Pattern Solving

Foundation Course on Data Structures & Algorithm - Part I

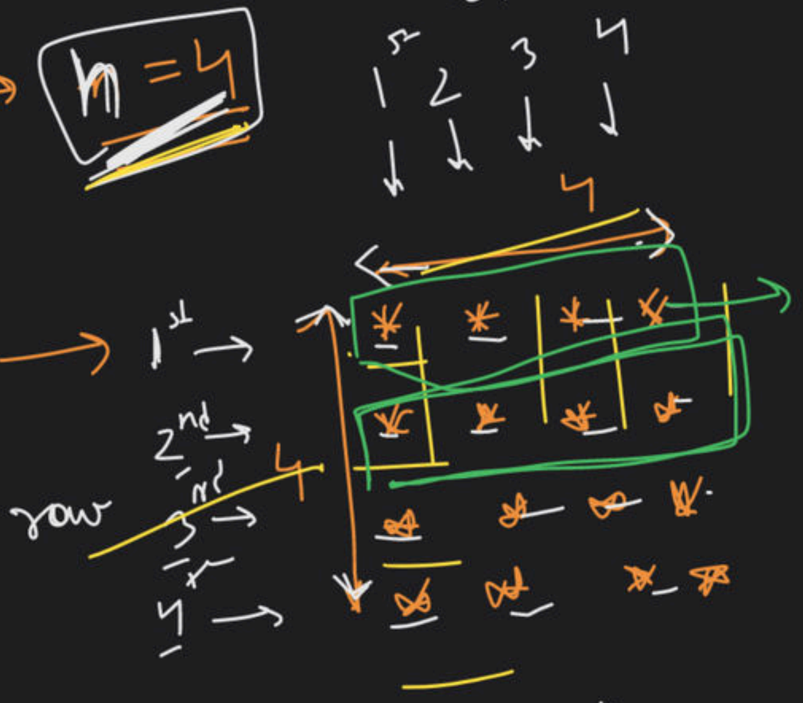
Love Babbar • Lesson 4 • Apr 19, 2022

→ Pattern

~25 ⟹ Solve

analytical

(1) i/p → n = 4



square pattern

o/p → 1st → * * * *

2nd → * * * *

3rd → * * * *

4th → * * * *

cout << endl

n = 4

↳ row

↳ col

Data

↳ row

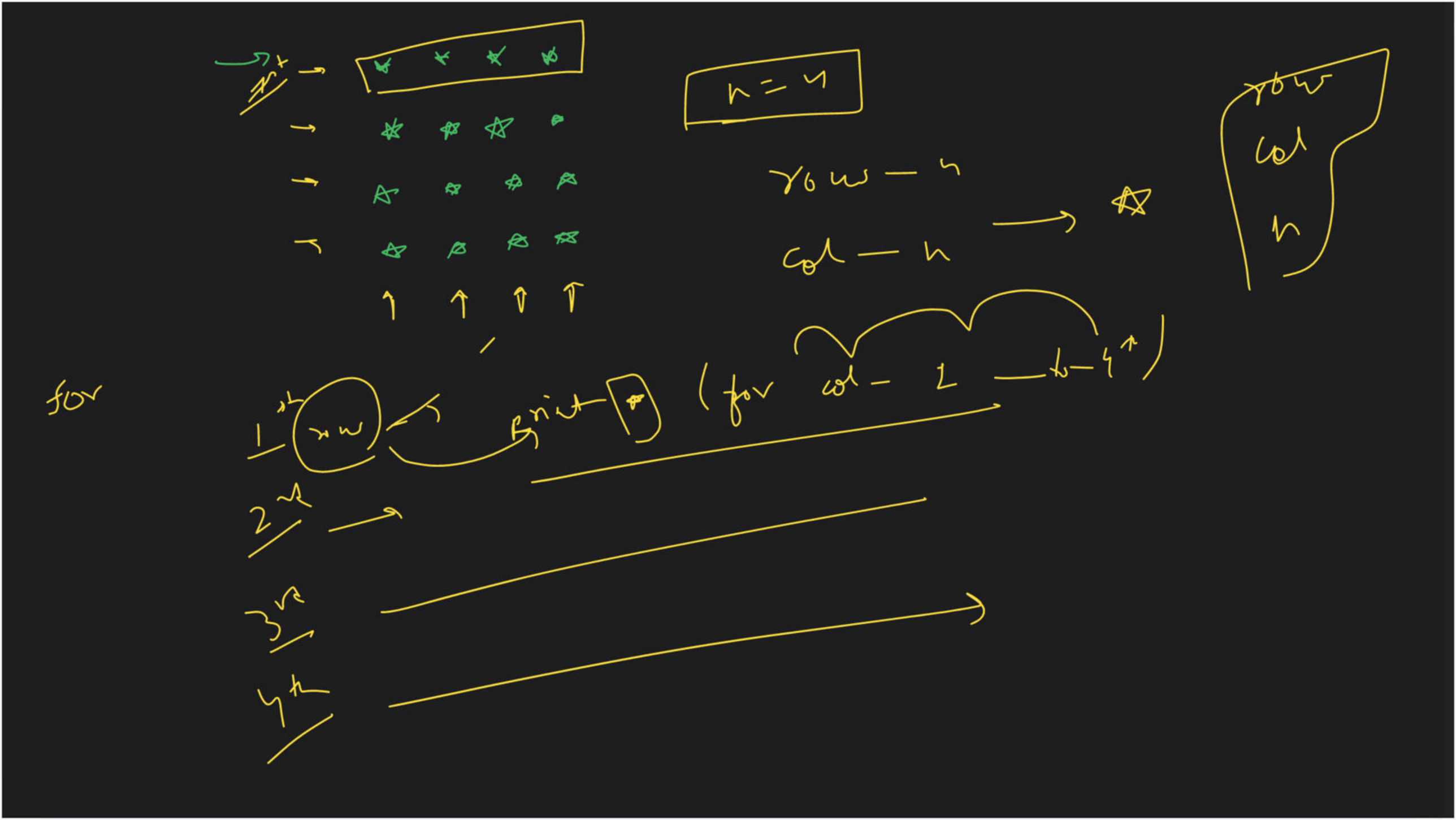
↳ col

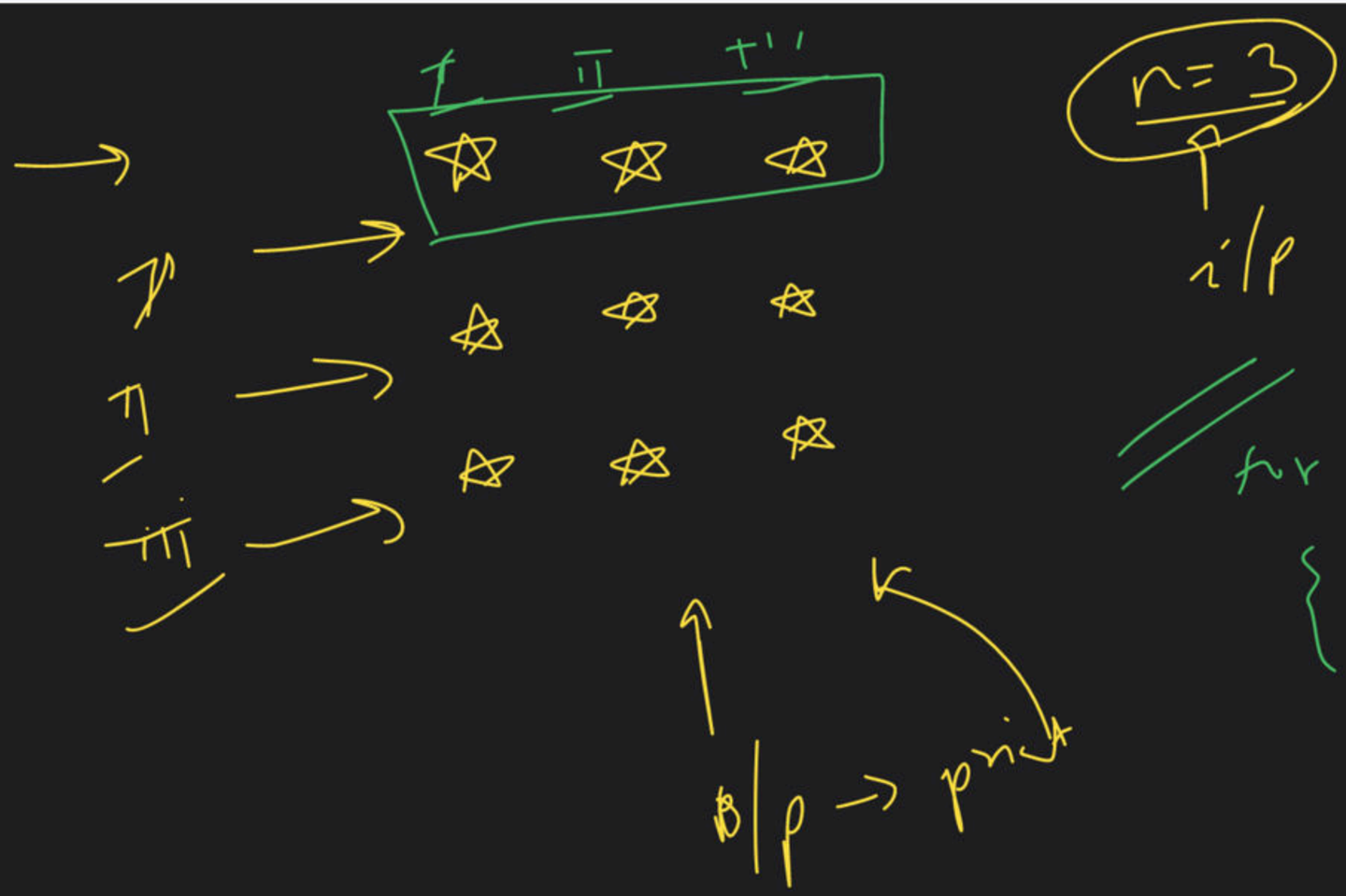
↳ 4

4 row

↳ each → 4 start

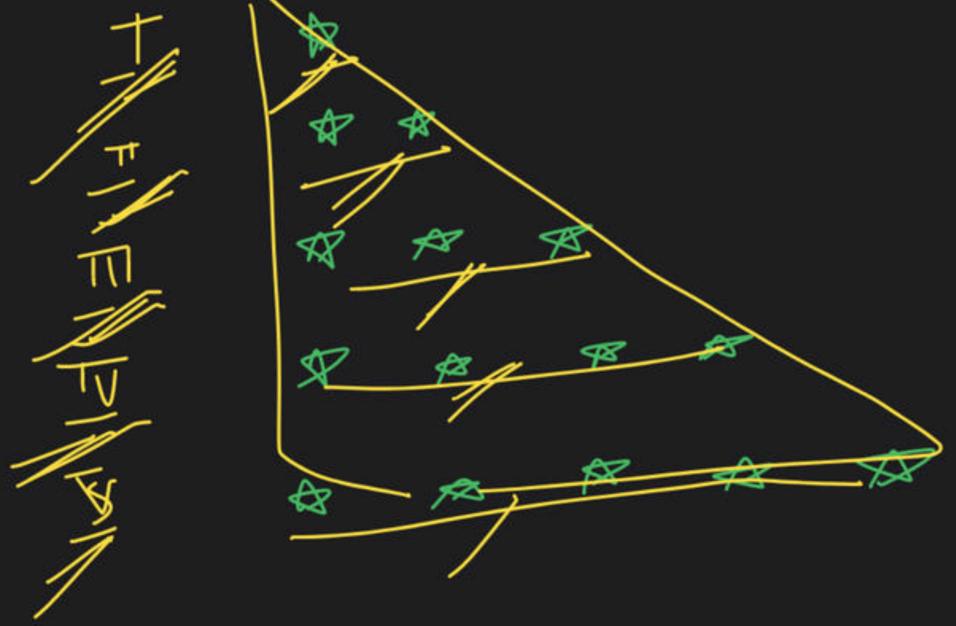
4 th r


for
print
(for
1
2
3
4


I
II
III
n=3
i/p
o/p -> print
for (row -> 1 to n)
{
for (col -> 1 to -
{
cout <<
}
cout <<endl
}

Half - Pyramid

ip → n = 5



print count = row no

n = 5

for (row - 1 to n)

{

// for each row

logic

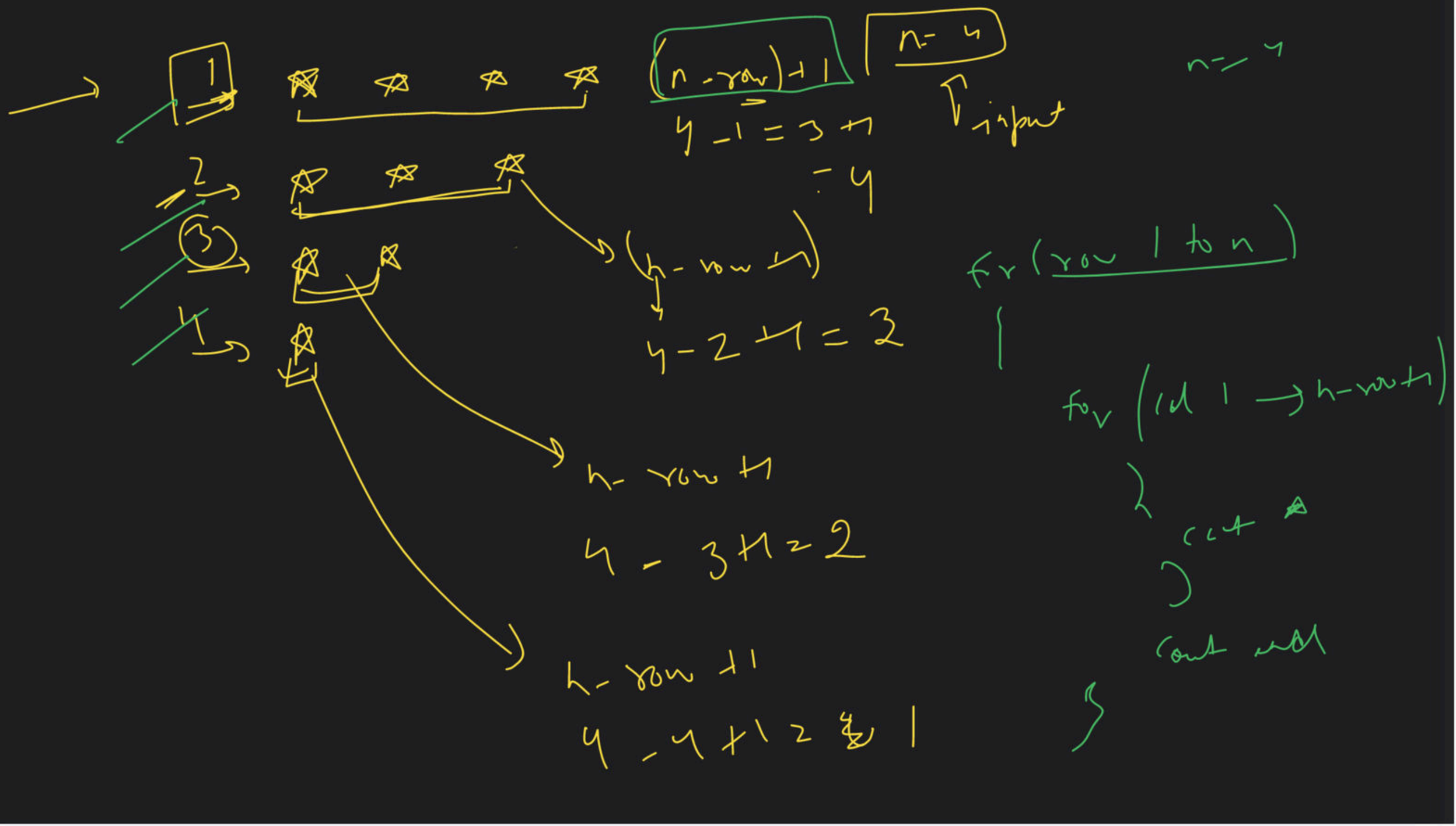
→ print count = row no

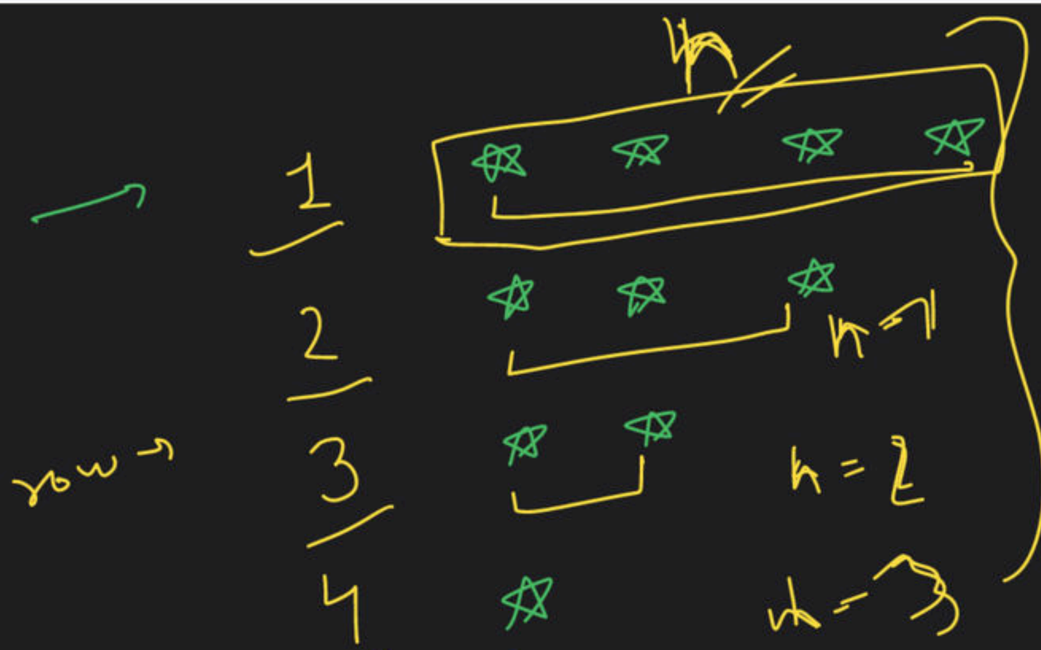
for (int col 1 → row no)

{

}

}


1
2
3
4
(n - row) + 1
n - 4
input
4 - 1 = 3 + 1
= 4
(4 - row + 1)
4 - 2 + 1 = 3
4 - row + 1
4 - 3 + 1 = 2
4 - row + 1
4 - 4 + 1 = 2 & 1
n - 4
for (row 1 to n)
{
for (id 1 → 4 - row)
}
cout *
)
cout endl
}



n - row + 1

r = 4

for (row → 1 to n)

{

// for each row start count

for (col 1 → n - row + 1)

{

count *

}

cout << endl

}

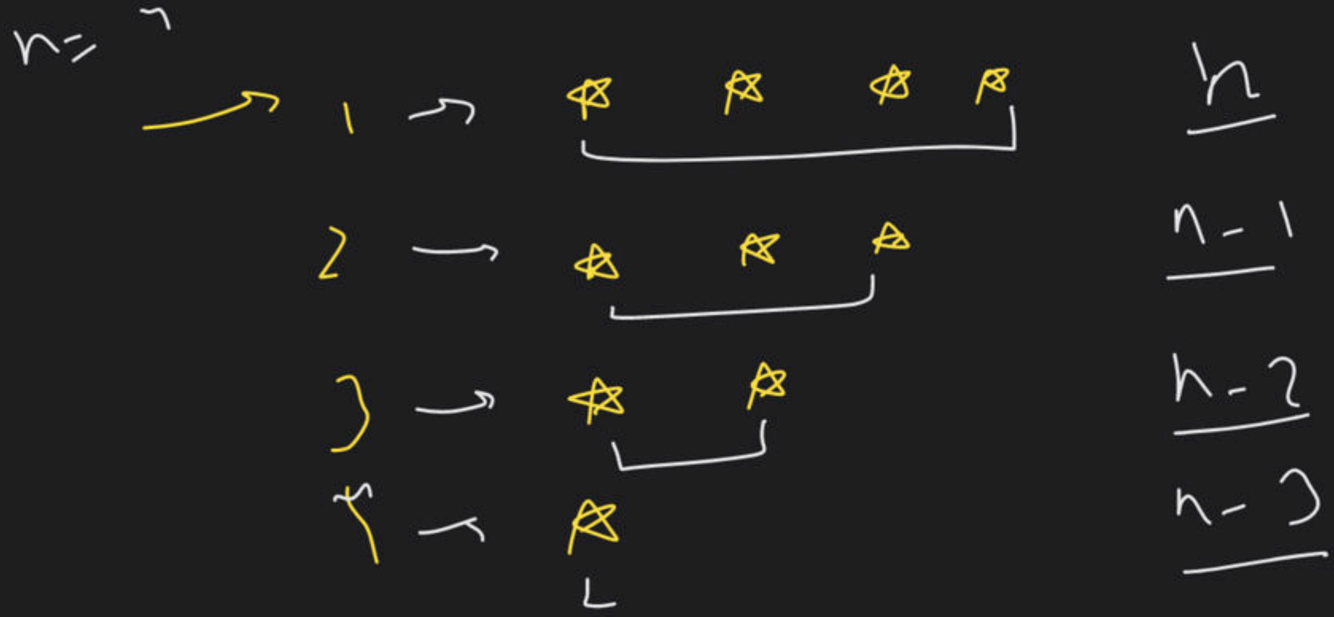
0 → < n

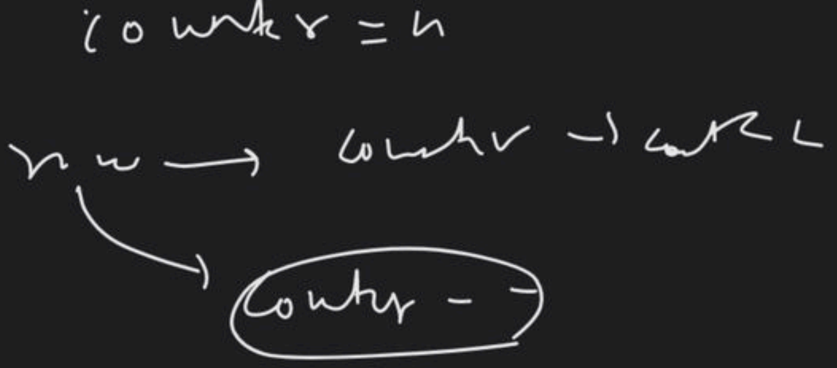
1st row → (4) start → 4 - 1 + 1 = 4

2nd → (3) star → 4 - 2 + 1 = 3

3rd → 2 star → 2

4th → 1 star → 1





$2 = 3^R$ approach

n = 4

n - row → spaces

4 - 1 = 3

(3)

n = 4

row = 1

for (int row → 1 — n)

{

n = 4, row = 2

n - row

spaces

(3) = n - row

spaces

row

for (col 1 → n - row)

{ cout << " " ;

}

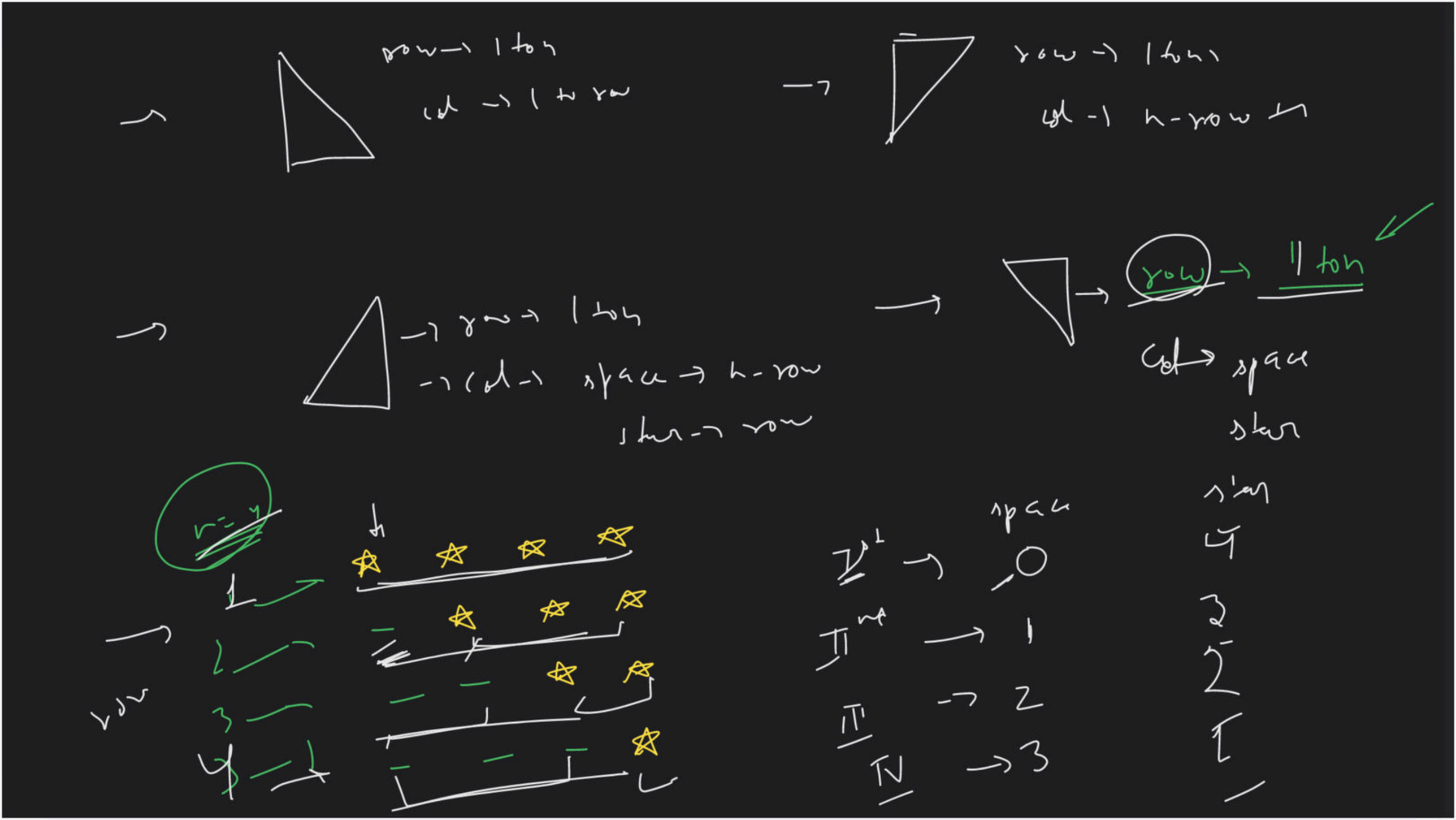
for (col → 1 → row)

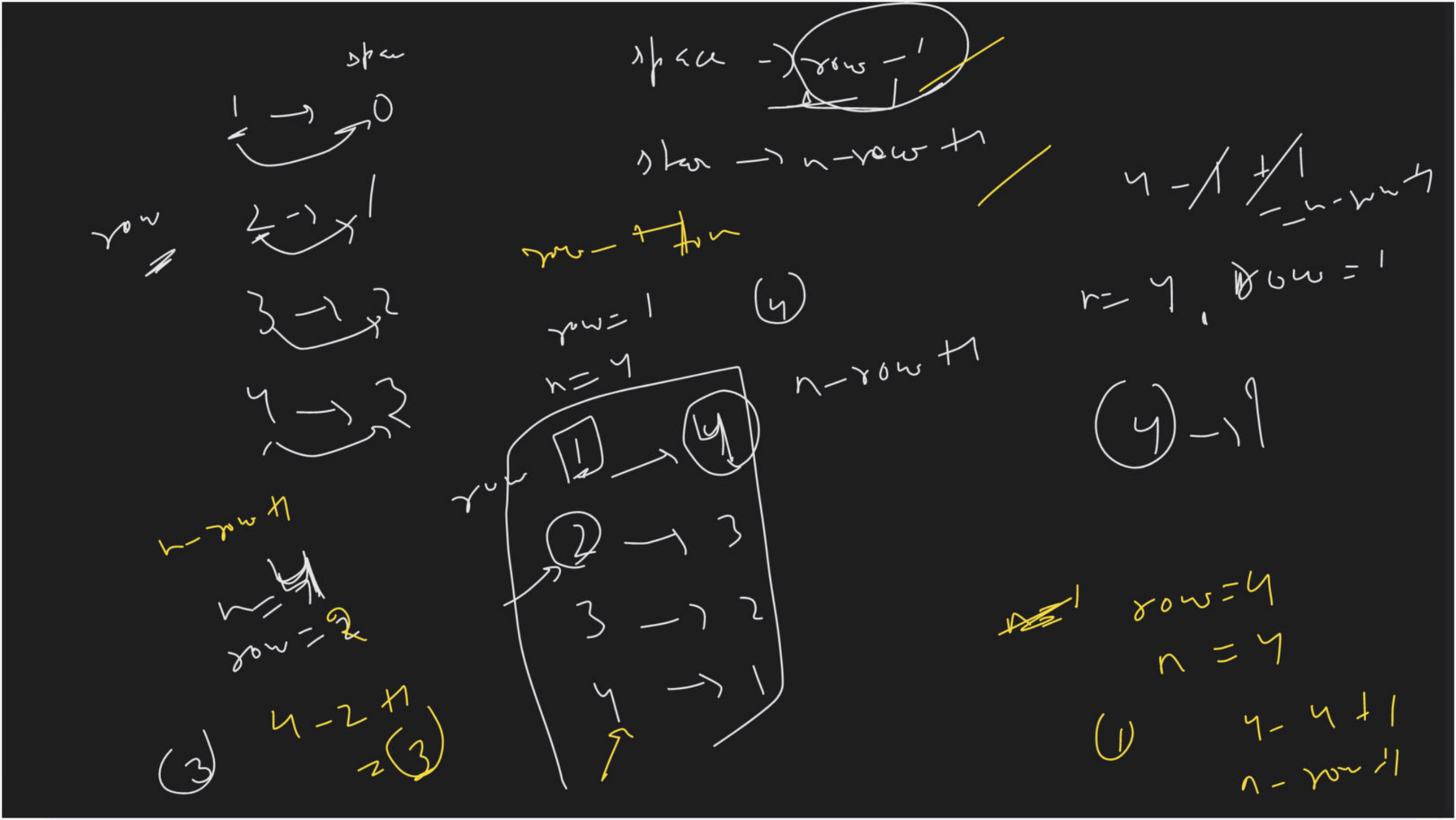
{ cout << '*' ;

}

cout << endl

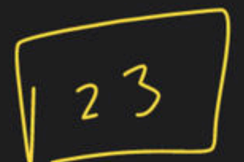
}


row
col
space
star
I
II
III
IV



= → assignment

V = 23

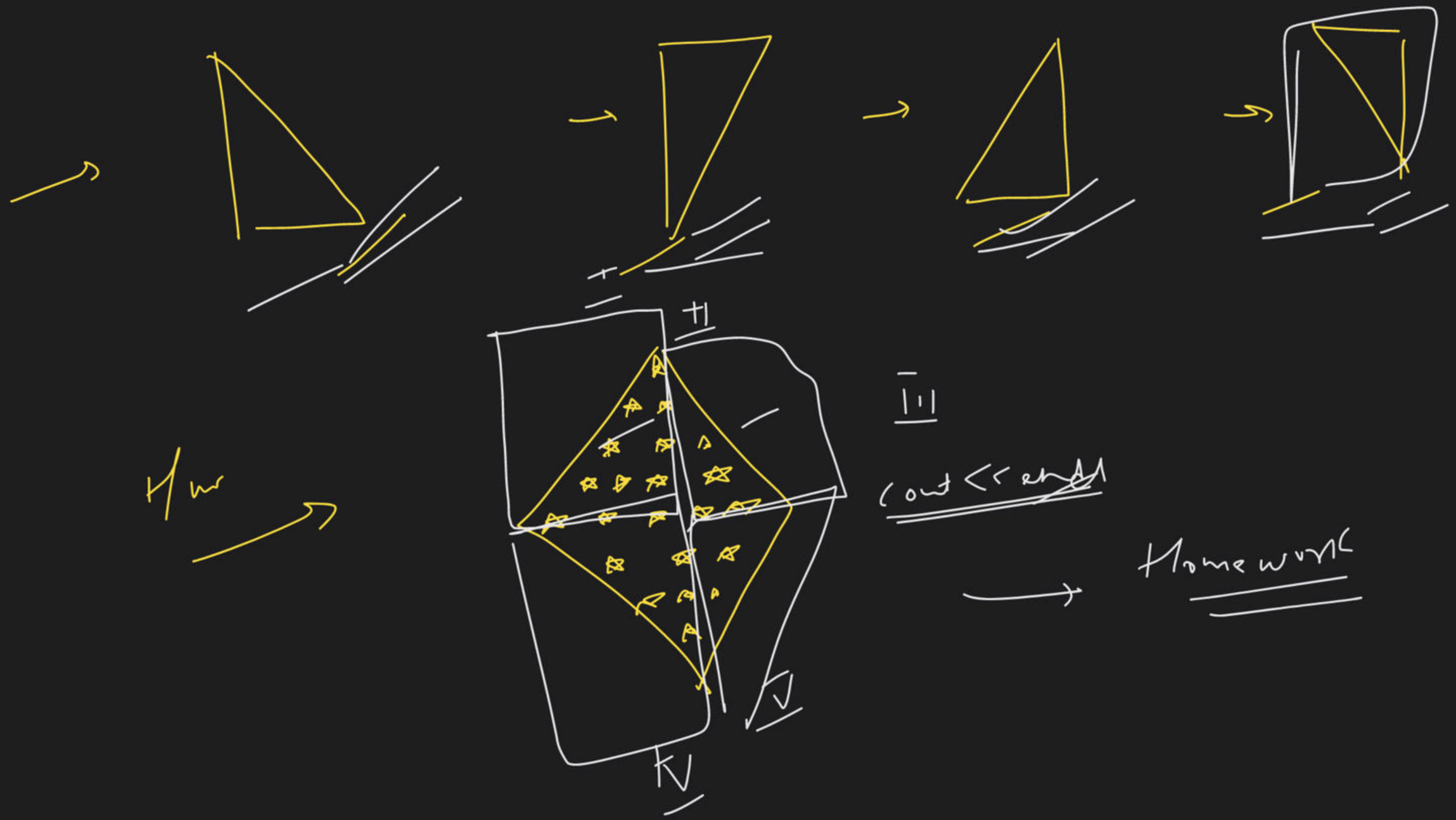


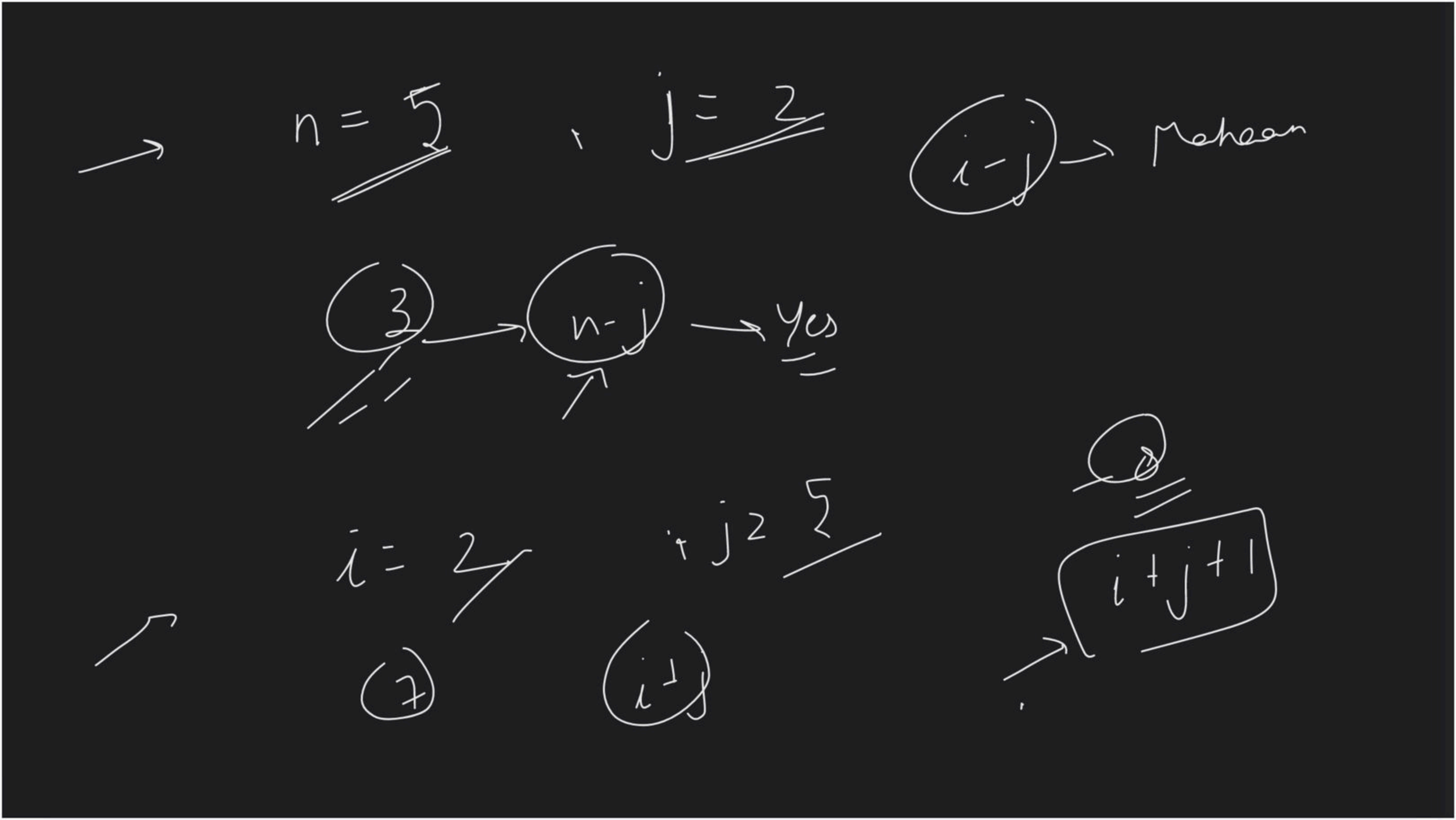
<= → T/F

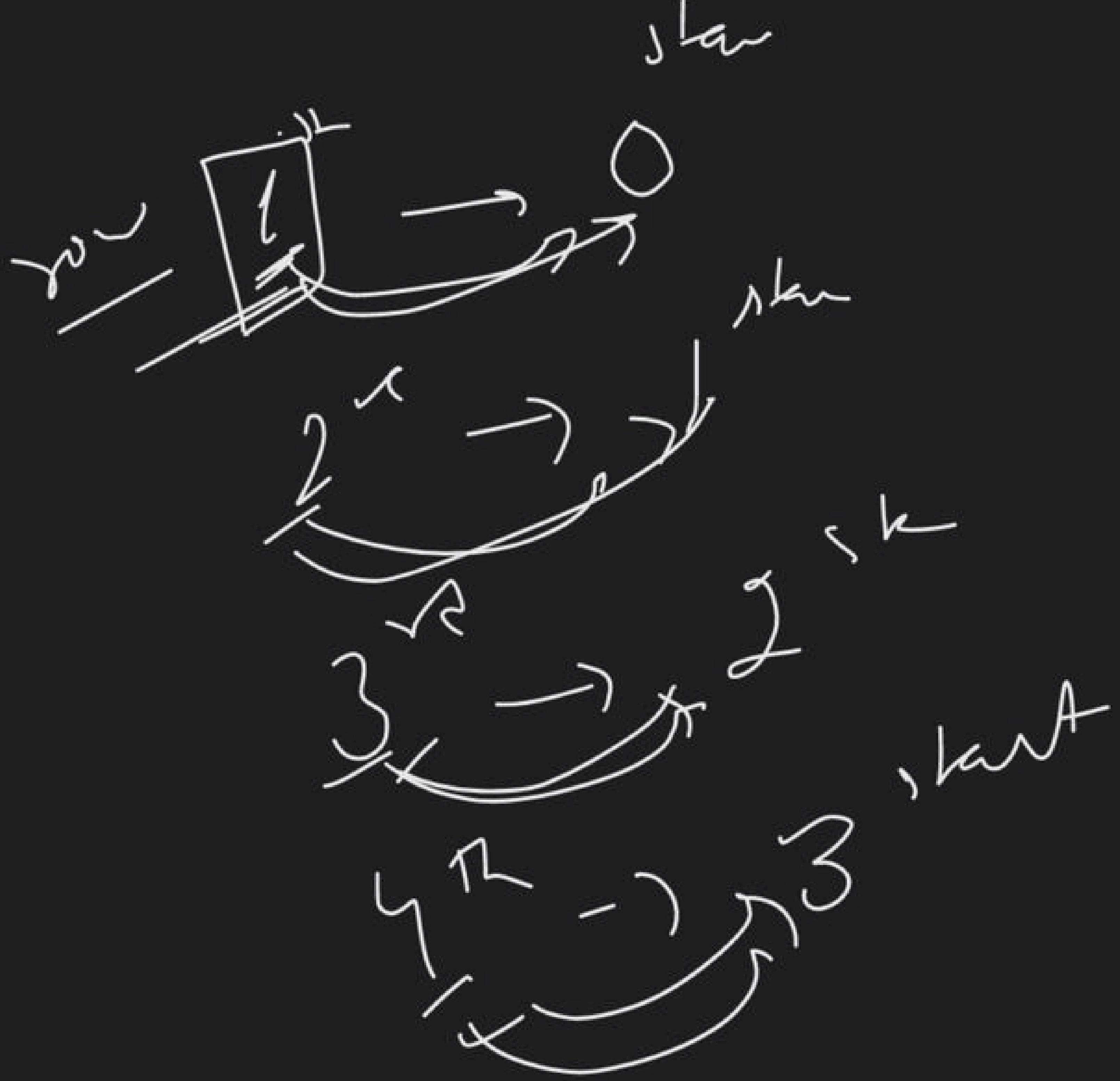
↳ comparision

a <= b

↳ T/F


II
III
cout << endl;
Homework
V
IV


n = 5
j = 2
i - j
Mohsan
3
n - j
Yes
i = 2
i + j ≥ 5
7
i - 1
i + j + 1

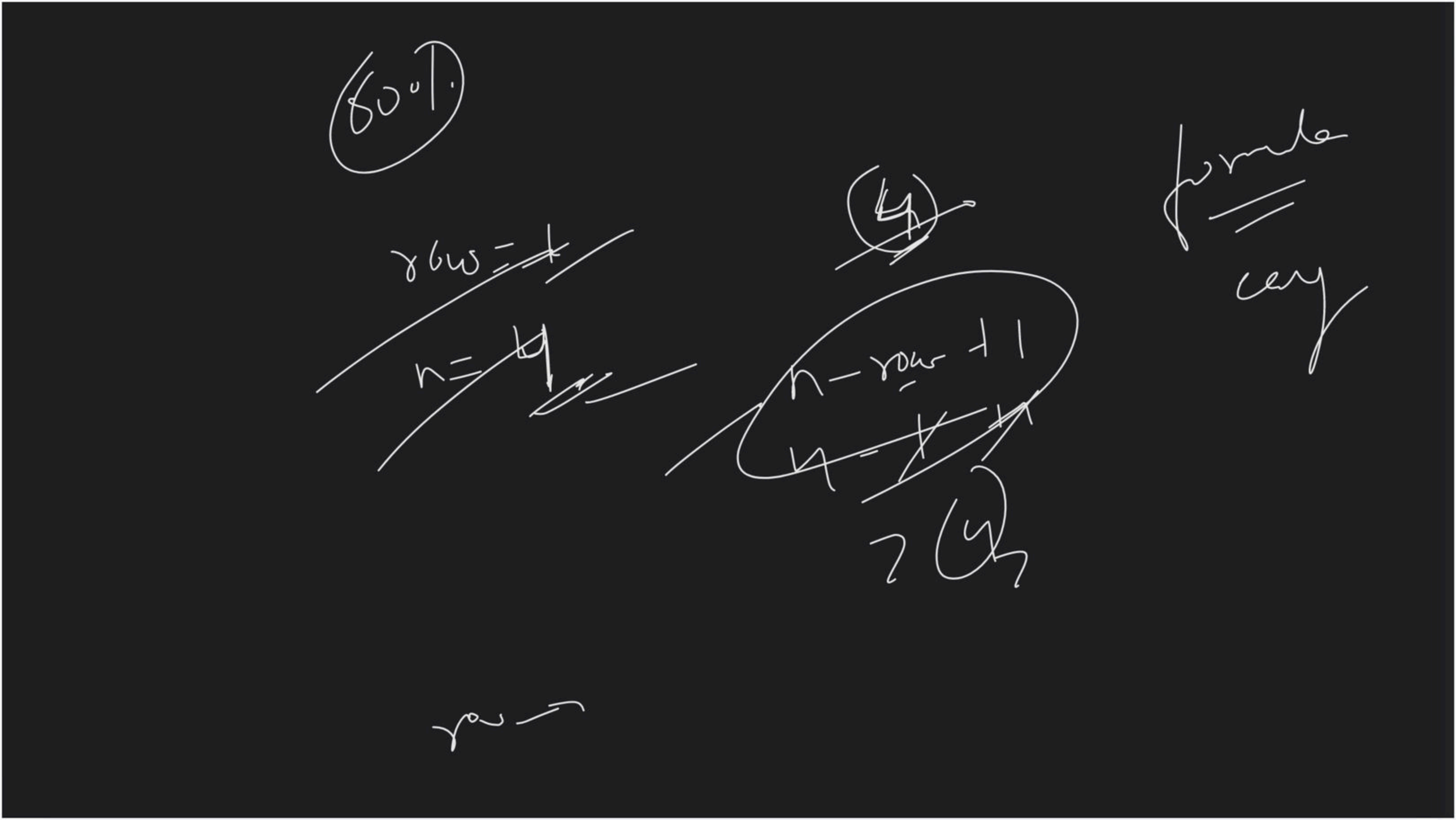


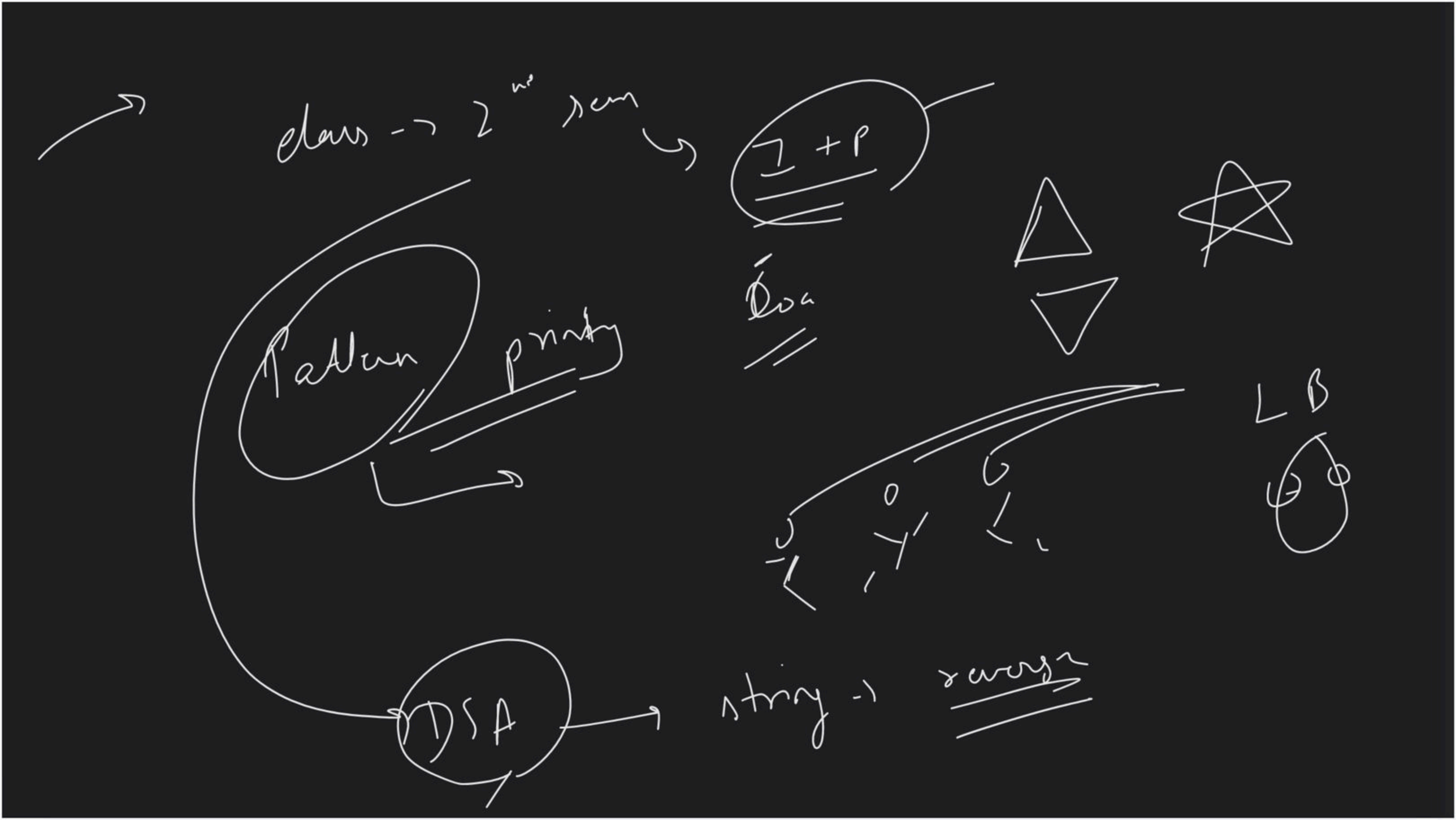
row - 1

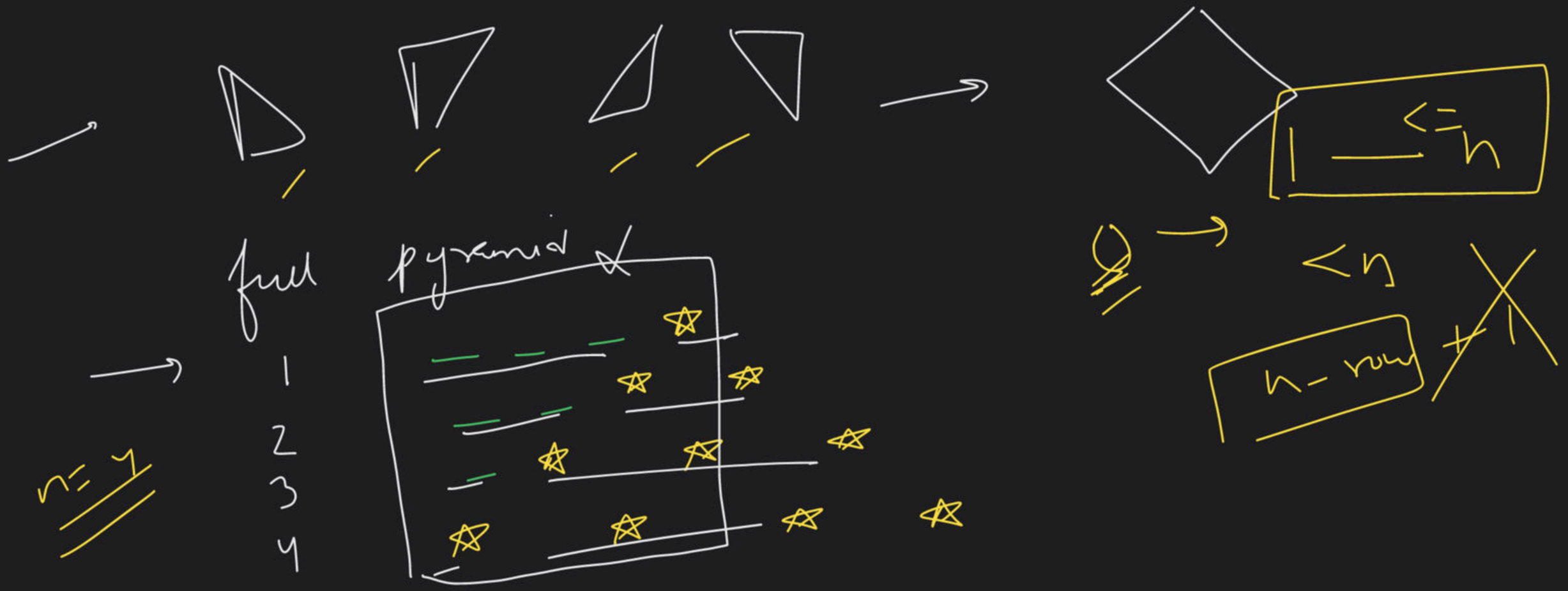
r = 4

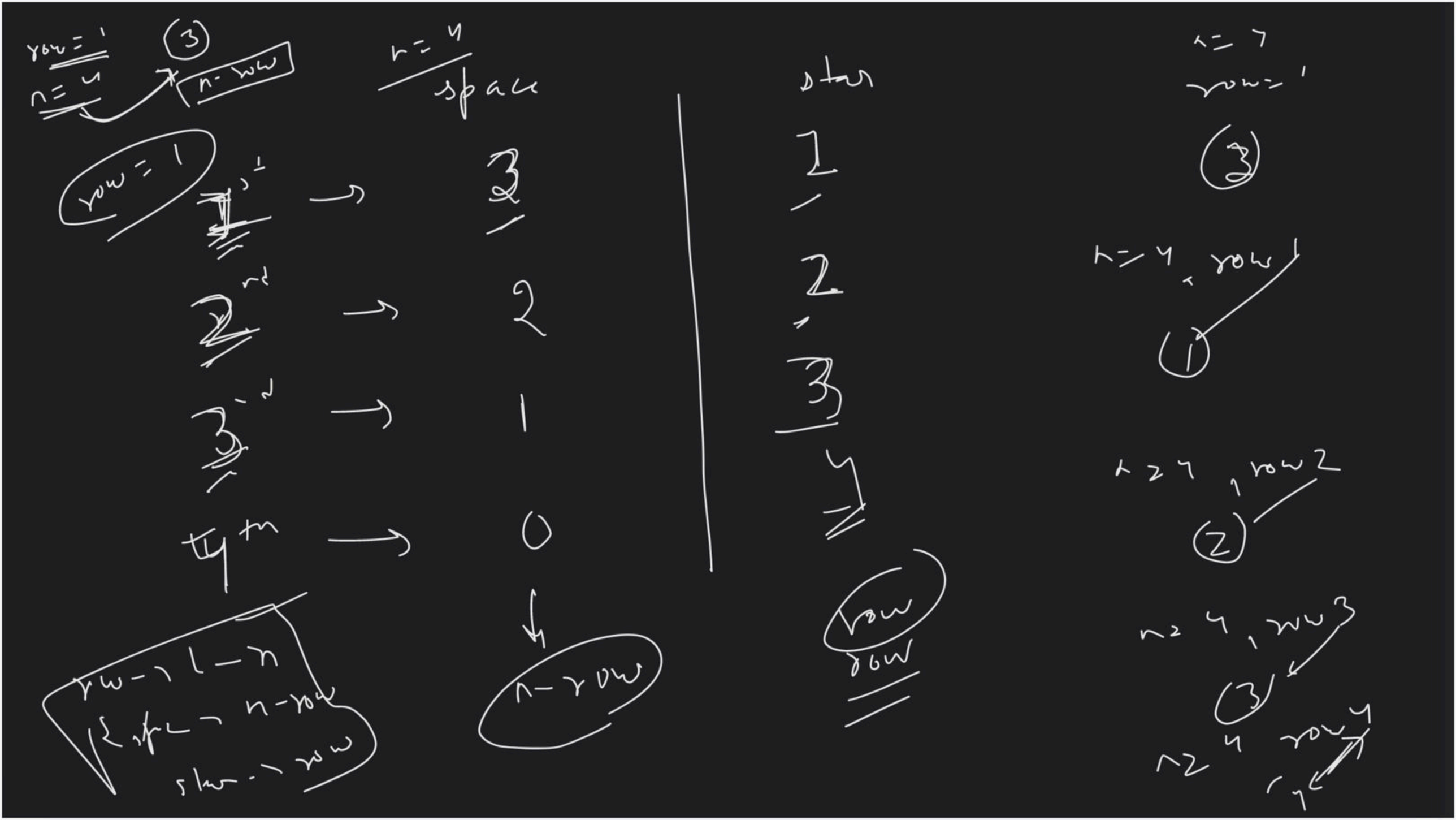
row = 1

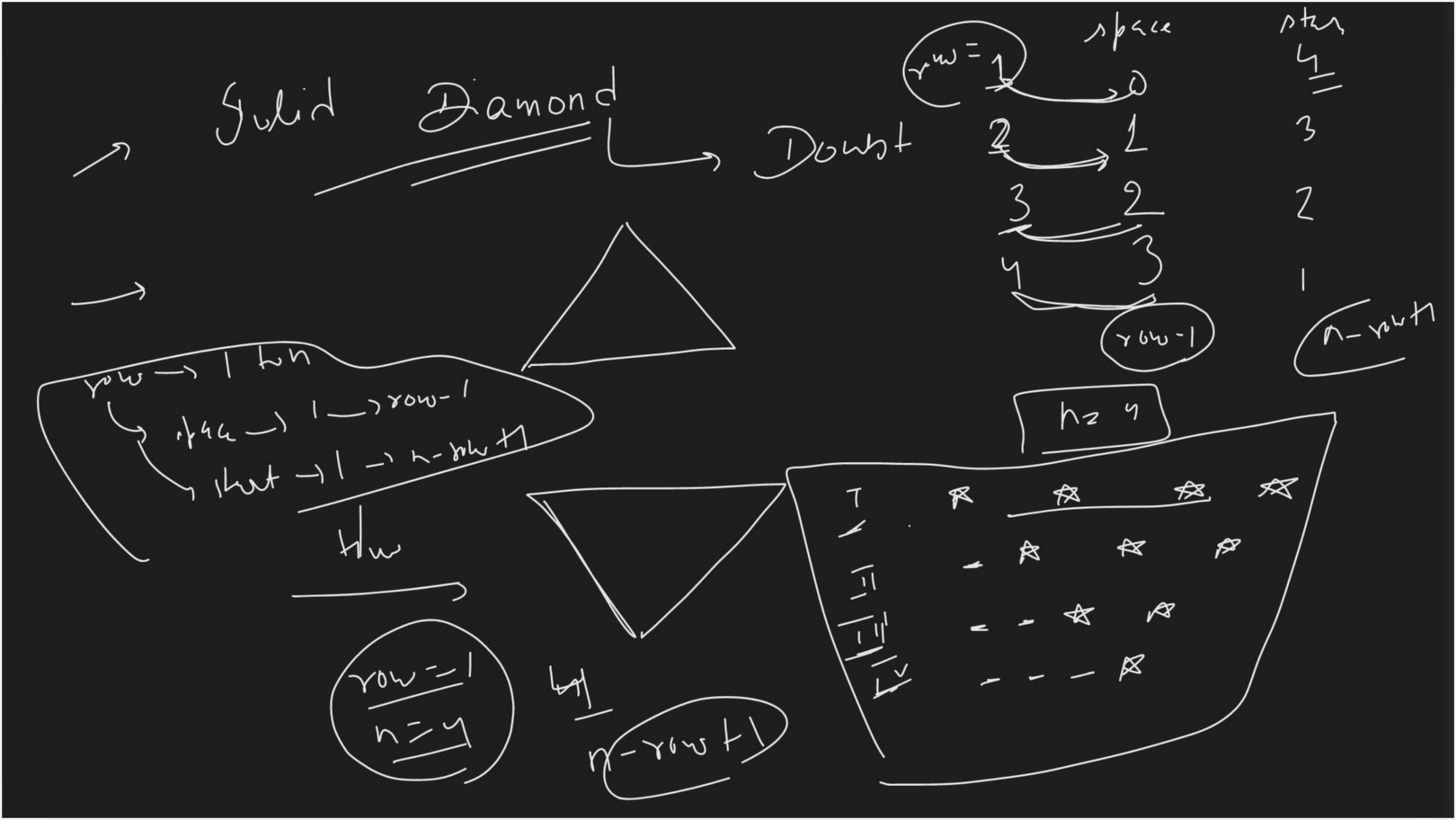
0


Formula


class -> 2 nos
sum
I + P
Pattern
print
DSA
DSA
string -1
average
LB


full pyramid
1
2
3
4
n = 4
<= n
0
< n
n - num
+ 1


row = 1
n = 4
3
n - row
n = 4
space
row = 1
1st → 3
2nd → 2
3rd → 1
4th → 0
4 - row
row = 1 - n
col - n - row
star - row
1
2
3
4
row
row
n = 7
row = 1
3
n = 4, row 1
1
n > 4, row 2
2
n > 4, row 3
3
n > 4 row 4
4


Solid Diamond
Doubt
row = 1
space
stars
4
0
2
1
3
3
2
2
4
3
1
row-1
n-row+1
row → 1 to n
space → 1 → row-1
stars → 1 → n-row+1
row = 1
n = 4
n-row+1
h = 4

→ Hollow inverted Half Pyramid

n = 6

n = 6
row = 3

(2)

n - row - 1
6 - 3 - 1

(1)

(3)

n = 6
row 2

n - row - 1

→ sahi - ?

row = 6

6 - 4 - 1

```
for(row → 1 → n)
{
  if(row==1 || row==n)
  {
    ==========
  }
  else
  {
    cout << *
    //spaces
    cout << *
  }
}
```

row = 5
n = 1

n - row - 1
6 - 5 - 1
= 0

for (row → 1 → n)

{

if (row == 1 || row == 1)

{

// star

for(col 1 → n-row+1)

cout << "*"

}

else

{ cout << "*"

for (col → 1 → n-row-1)

{ cout << " "

}

cout << "*"

}

1 → i

n = 6 (6)

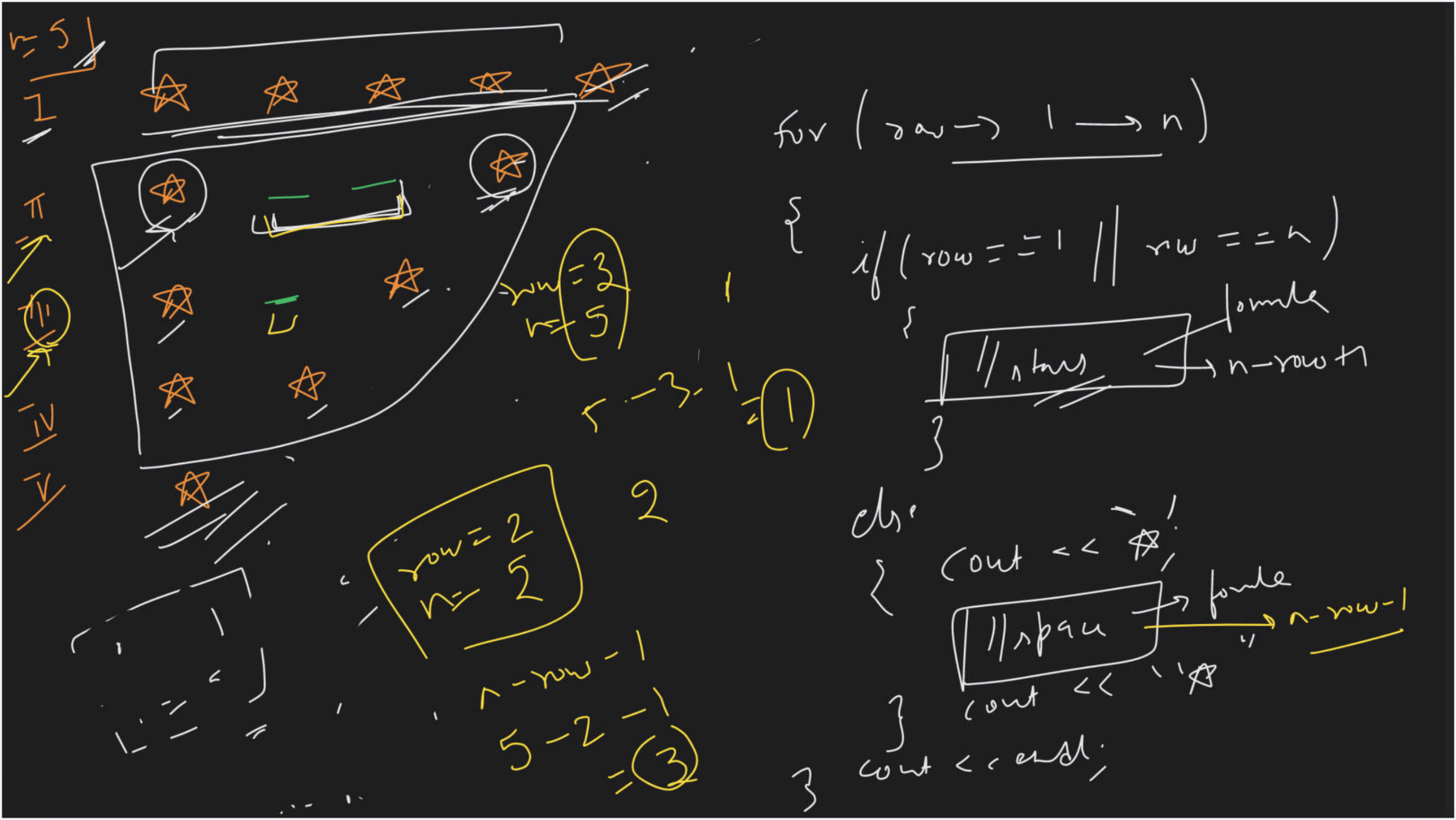
row = 1

n-row+1

n = 6 1

row = 6

n-row-1


n = 5
I
II
III
IV
V
row = 3
n = 5
r = 3 - 1 = (1)
row = 2
n = 2
2
n - row - 1
5 - 2 - 1
= (3)
for (row -> 1 -> n)
{
if (row == 1 || row == n)
{
// stars
print n-row th
}
else
{
cout << " ";
// space
print n-row-1
cout << "*";
}
cout << endl;
}