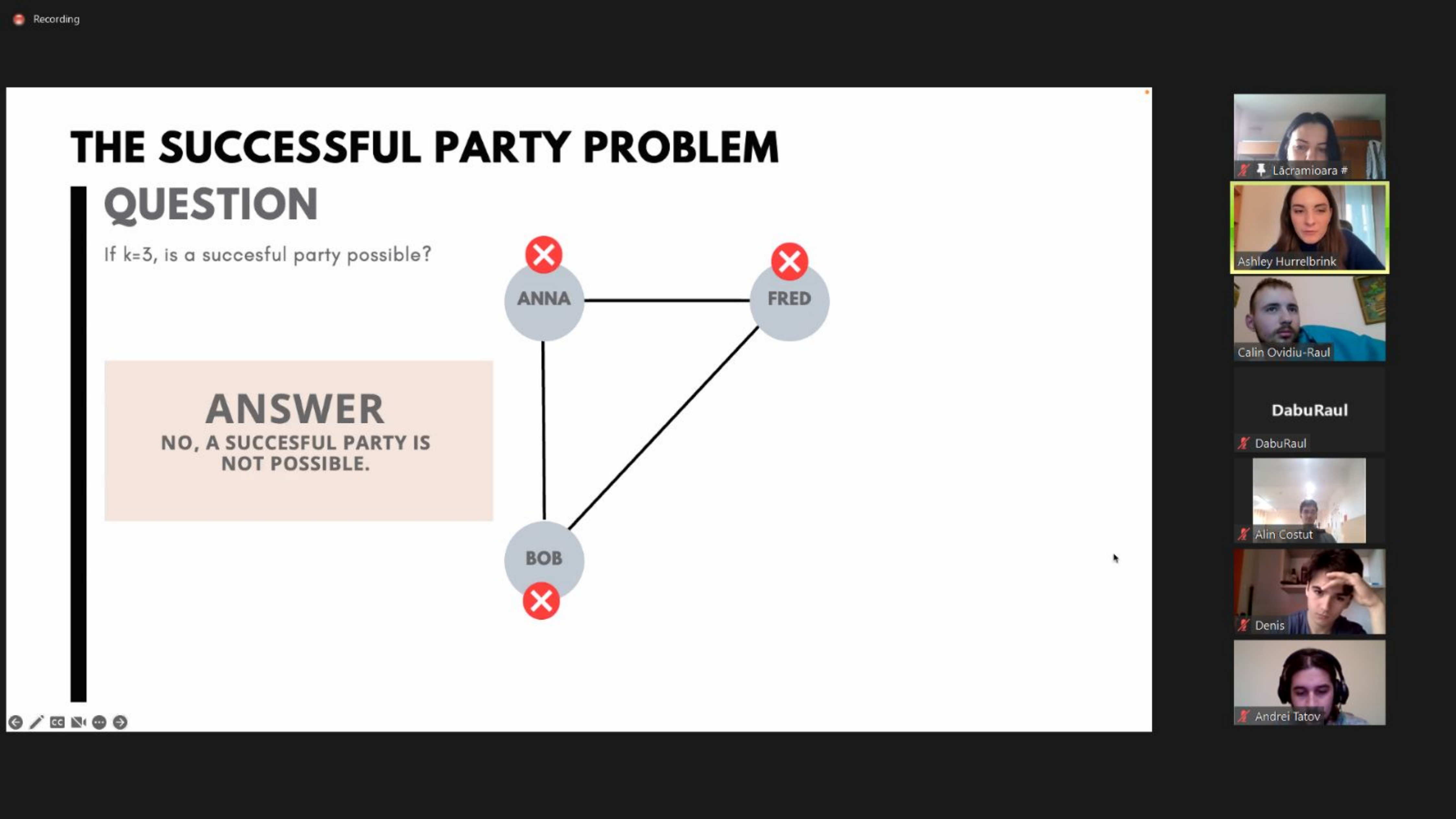
If k=3, is a succesful party possible?



THE SUCCESSFUL PARTY PROBLEM

QUESTION

If k=3, is a succesful party possible?



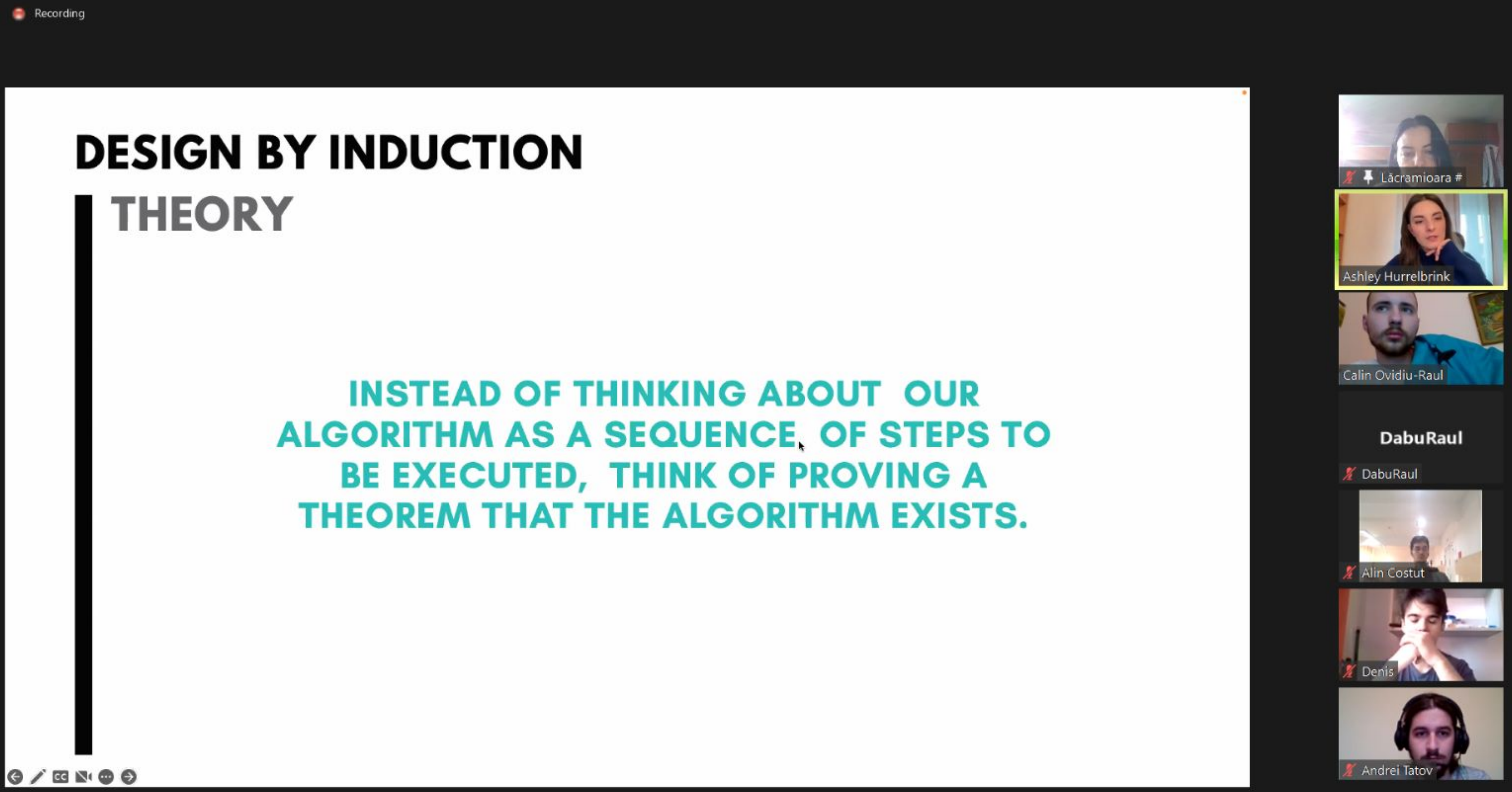
THE SUCCESSFUL PARTY PROBLEM

QUESTION

If k=3, is a succesful party possible?

ANSWER

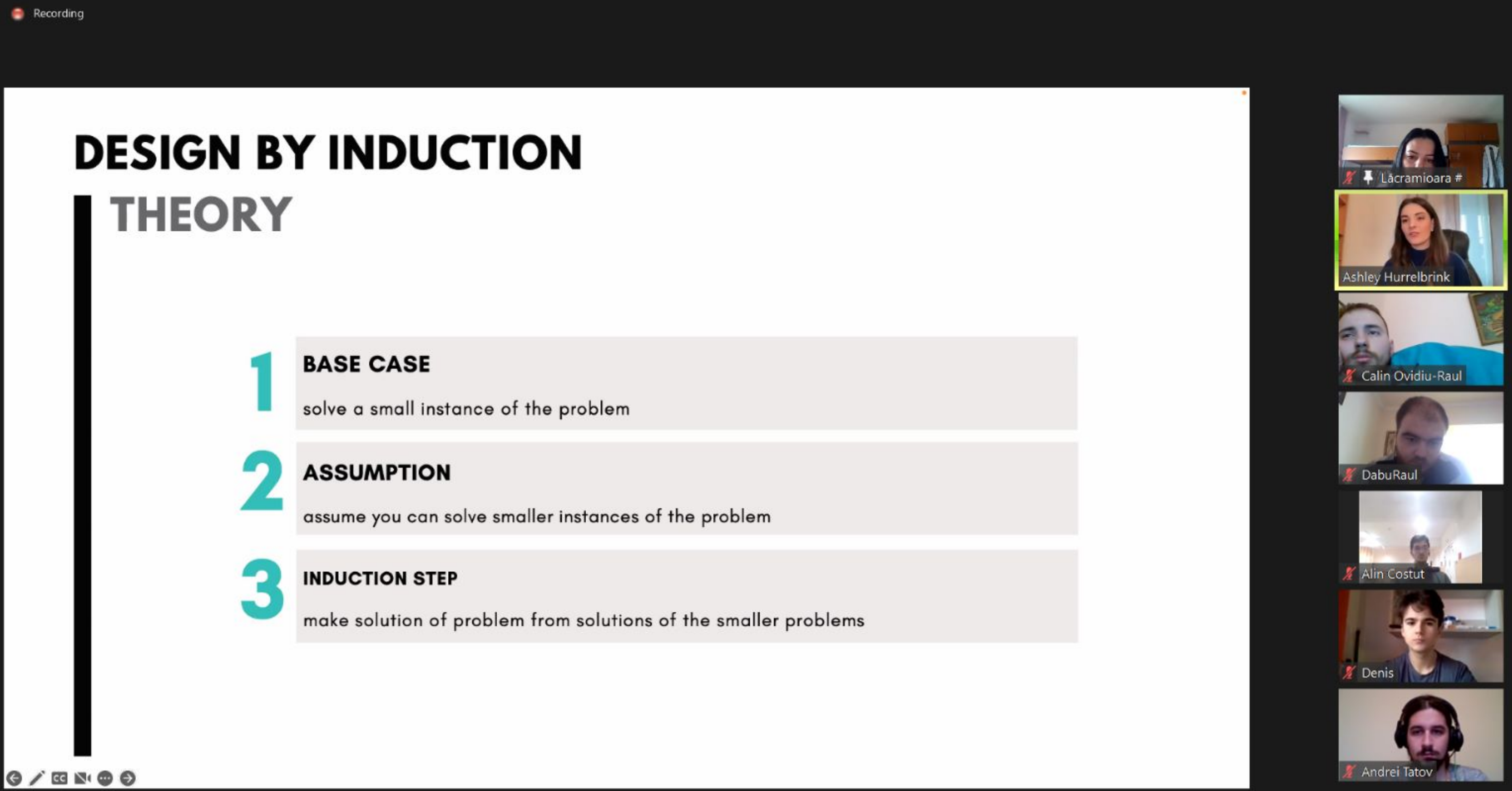
NO, A SUCCESFUL PARTY IS NOT POSSIBLE.



DESIGN BY INDUCTION

THEORY

INSTEAD OF THINKING ABOUT OUR ALGORITHM AS A SEQUENCE OF STEPS TO BE EXECUTED, THINK OF PROVING A THEOREM THAT THE ALGORITHM EXISTS.



DESIGN BY INDUCTION

THEORY

1 **BASE CASE**

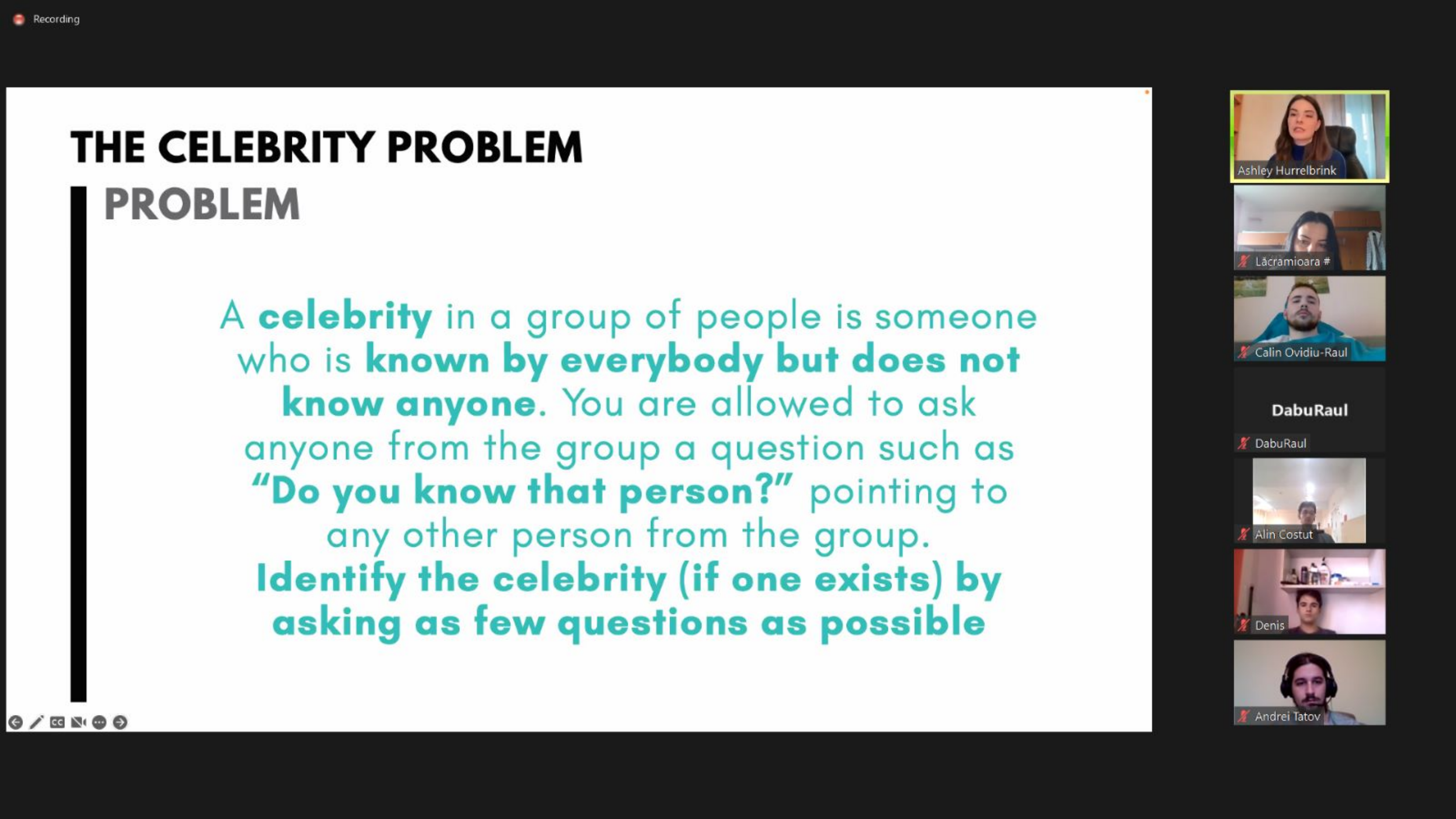
solve a small instance of the problem

2 **ASSUMPTION**

assume you can solve smaller instances of the problem

3 **INDUCTION STEP**

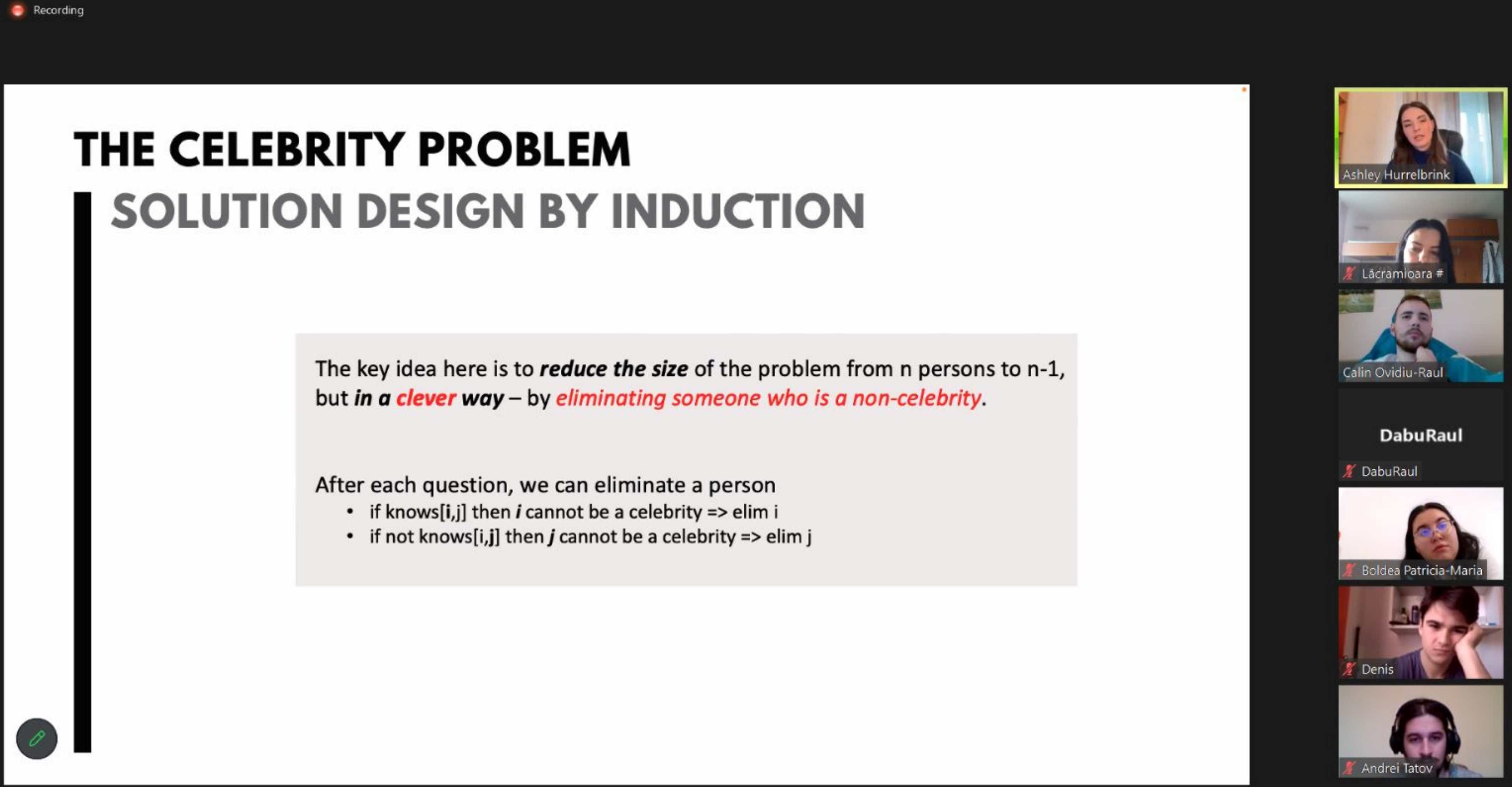
make solution of problem from solutions of the smaller problems



THE CELEBRITY PROBLEM

PROBLEM

A **celebrity** in a group of people is someone who is **known by everybody but does not know anyone**. You are allowed to ask anyone from the group a question such as **"Do you know that person?"** pointing to any other person from the group. **Identify the celebrity (if one exists) by asking as few questions as possible**



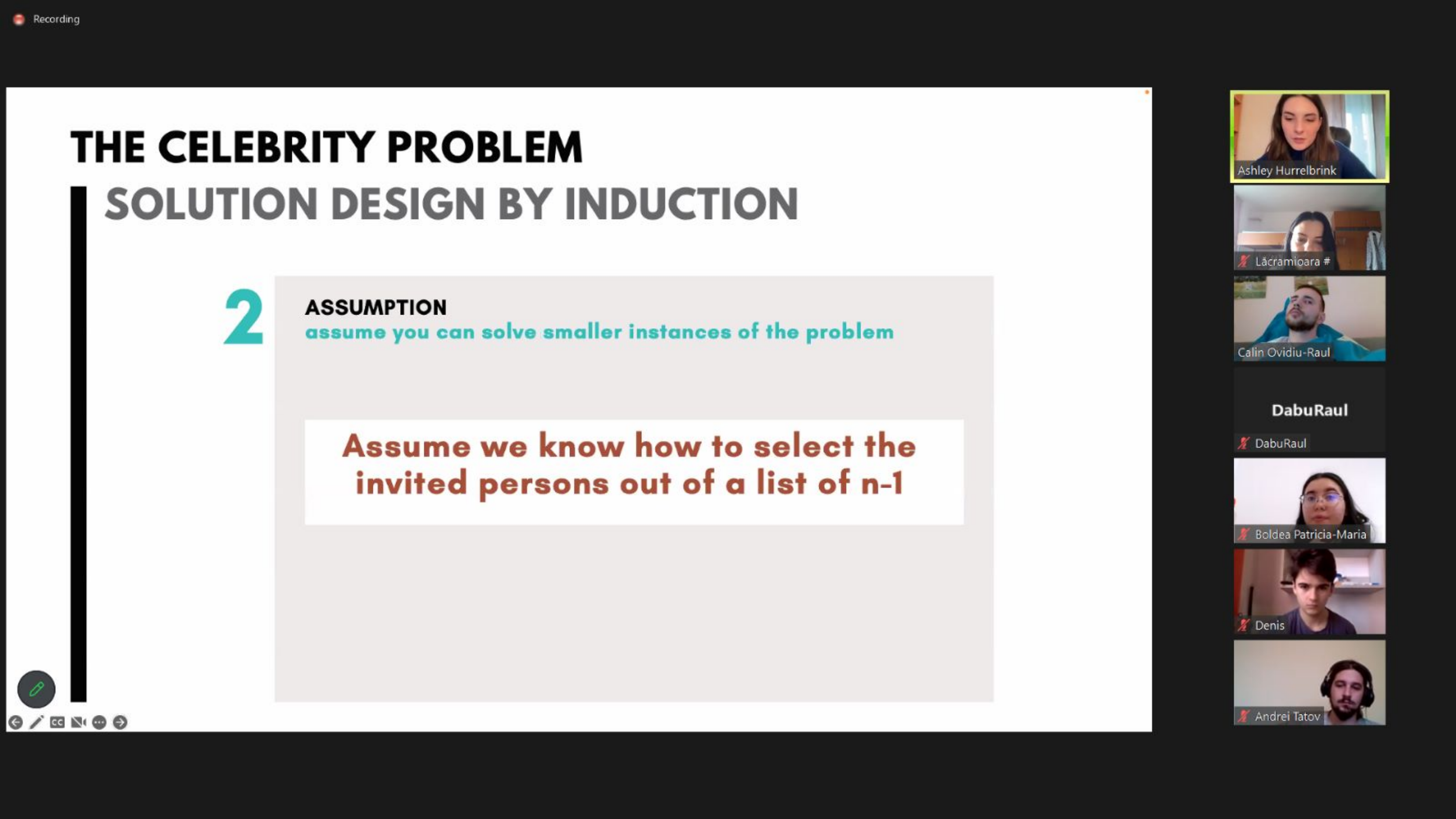
THE CELEBRITY PROBLEM

SOLUTION DESIGN BY INDUCTION

The key idea here is to ***reduce the size*** of the problem from n persons to n-1, but ***in a clever way*** – by *eliminating someone who is a non-celebrity*.

After each question, we can eliminate a person

- if knows[**i**,j] then ***i*** cannot be a celebrity => elim i
- if not knows[i,**j**] then ***j*** cannot be a celebrity => elim j



THE CELEBRITY PROBLEM

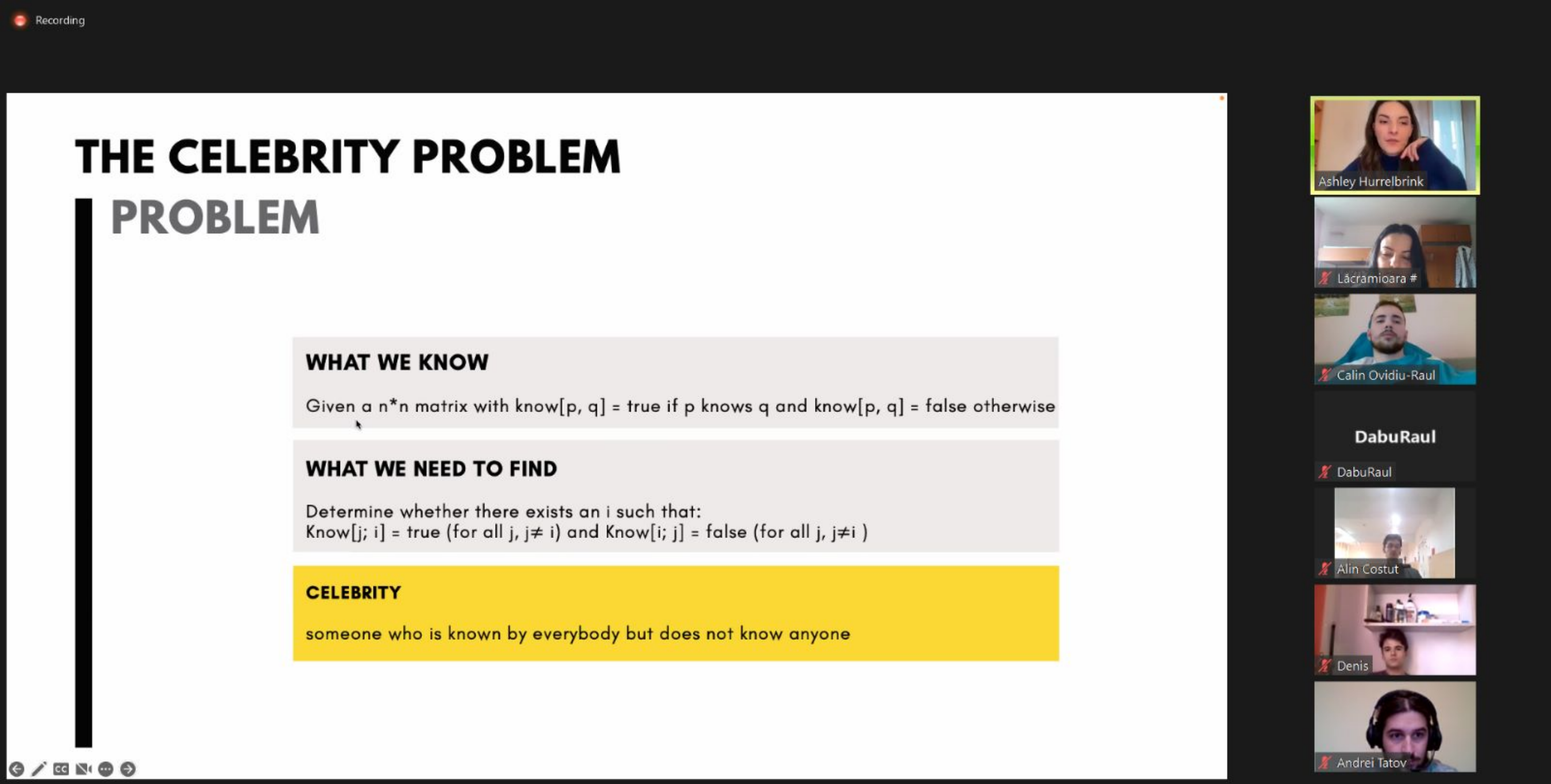
SOLUTION DESIGN BY INDUCTION

2

ASSUMPTION

assume you can solve smaller instances of the problem

Assume we know how to select the invited persons out of a list of n-1



THE CELEBRITY PROBLEM

PROBLEM

WHAT WE KNOW

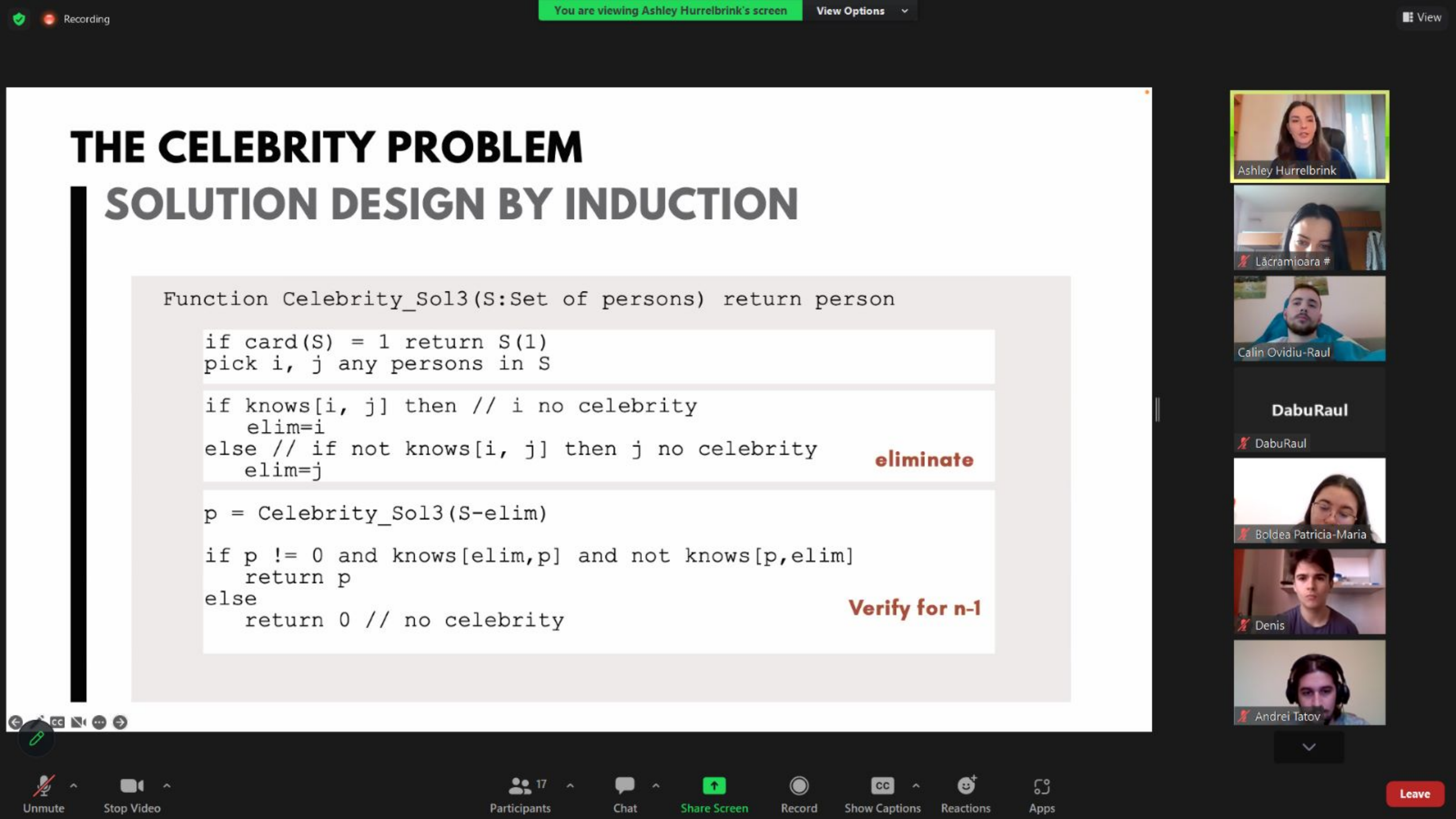
Given a n*n matrix with know[p, q] = true if p knows q and know[p, q] = false otherwise

WHAT WE NEED TO FIND

Determine whether there exists an i such that:
Know[j; i] = true (for all j, j≠ i) and Know[i; j] = false (for all j, j≠i)

CELEBRITY

someone who is known by everybody but does not know anyone



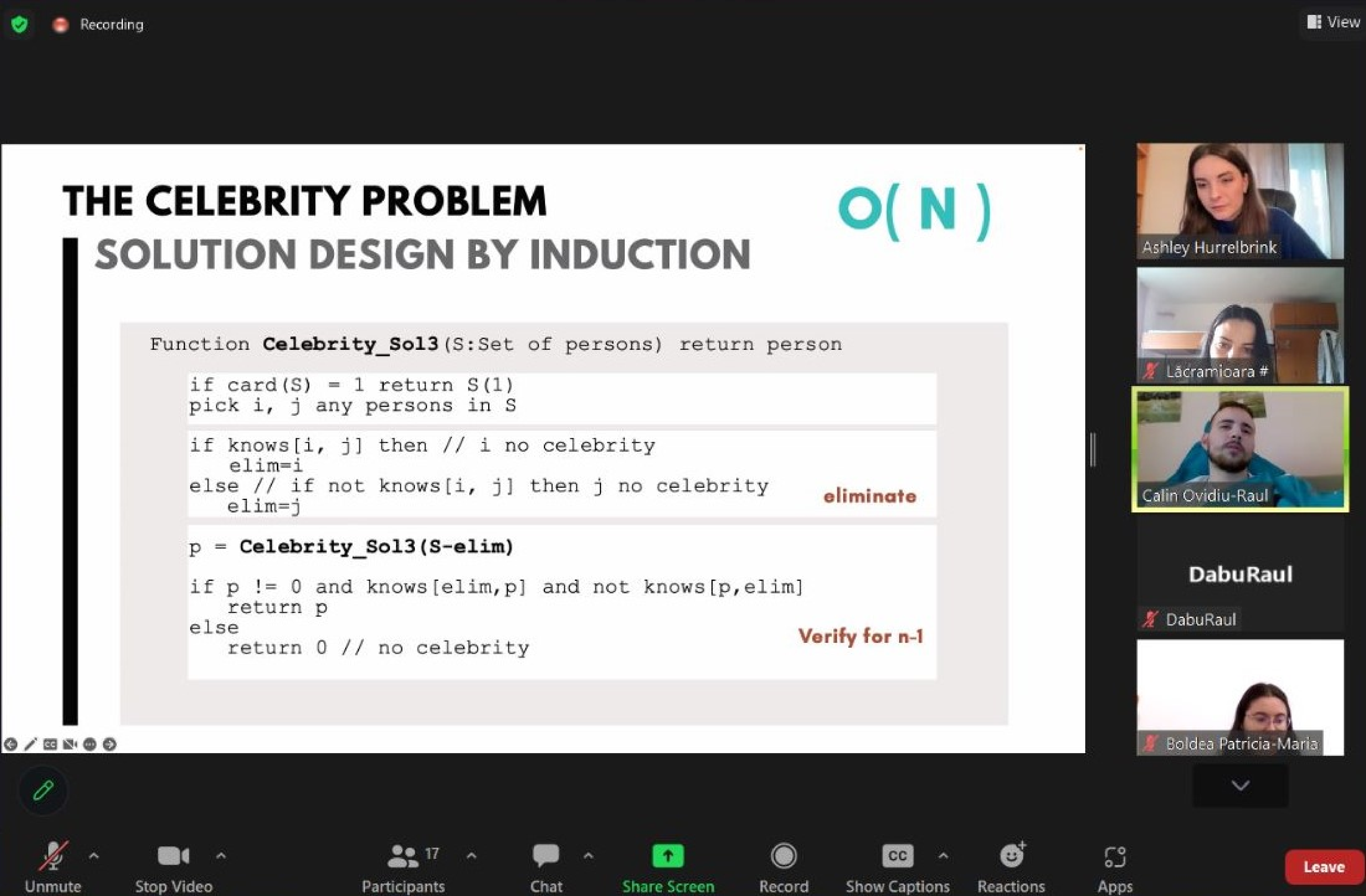
THE CELEBRITY PROBLEM

SOLUTION DESIGN BY INDUCTION

```
Function Celebrity_Sol3(S:Set of persons) return person

    if card(S) = 1 return S(1)
    pick i, j any persons in S

    if knows[i, j] then // i no celebrity
       elim=i
    else // if not knows[i, j] then j no celebrity
       elim=j

    p = Celebrity_Sol3(S-elim)

    if p != 0 and knows[elim,p] and not knows[p,elim]
       return p
    else
       return 0 // no celebrity
```

eliminate

Verify for n-1



THE CELEBRITY PROBLEM

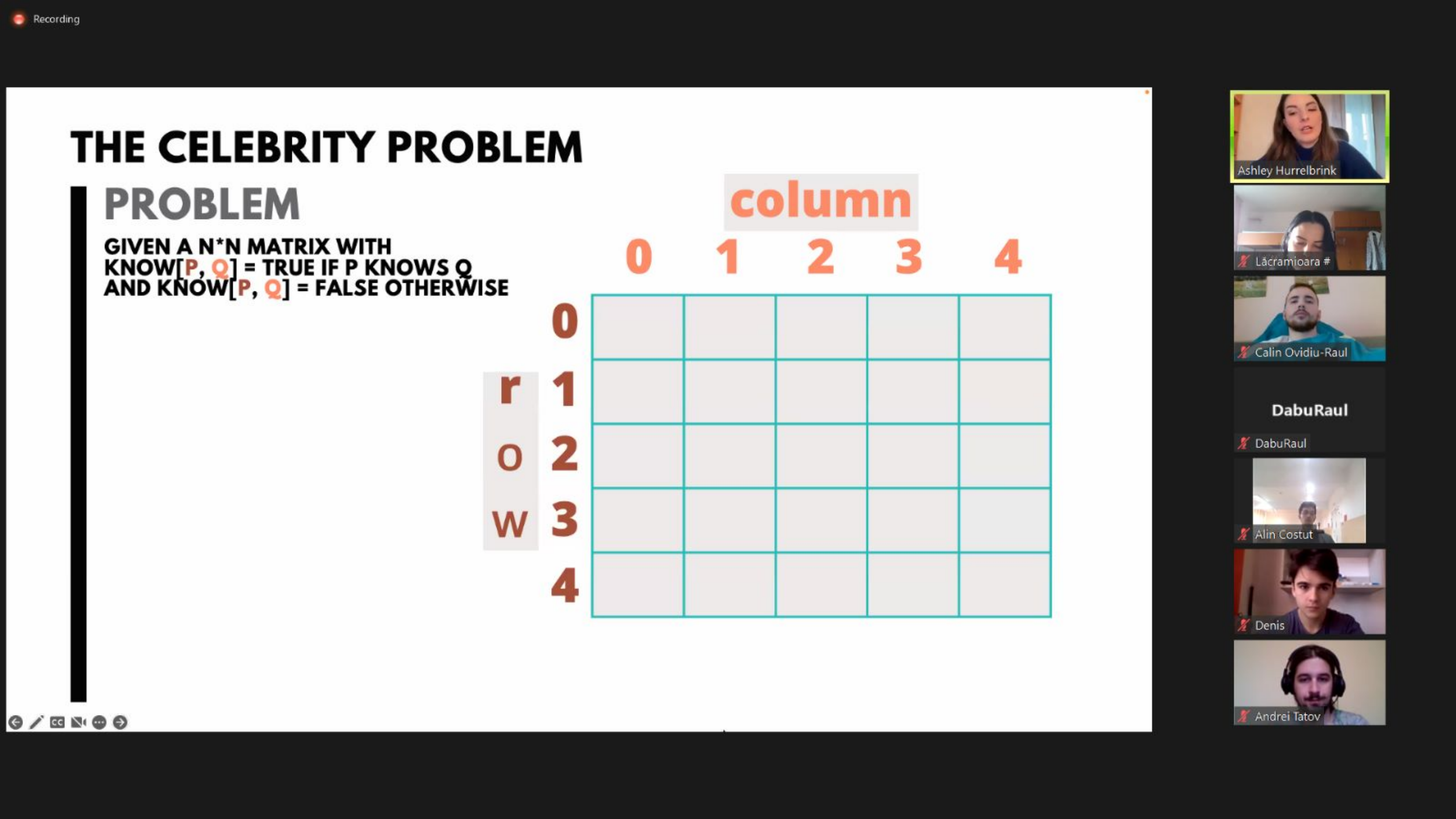
SOLUTION DESIGN BY INDUCTION

O(N)

```
Function Celebrity_Sol3(S:Set of persons) return person

    if card(S) = 1 return S(1)
    pick i, j any persons in S

    if knows[i, j] then // i no celebrity
       elim=i
    else // if not knows[i, j] then j no celebrity
       elim=j

    p = Celebrity_Sol3(S-elim)

    if p != 0 and knows[elim,p] and not knows[p,elim]
       return p
    else
       return 0 // no celebrity
```

eliminate

Verify for n-1



THE CELEBRITY PROBLEM

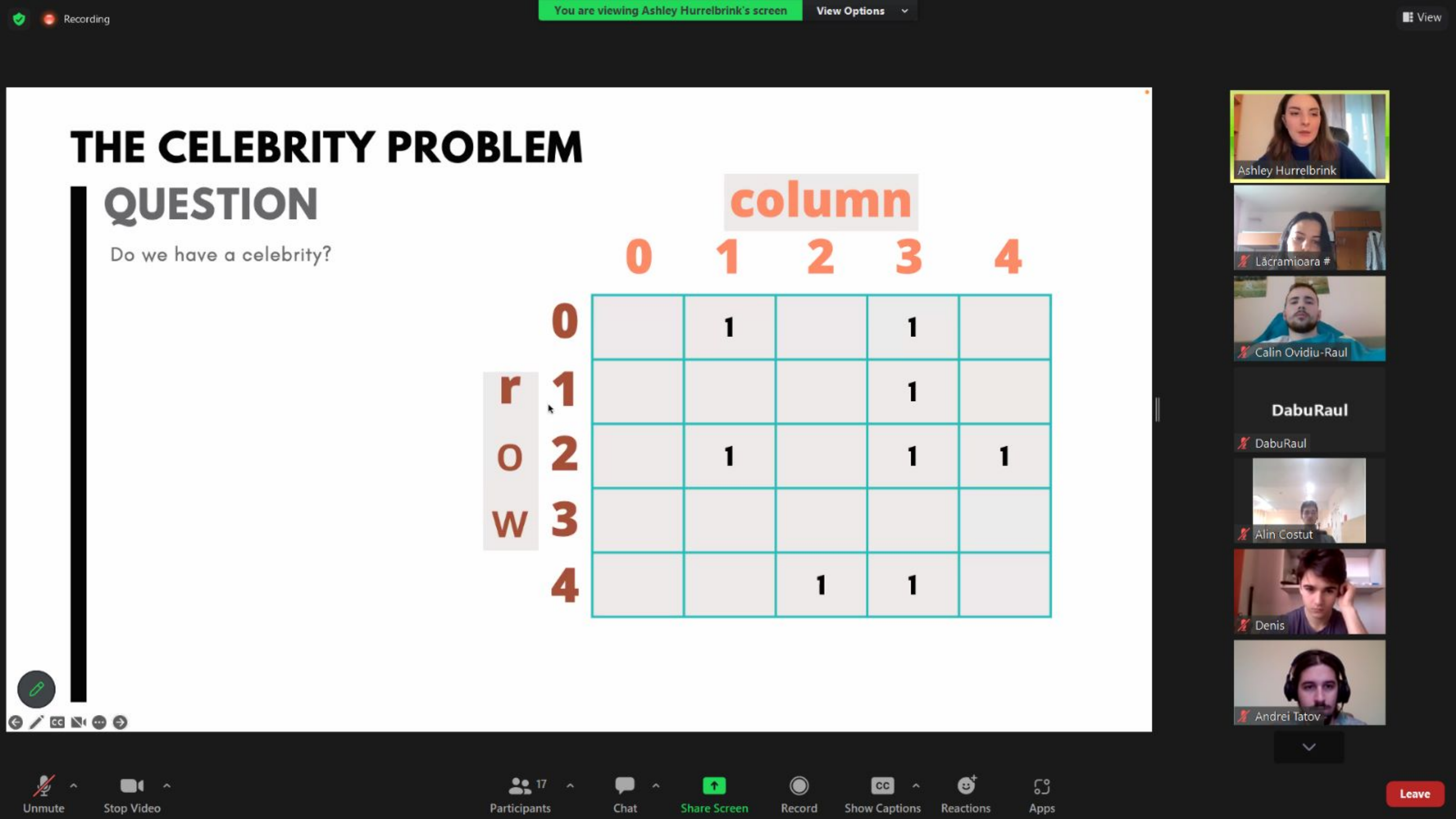
PROBLEM

GIVEN A N*N MATRIX WITH
KNOW[P, Q] = TRUE IF P KNOWS Q
AND KNOW[P, Q] = FALSE OTHERWISE

column

row

	0	1	2	3	4
0					
1					
2					
3					
4					

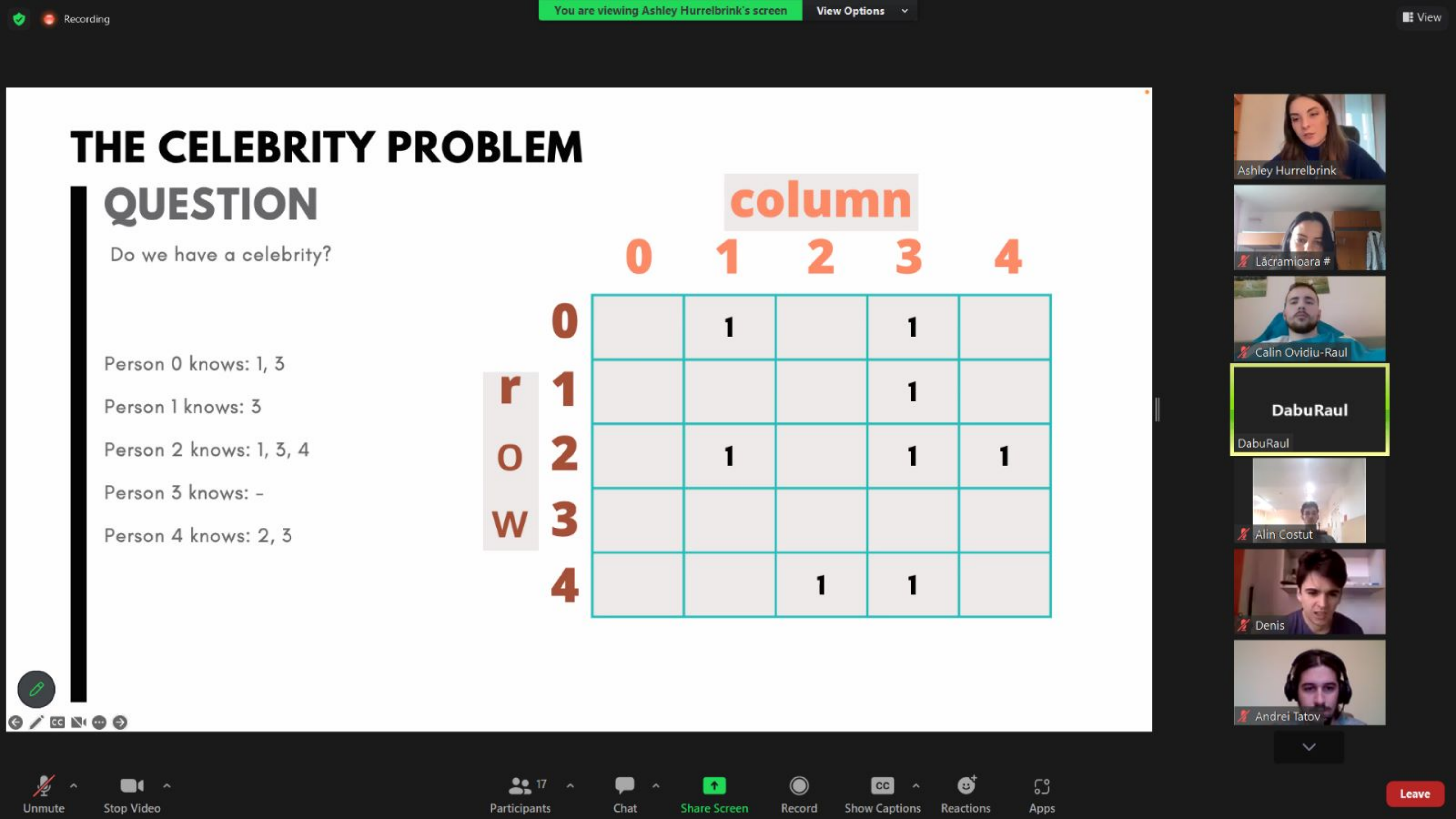


THE CELEBRITY PROBLEM

QUESTION

Do we have a celebrity?

		column				
		0	1	2	3	4
	0		1		1	
r	1				1	
o	2		1		1	1
w	3					
	4			1	1	



THE CELEBRITY PROBLEM

QUESTION

Do we have a celebrity?

Person 0 knows: 1, 3

Person 1 knows: 3

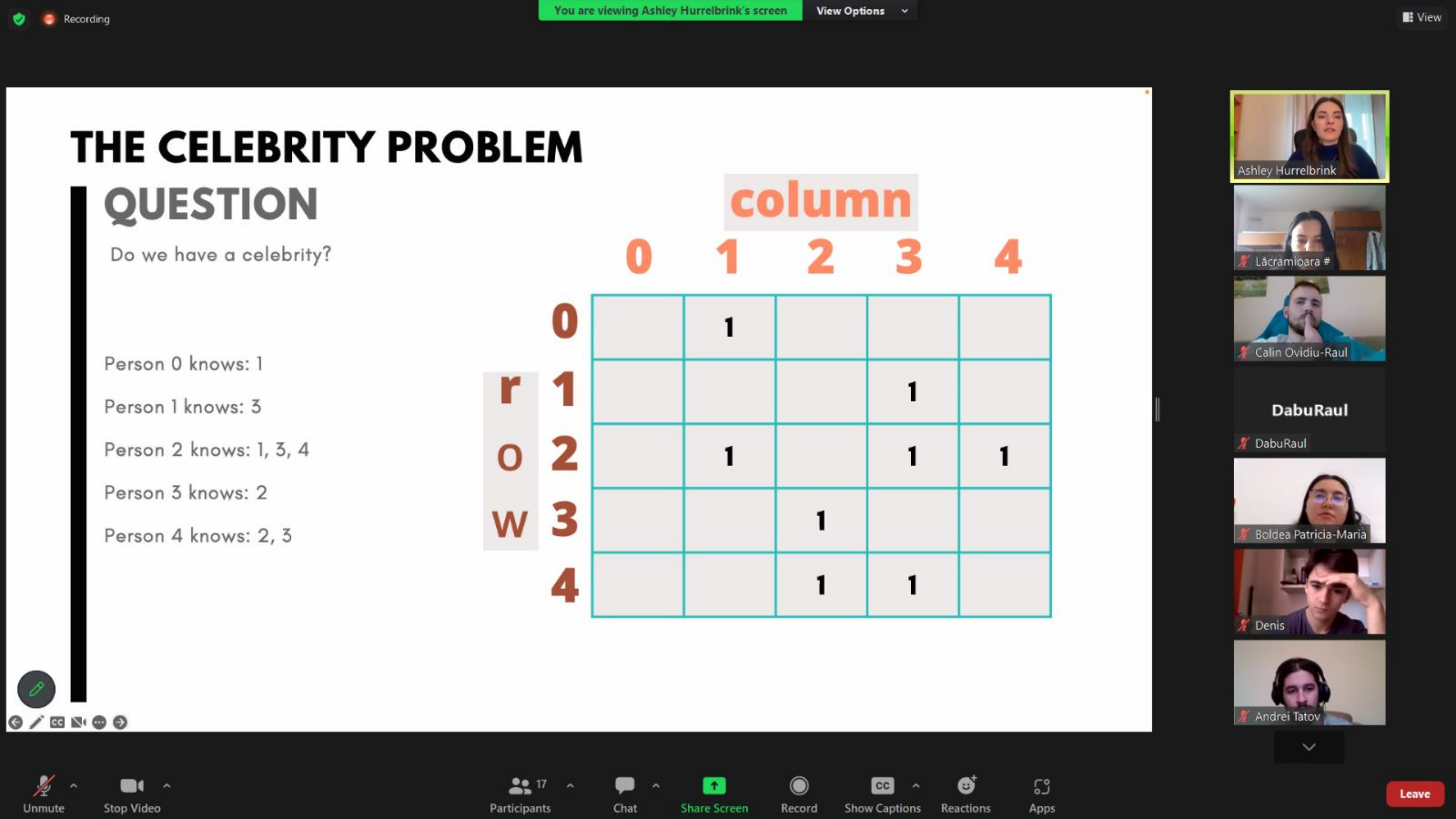
Person 2 knows: 1, 3, 4

Person 3 knows: -

Person 4 knows: 2, 3

column

		0	1	2	3	4
	0		1		1	
r	1				1	
o	2		1		1	1
w	3					
	4			1	1	



THE CELEBRITY PROBLEM

QUESTION

Do we have a celebrity?

Person 0 knows: 1

Person 1 knows: 3

Person 2 knows: 1, 3, 4

Person 3 knows: 2

Person 4 knows: 2, 3

column

row

	0	1	2	3	4
0		1			
1				1	
2		1		1	1
3			1		
4			1	1	