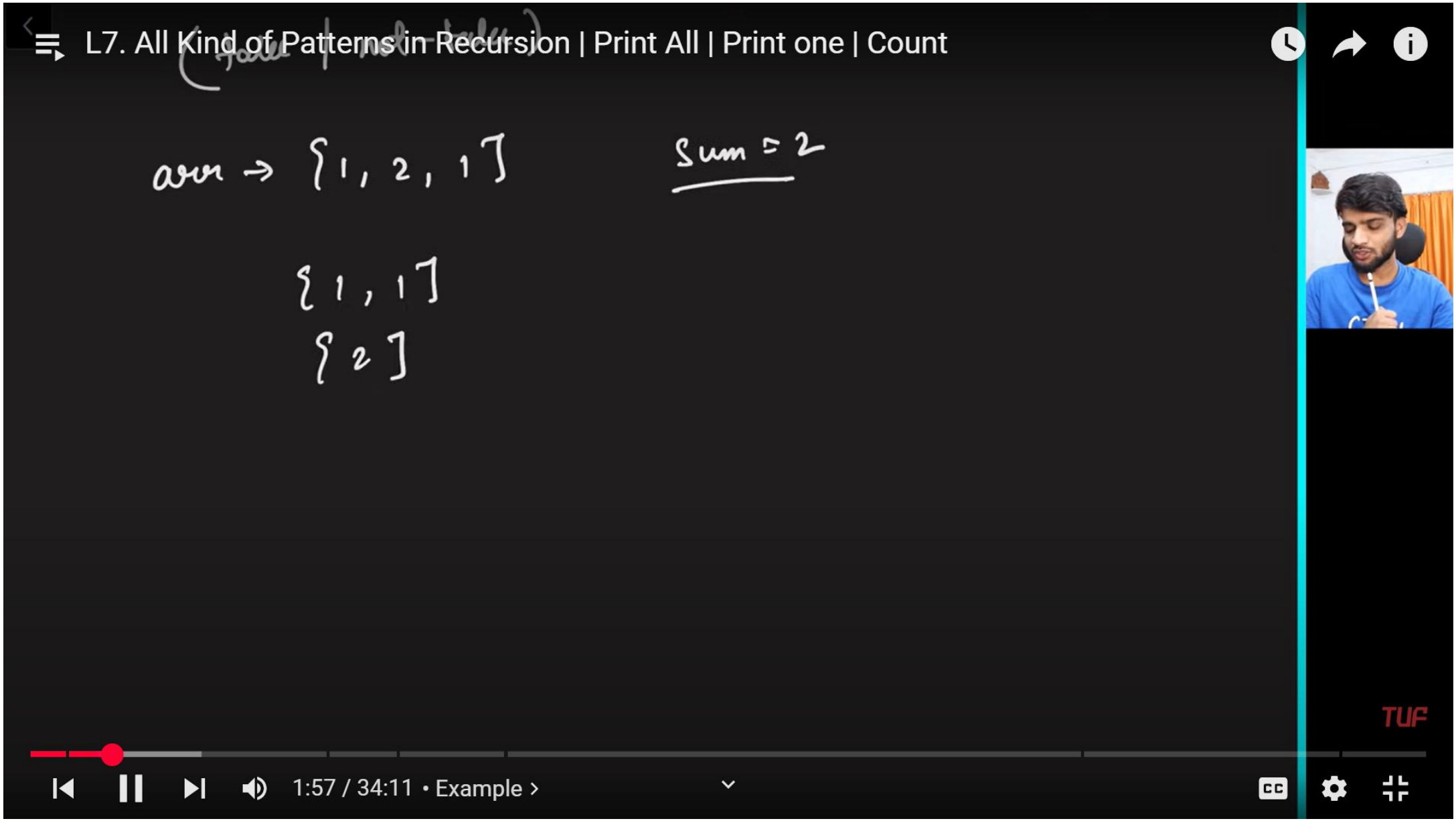
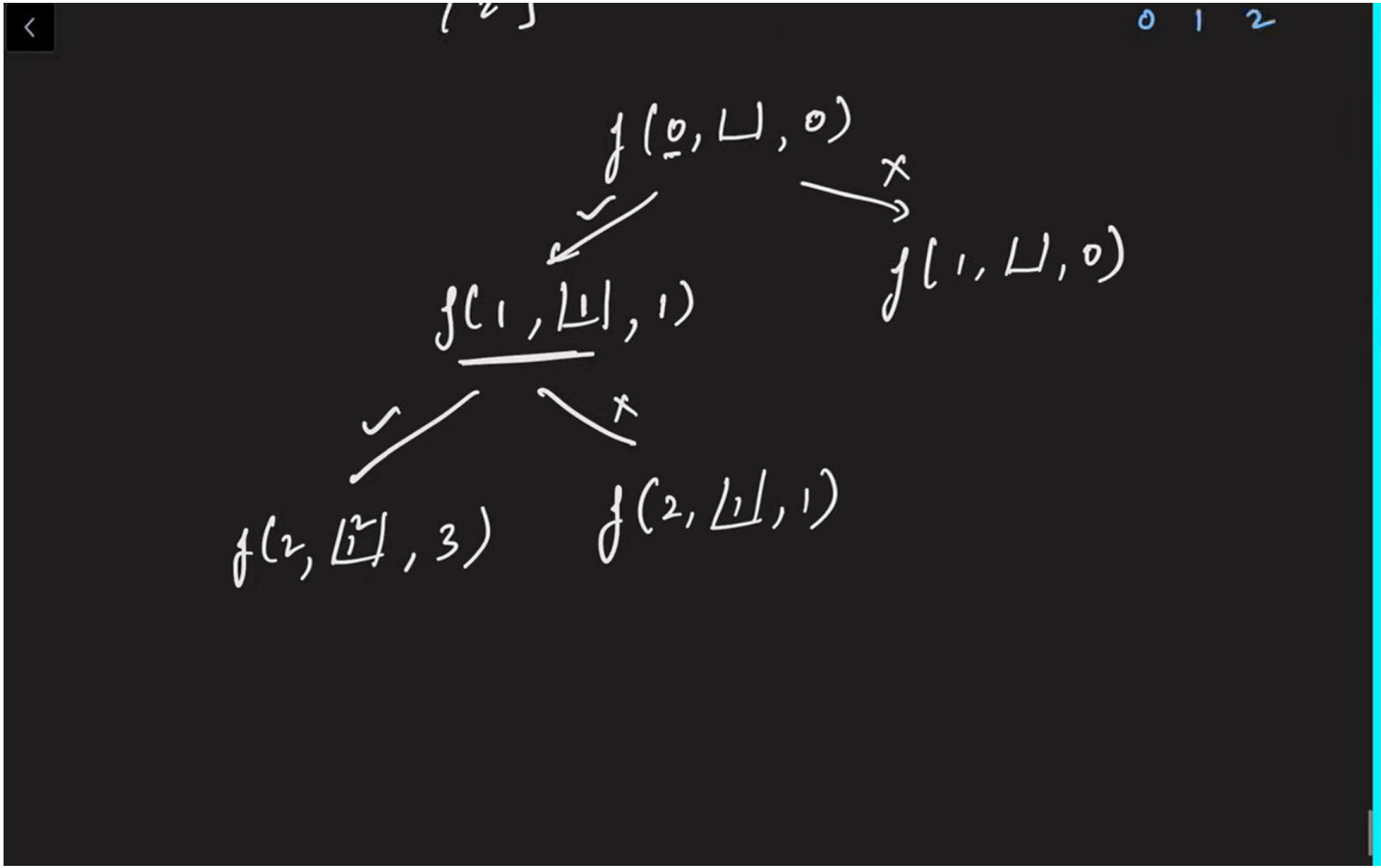
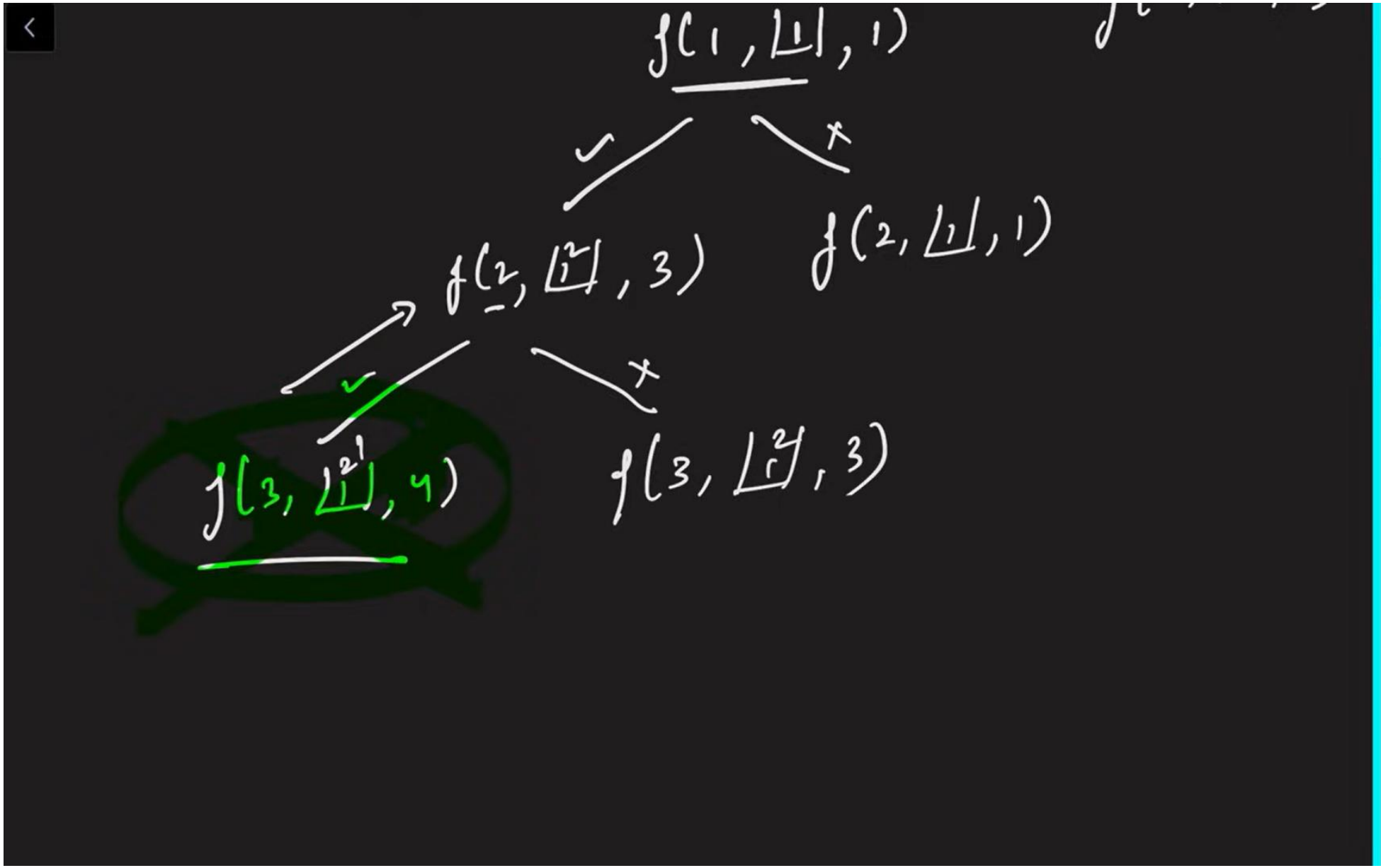
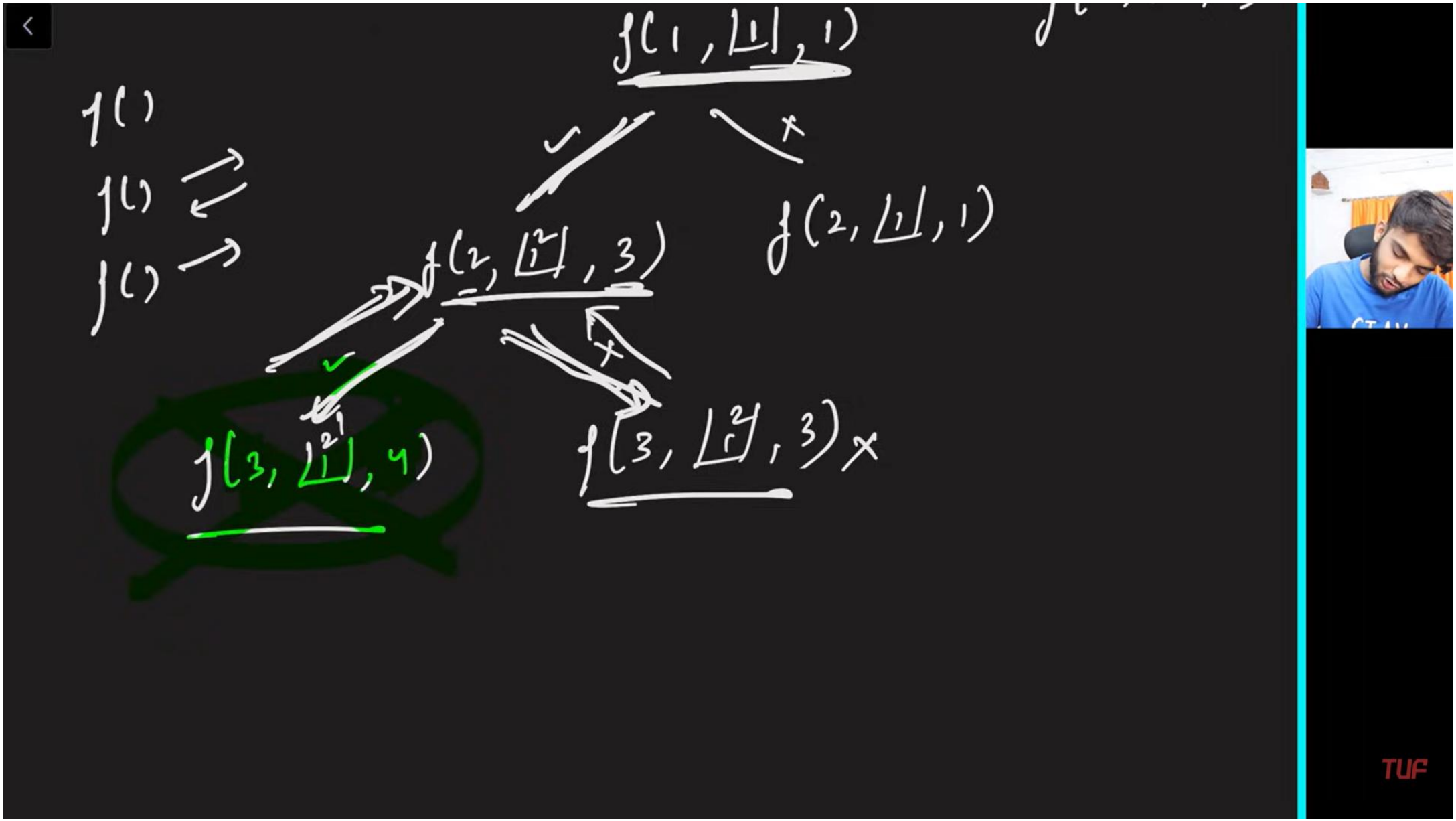
Printing Subsequences whose Sum is K

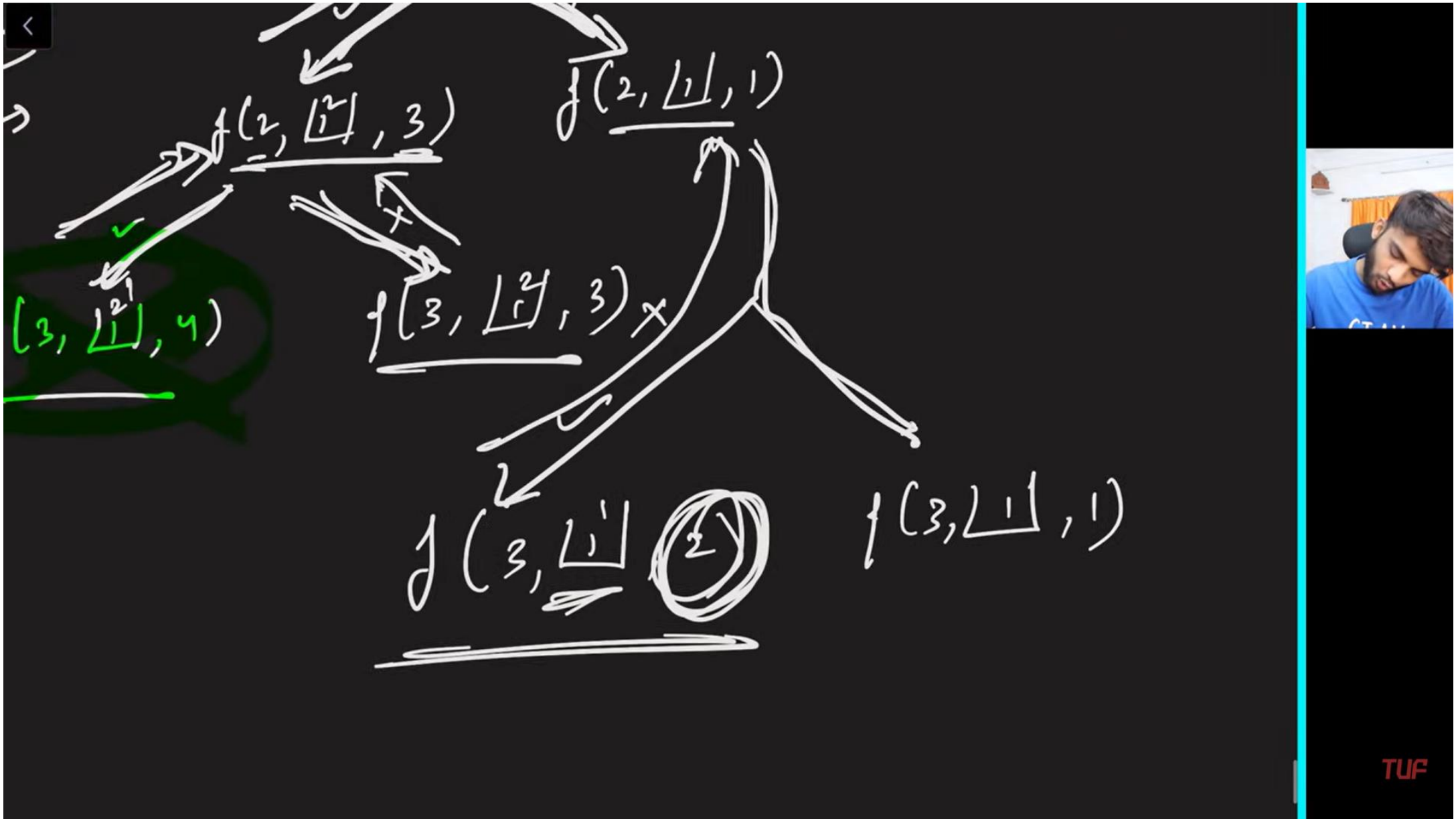
TUF

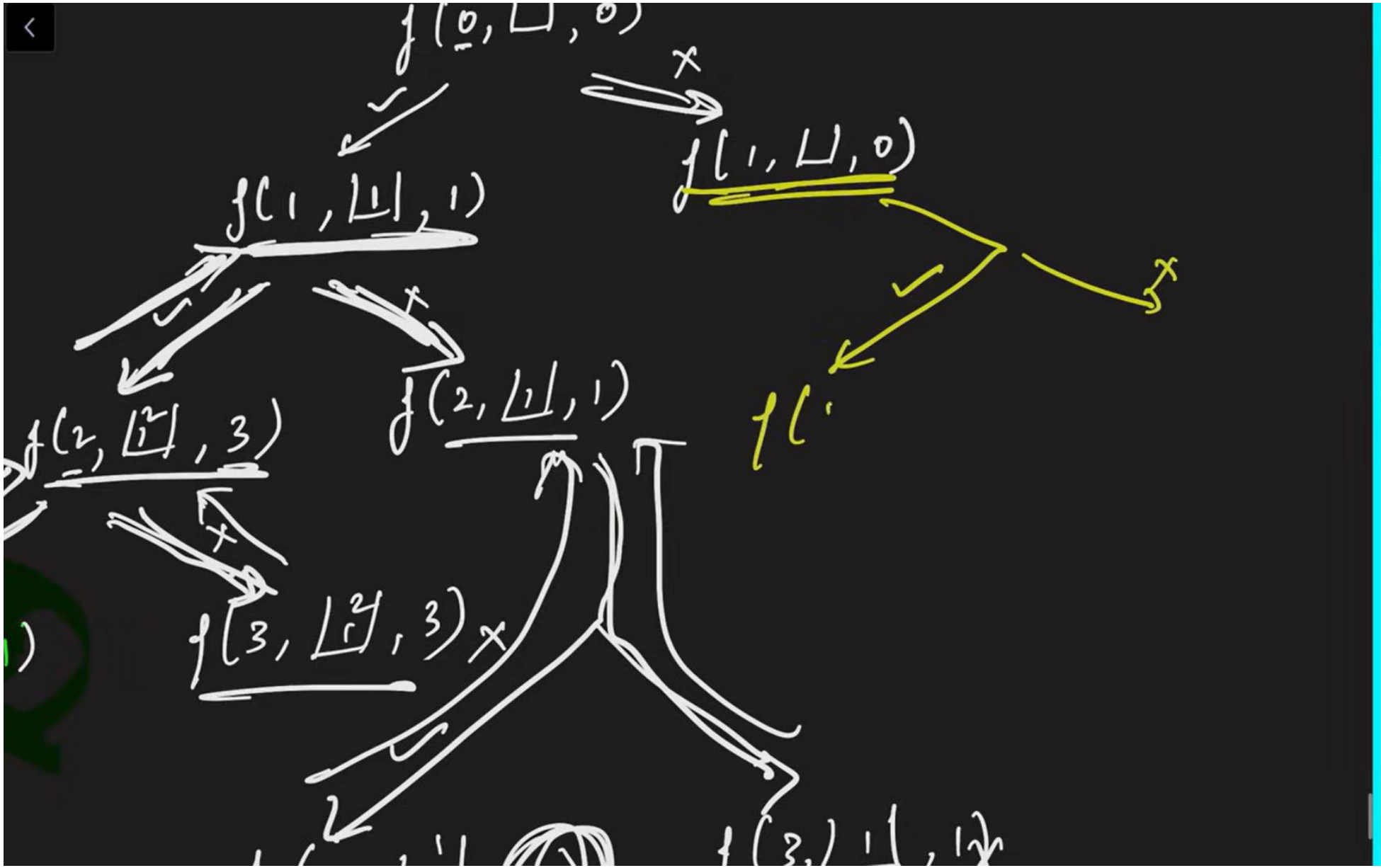

L7. All Kind of Patterns in Recursion | Print All | Print one | Count
arr → [1, 2, 1]
Sum = 2
[1, 1]
[2]
TUF
1:57 / 34:11 • Example ›

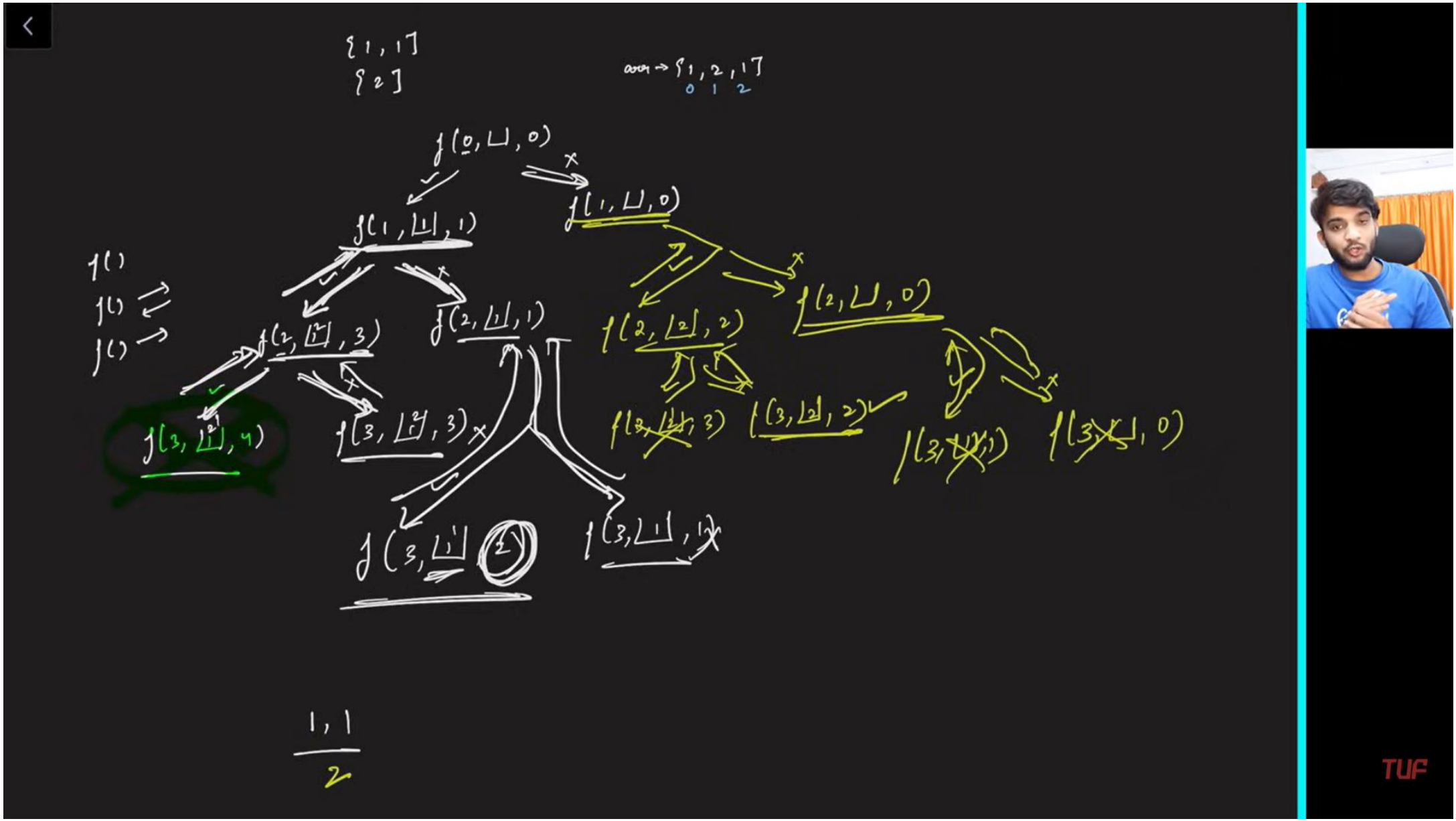

0 1 2
f(0, [1], 0)
f(1, [1], 1)
f(1, [1], 0)
f(2, [1,2], 3)
f(2, [1], 1)

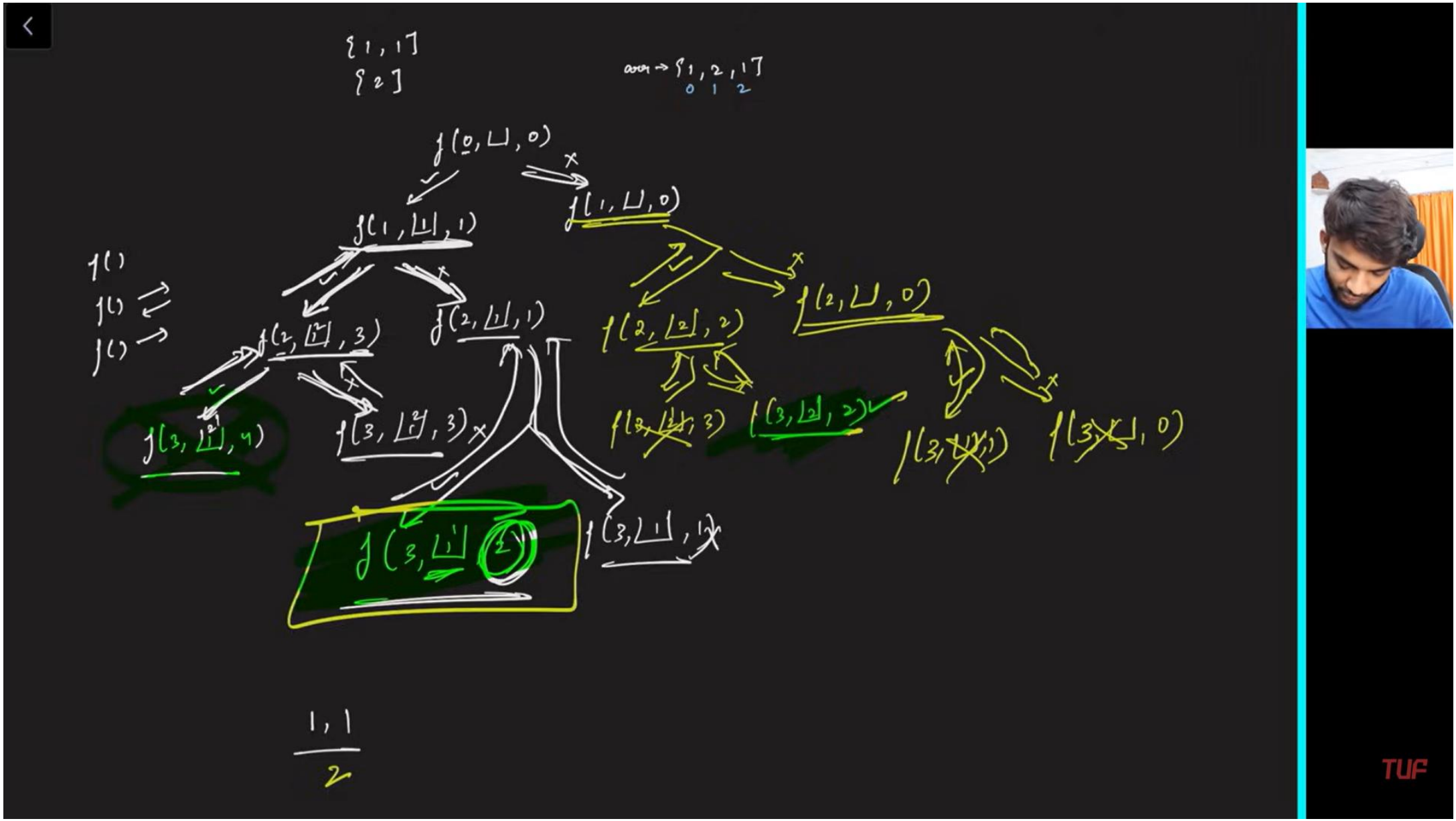

f(1, [1], 1)
=
x
f(2, [1], 1)
f(2, [1,2], 3)
=
x
f(3, [1,2,1], 4)
f(3, [1,2], 3)

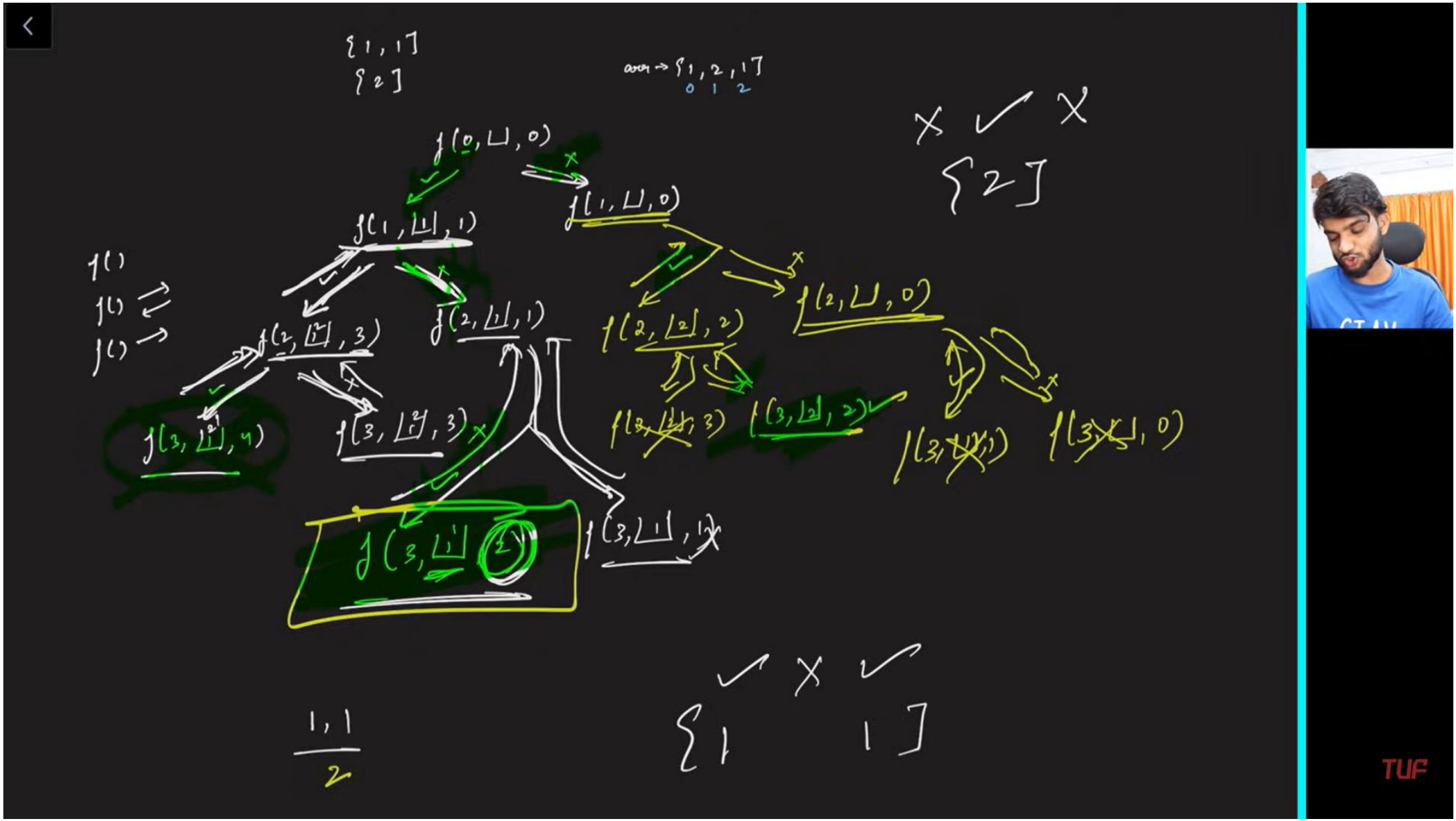

f(1, [1], 1)
f(2, [1,2], 3)
f(2, [1], 1)
f(3, [1,2,1], 4)
f(3, [1,2], 3) x
x
TUF


f(2, 12, 3)
f(2, 11, 1)
f(3, 12, 4)
f(3, 12, 3)
f(3, 11, 2)
f(3, 11, 1)

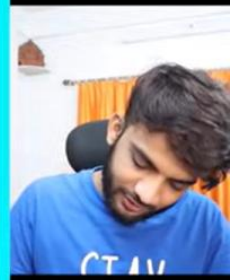

f(0, □, 0)
x
f(1, □, 0)
f(1, L1, 1)
f(2, L1, 1)
f(2, L2, 3)
f(3, L2, 3)
TUF


{1,1}
{2}
arr → {1,2,1}
0 1 2
f(0,[],0)
f(1,[1],1)
f(1,[],0)
f(2,[1,2],3)
f(2,[1],1)
f(2,[2],2)
f(2,[],0)
f(3,[1,2,1],4)
f(3,[1,2],3)
f(3,[2],2)
f(3,[],0)
f(3,[1],1)
1,1
2


{1,1}
{2}
arr → {1,2,1}
0 1 2
f(0,⊔,0)
f(1,⊔,1)
f(1,⊔,0)
f(2,⊔,3)
f(2,⊔,1)
f(2,⊔,2)
f(2,⊔,0)
f(3,⊔,4)
f(3,⊔,3)
f(3,⊔,2)
f(3,⊔,1)
f(3,⊔,0)
f(3,⊔,1)
1,1
2


{1,1}
{2}
arr → {1,2,1}
0 1 2
f(0,⊔,0)
f(1,⊔,1)
f(1,⊔,0)
f(2,⊔,3)
f(2,⊔,1)
f(2,⊔,2)
f(2,⊔,0)
f(3,⊔,4)
f(3,⊔,3)
f(3,⊔,2)
f(3,⊔,1)
f(3,⊔,0)
f(3,⊔,1)
{2}
{1 1}
1,1
2

```
f(i, ds, s)
{
  if (i == n)
  {
    if (s == sum)
      print(ds)
    return ;
  }
  ds.add(arr[i]);
  s += arr[i]
  f(i+1, 
```



TUF

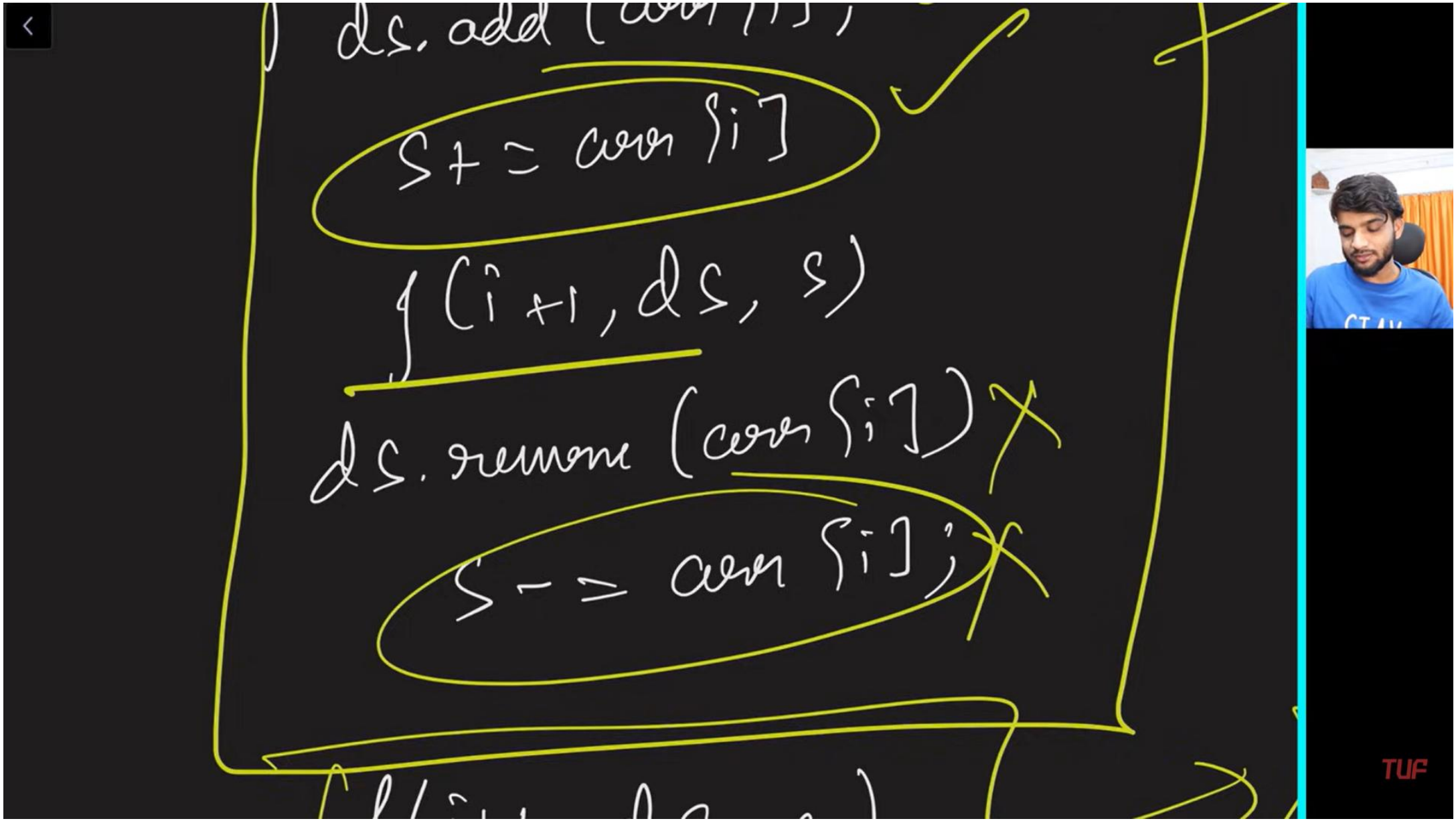
```
ds.add(arr[i]);
s += arr[i]
f(i+1, ds, s)
ds.remove(arr[i])
s -= arr[i];
```

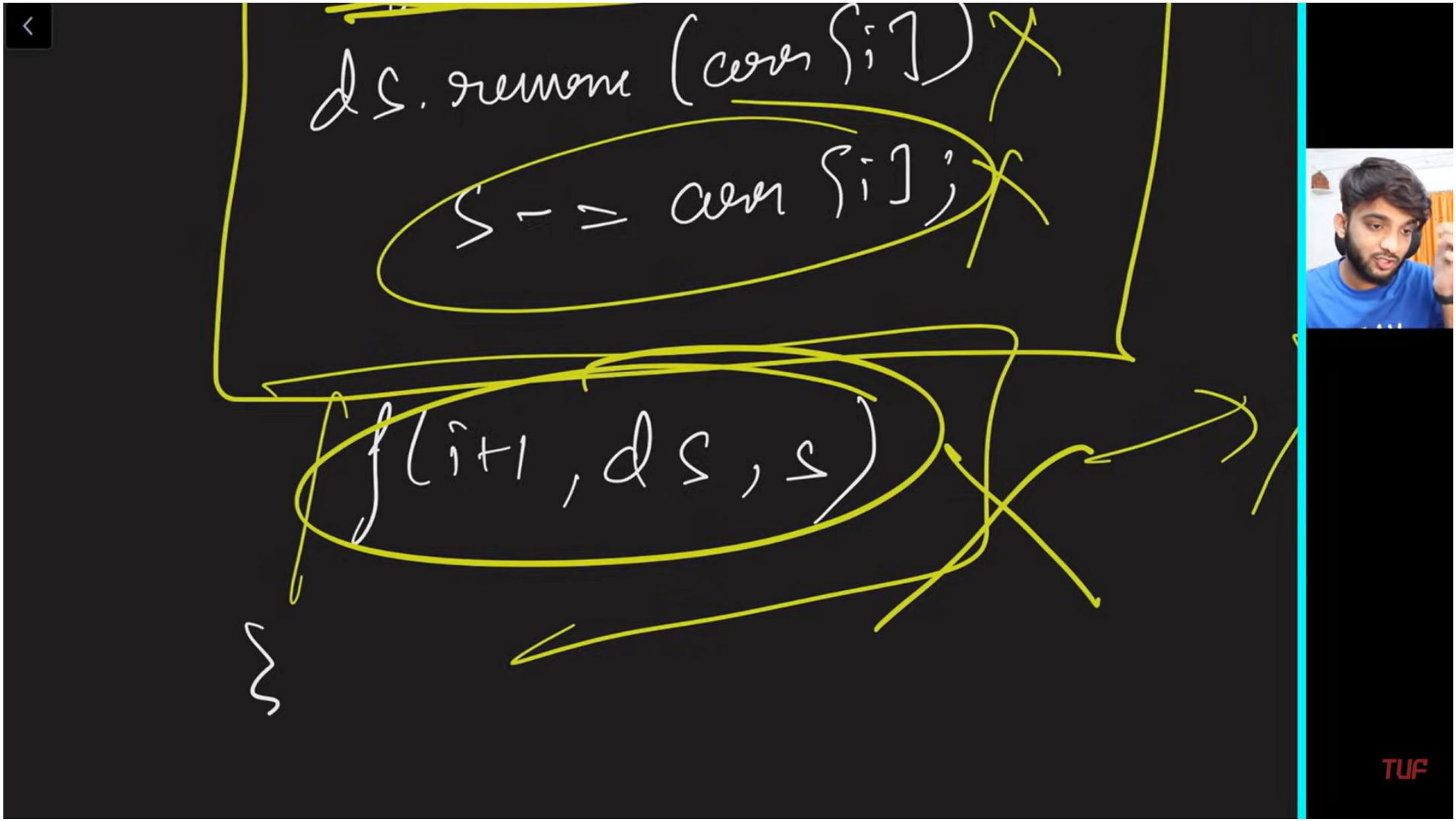
ds

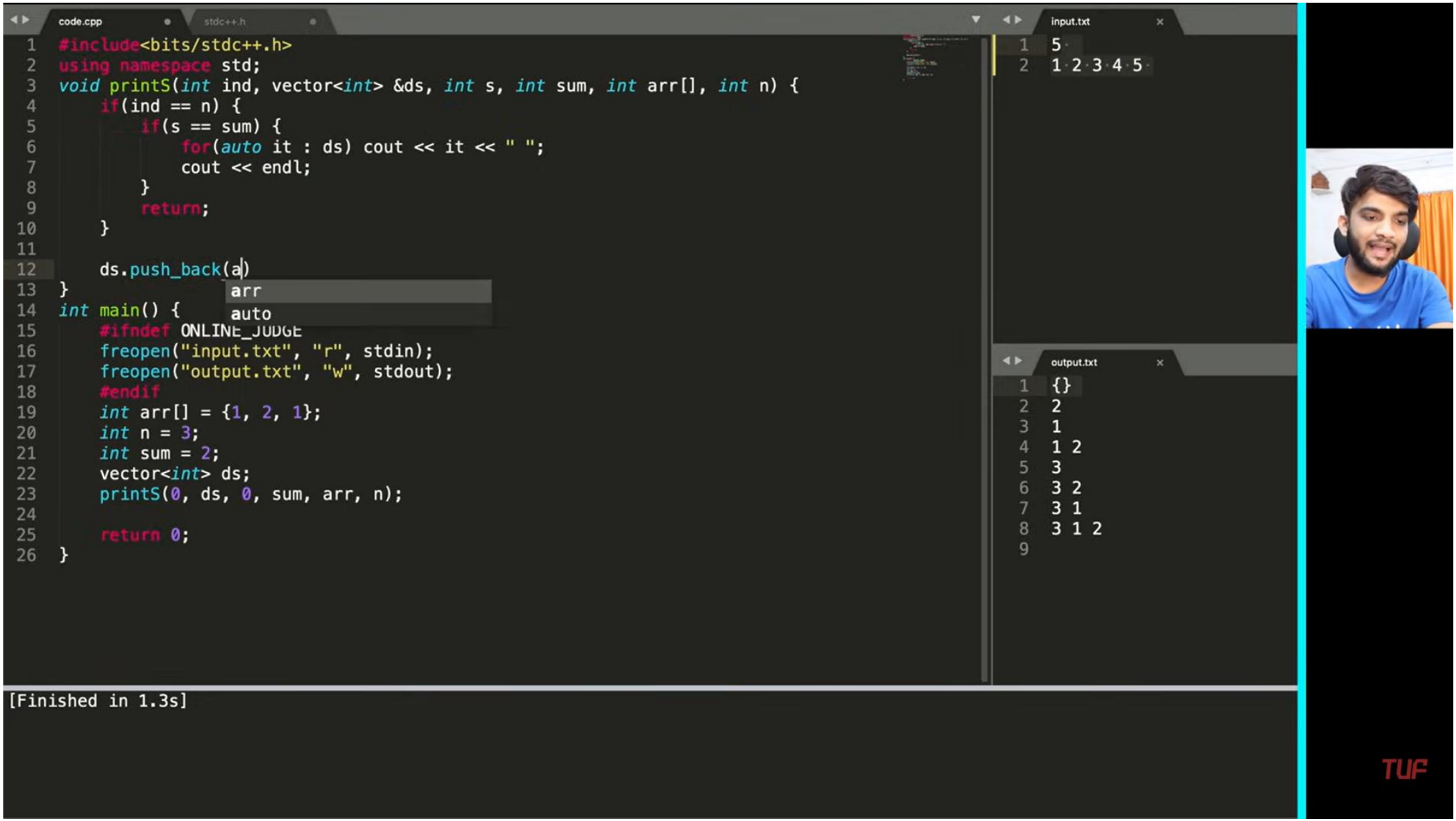
TUF

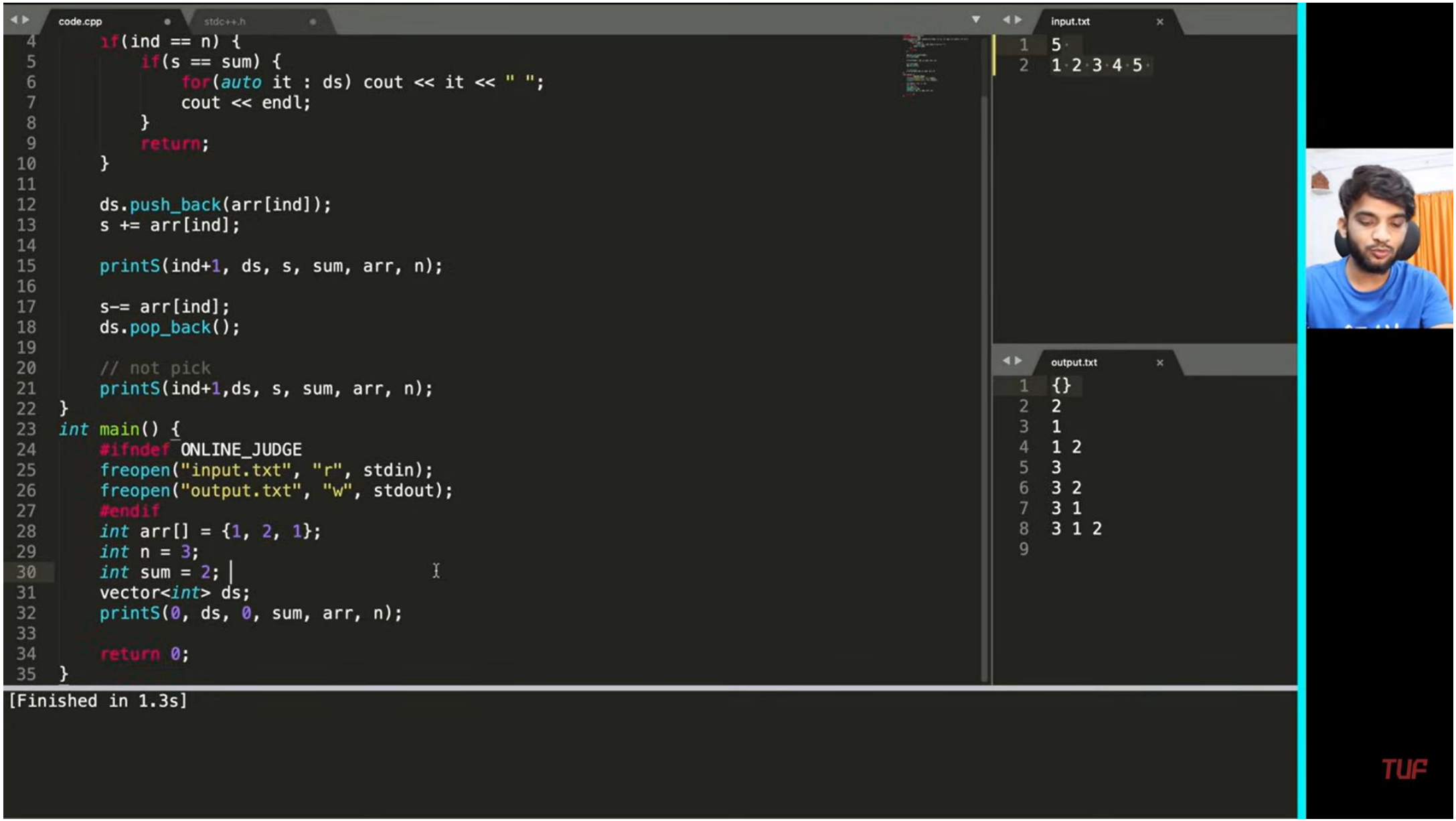
```
f(i, ds, s)
{
  if (i == n)
  {
    if (s == sum)
      print(ds)
    return;
  }
  ds.add(arr[i]);
  s += arr[i]
  f(i+1, ds, s)
  ds.remove(arr[i])
  s -= arr[i];

  f(i+1, ds, s)
}
```

TUF


ds.add(
s+= arr[i]
f(i+1, ds, s)
ds.remove(arr[i])
s-= arr[i];


ds.remove(arr[i])
s += arr[i];
f(i+1, ds, s)
}


code.cpp
stdc++.h
#include<bits/stdc++.h>
using namespace std;
void printS(int ind, vector<int> &ds, int s, int sum, int arr[], int n) {
if(ind == n) {
if(s == sum) {
for(auto it : ds) cout << it << " ";
cout << endl;
}
return;
}
ds.push_back(a)
arr
auto
}
int main() {
#ifndef ONLINE_JUDGE
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
#endif
int arr[] = {1, 2, 1};
int n = 3;
int sum = 2;
vector<int> ds;
printS(0, ds, 0, sum, arr, n);
return 0;
}
input.txt
5
1 2 3 4 5
output.txt
{}
2
1
1 2
3
3 2
3 1
3 1 2
[Finished in 1.3s]
TUF


code.cpp
stdc++.h
if(ind == n) {
if(s == sum) {
for(auto it : ds) cout << it << " ";
cout << endl;
}
return;
}
ds.push_back(arr[ind]);
s += arr[ind];
printS(ind+1, ds, s, sum, arr, n);
s-= arr[ind];
ds.pop_back();
// not pick
printS(ind+1,ds, s, sum, arr, n);
}
int main() {
#ifndef ONLINE_JUDGE
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
#endif
int arr[] = {1, 2, 1};
int n = 3;
int sum = 2;
vector<int> ds;
printS(0, ds, 0, sum, arr, n);
return 0;
}
input.txt
5
1 2 3 4 5
output.txt
{}
2
1
1 2
3
3 2
3 1
3 1 2
[Finished in 1.3s]
TUF

code.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;
void printS(int ind, vector<int> &ds, int s, int sum, int arr[], int n) {
    if(ind == n) {
        if(s == sum) {
            for(auto it : ds) cout << it << " ";
            cout << endl;
        }
        return;
    }

    ds.push_back(arr[ind]);
    s += arr[ind];

    printS(ind+1, ds, s, sum, arr, n);

    s-= arr[ind];
    ds.pop_back();

    // not pick
    printS(ind+1,ds, s, sum, arr, n);
}
int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif
    int arr[] = {1, 2, 1};
    int n = 3;
    int sum = 2;
    vector<int> ds;
    printS(0, ds, 0, sum, arr, n);

    return 0;
}
```

input.txt

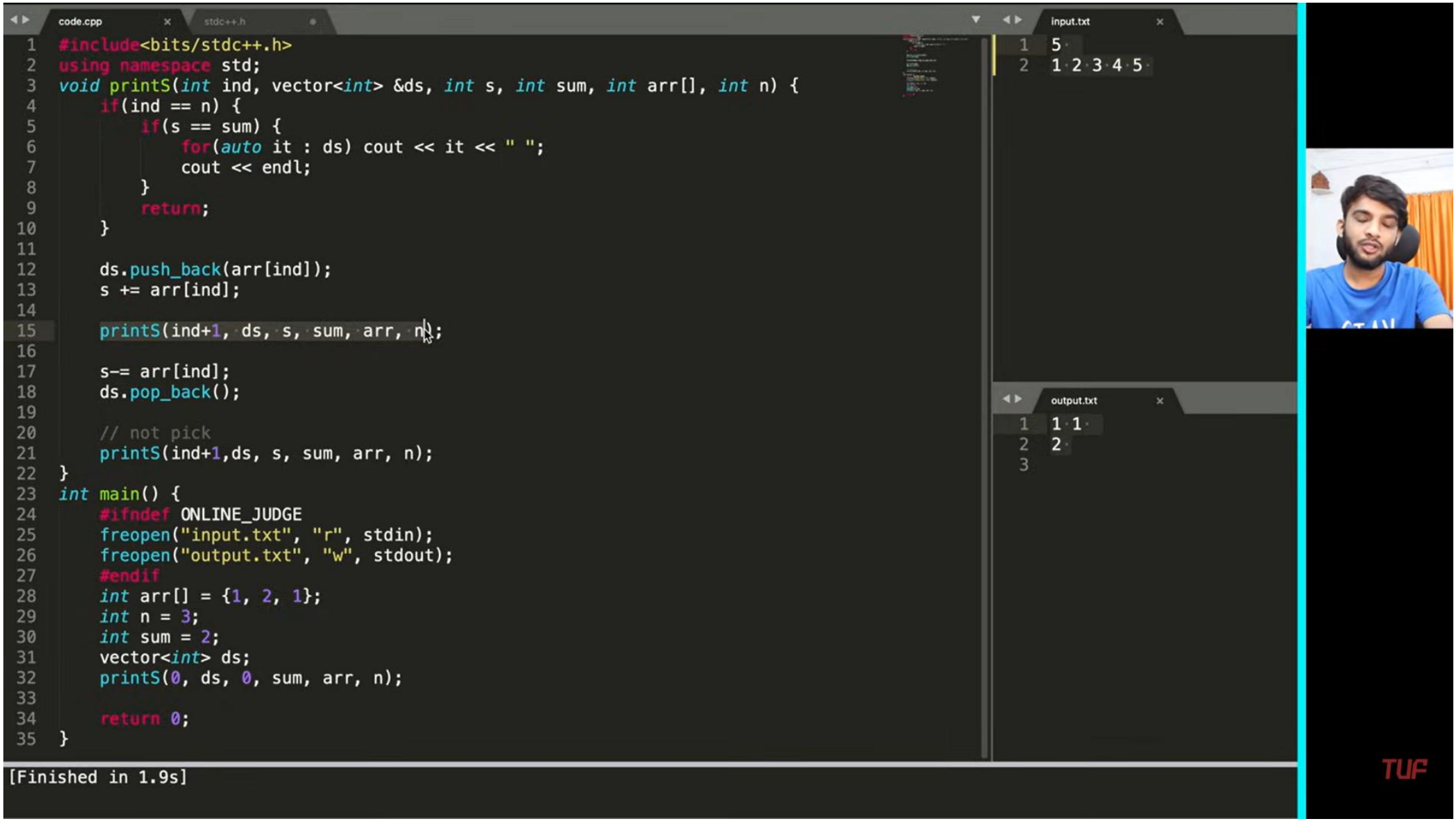
```
5
1 2 3 4 5
```

output.txt

```
1 1
2
```

```
[Finished in 1.9s]
```

TUF


code.cpp
stdc++.h
#include<bits/stdc++.h>
using namespace std;
void printS(int ind, vector<int> &ds, int s, int sum, int arr[], int n) {
if(ind == n) {
if(s == sum) {
for(auto it : ds) cout << it << " ";
cout << endl;
}
return;
}
ds.push_back(arr[ind]);
s += arr[ind];
printS(ind+1, ds, s, sum, arr, n);
s-= arr[ind];
ds.pop_back();
// not pick
printS(ind+1,ds, s, sum, arr, n);
}
int main() {
#ifndef ONLINE_JUDGE
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
#endif
int arr[] = {1, 2, 1};
int n = 3;
int sum = 2;
vector<int> ds;
printS(0, ds, 0, sum, arr, n);
return 0;
}
input.txt
5
1 2 3 4 5
output.txt
1 1
2
[Finished in 1.9s]
TUF

code.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;

bool flag = false;
void printS(int ind, vector<int> &ds, int s, int sum, int arr[], int n) {
    if(ind == n) {
        if(s == sum && flag == false) {
            flag = true;
            for(auto it : ds) cout << it << " ";
            cout << endl;
        }
        return;
    }

    ds.push_back(arr[ind]);
    s += arr[ind];

    printS(ind+1, ds, s, sum, arr, n);

    s-= arr[ind];
    ds.pop_back();

    // not pick
    printS(ind+1,ds, s, sum, arr, n);
}
int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif
    int arr[] = {1, 2, 1};
    int n = 3;
    int sum = 2;
    vector<int> ds;
    printS(0, ds, 0, sum, arr, n);
```

input.txt

```
5
1 2 3 4 5
```

output.txt

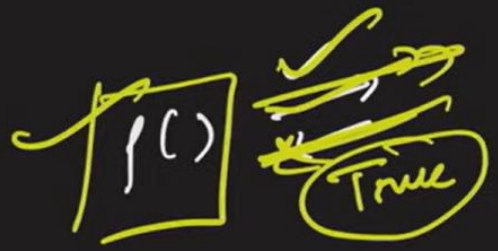
```
1 1
```

```
[Finished in 1.9s]
```

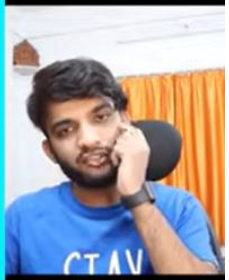
TUF

Print any 1 subsequence where Sm is Sum.

```
f()
{
```




```
}
```

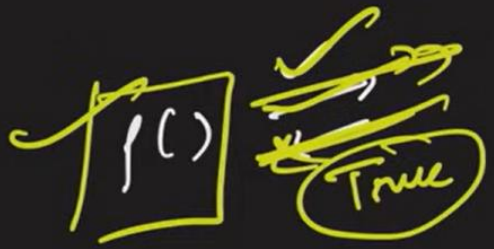


TUF

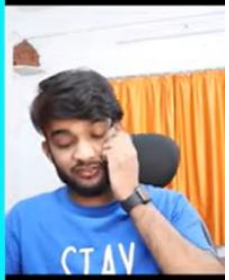
Subsequence where Sm is Sum.

(Technique to print One answer)

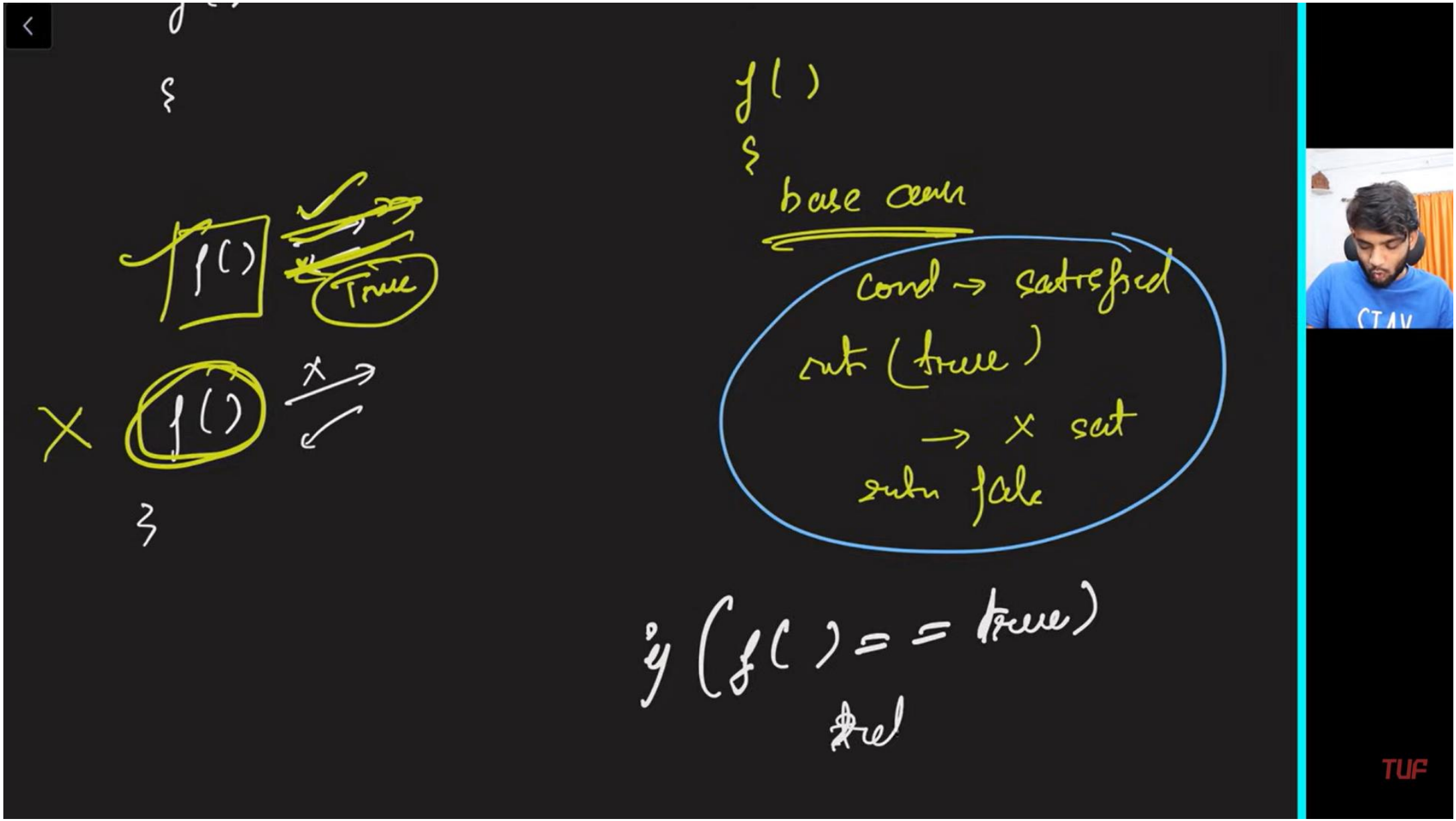
```
f()
{
```




```
}
```



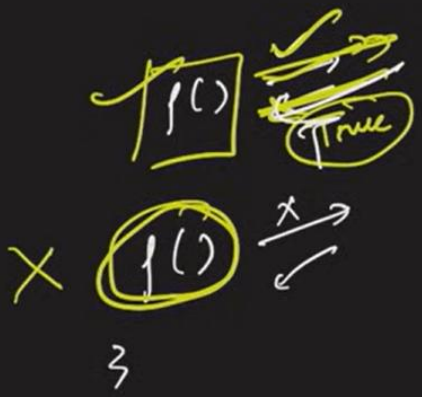
TUF


f()
{
f()
True
X
f()
X
}
f()
{
base cases
cond → satisfied
ret (true)
→ X ret
ret false
if (f() == true)
#el

any subsequence where sum is sum.

(Technique to print one answer)

```
f()
{
}
```




```
f()
{
  base case
}
```

cond → satisfied
out (true)
→ x out
outer false

```
if (f() == true)
    return;
```

f()

TUF

code.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;


bool printS(int ind, vector<int> &ds, int s, int sum, int arr[], int n) {
    if(ind == n) {
        // condition satisfied
        if(s == sum) {
            for(auto it : ds) cout << it << " ";
            cout << endl;
            return true;
        }
        // condition not satisfied
        else return false;
    }

    ds.push_back(arr[ind]);
    s += arr[ind];

    if(printS(ind+1, ds, s, sum, arr, n) == true) {
        return true;
    }

    s-= arr[ind];
    ds.pop_back();

    // not pick
    if(printS(ind+1,ds, s, sum, arr, n) == true) return true;

    return false;
}
int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif
```

input.txt

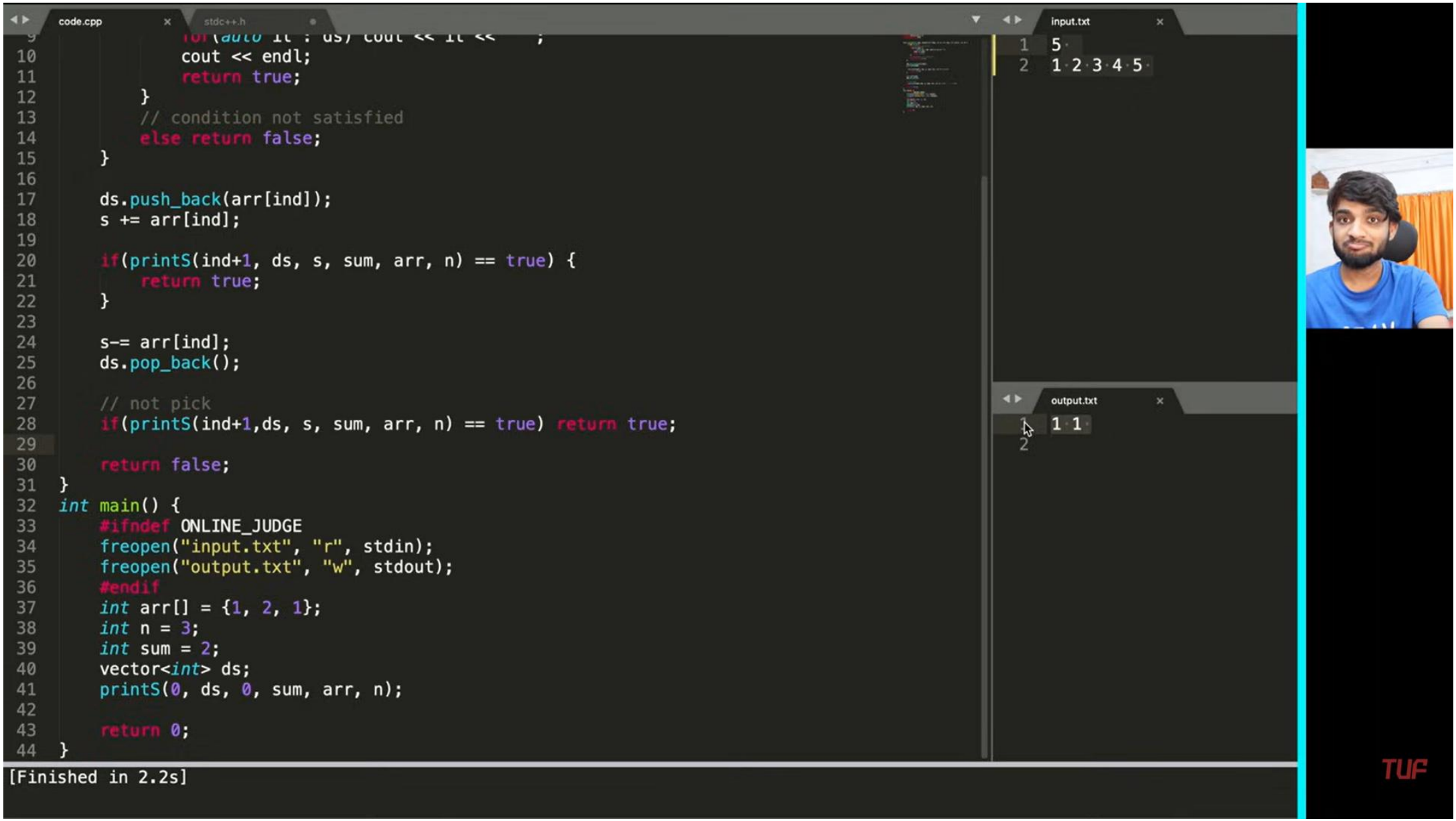
```
5
1 2 3 4 5
```

output.txt

```
1 1

```

```
[Finished in 1.9s]
```

TUF


code.cpp
stdc++.h
cout << endl;
return true;
}
// condition not satisfied
else return false;
}
ds.push_back(arr[ind]);
s += arr[ind];
if(printS(ind+1, ds, s, sum, arr, n) == true) {
return true;
}
s-= arr[ind];
ds.pop_back();
// not pick
if(printS(ind+1,ds, s, sum, arr, n) == true) return true;
return false;
}
int main() {
#ifndef ONLINE_JUDGE
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
#endif
int arr[] = {1, 2, 1};
int n = 3;
int sum = 2;
vector<int> ds;
printS(0, ds, 0, sum, arr, n);
return 0;
}
input.txt
5
1 2 3 4 5
output.txt
1 1
[Finished in 2.2s]
TUF

Sm = 2 [1, 2, 1]

f(0, LI, 0)
{
ds.add()
if (s+= f(1, LI, 1))
ds.remove
s -=
f(8, LI, 0))

Sm = 2 [1, 2, 1]

f(1, [1], 1)

i++ (1) x

ds.add

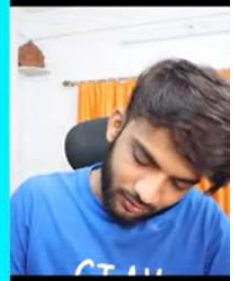
i++ f(2, [1, 2], 3)

f(2, [1, 2], 3)

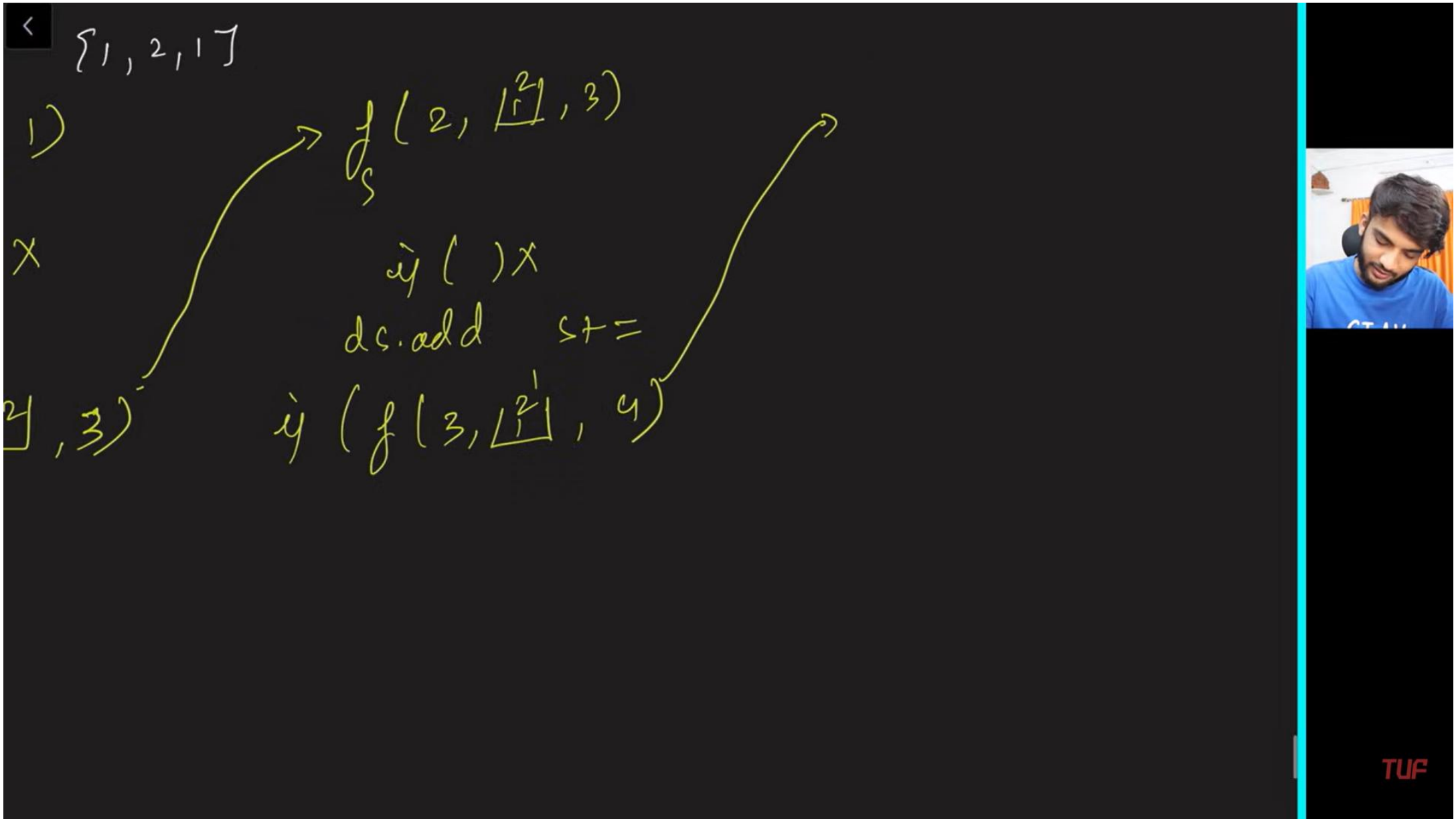
i++ ( ) x

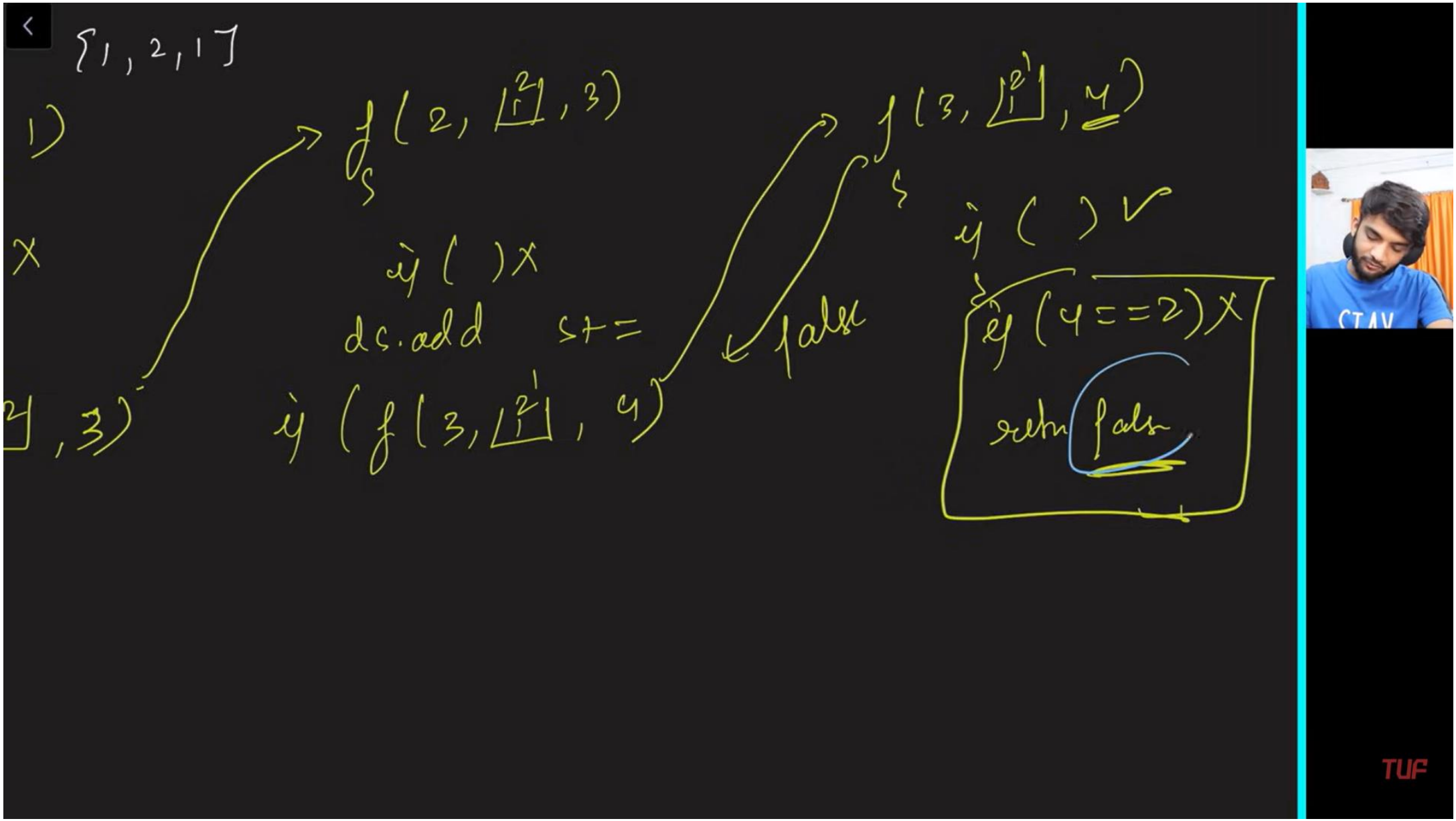
ds.add

i++ (f(



TUF


[1, 2, 1]
1)
x
4, 3)
f(2, [2], 3)
if ( )x
ds.add
st+=
if (f(3, [2], 4)

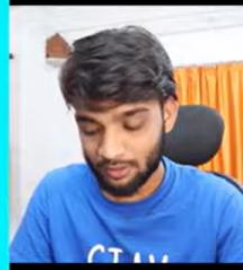

[1, 2, 1]
1)
x
2, 3)
f(2, 2, 3)
if ( )x
ds.add
s+=
if (f(3, 2, 4)
false
f(3, 2, 4)
if ( )✓
if (4==2)x
retu false

L7. All Kind of Patterns in Recursion | Print All | Print one | Count

→ Count the subsequences with sum = k

[1, 2, 1] k = 2

(2 subsequences)



TUF

23:57 / 34:11 • Modify ›

L7. All Kind of Patterns in Recursion | Print All | Print one | Count

→ Count the subsequences with sum = k

[1, 2, 1] k = 2

(2 subsequences)

TUF

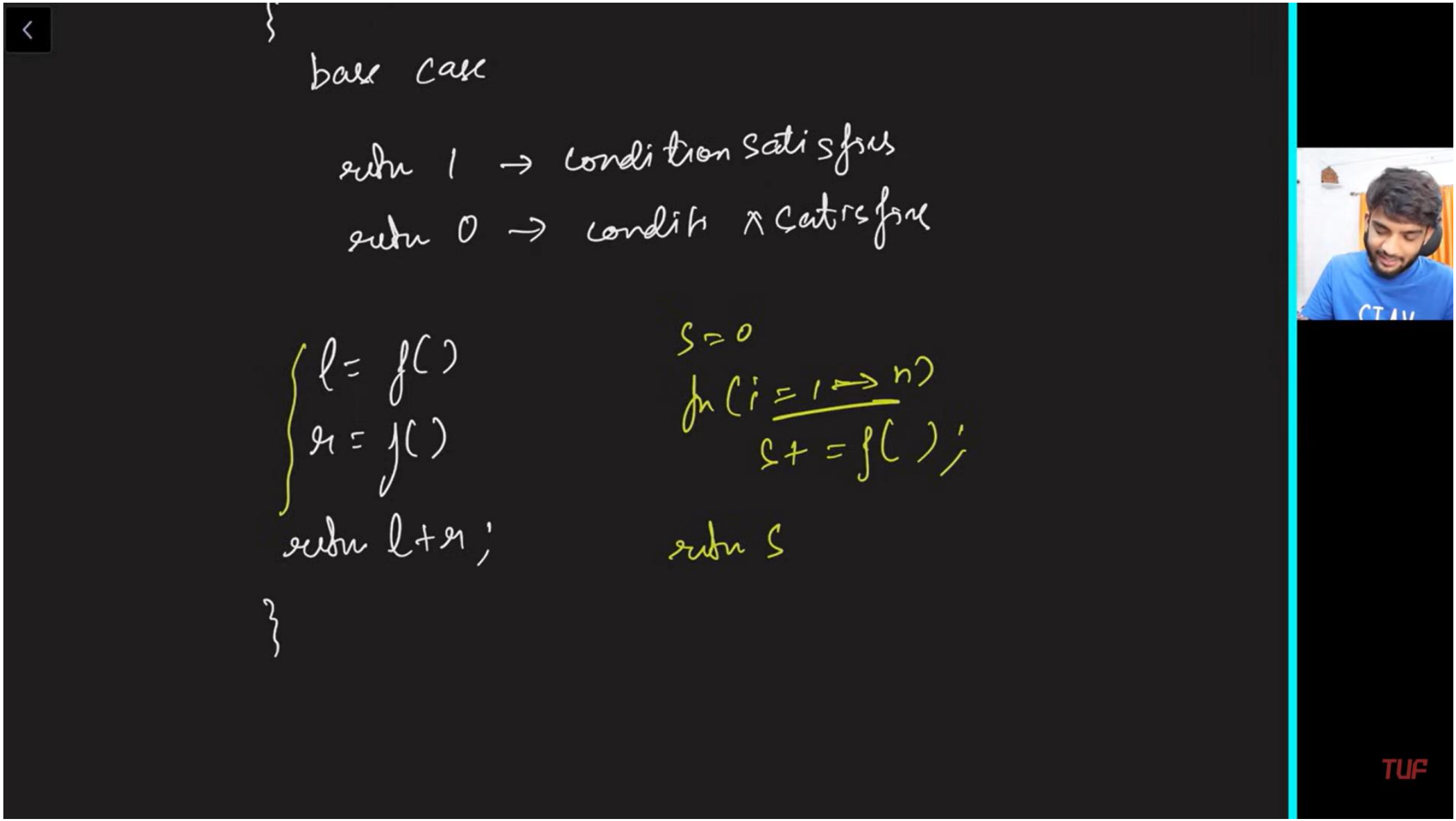
23:52 / 34:11 • Modify ›

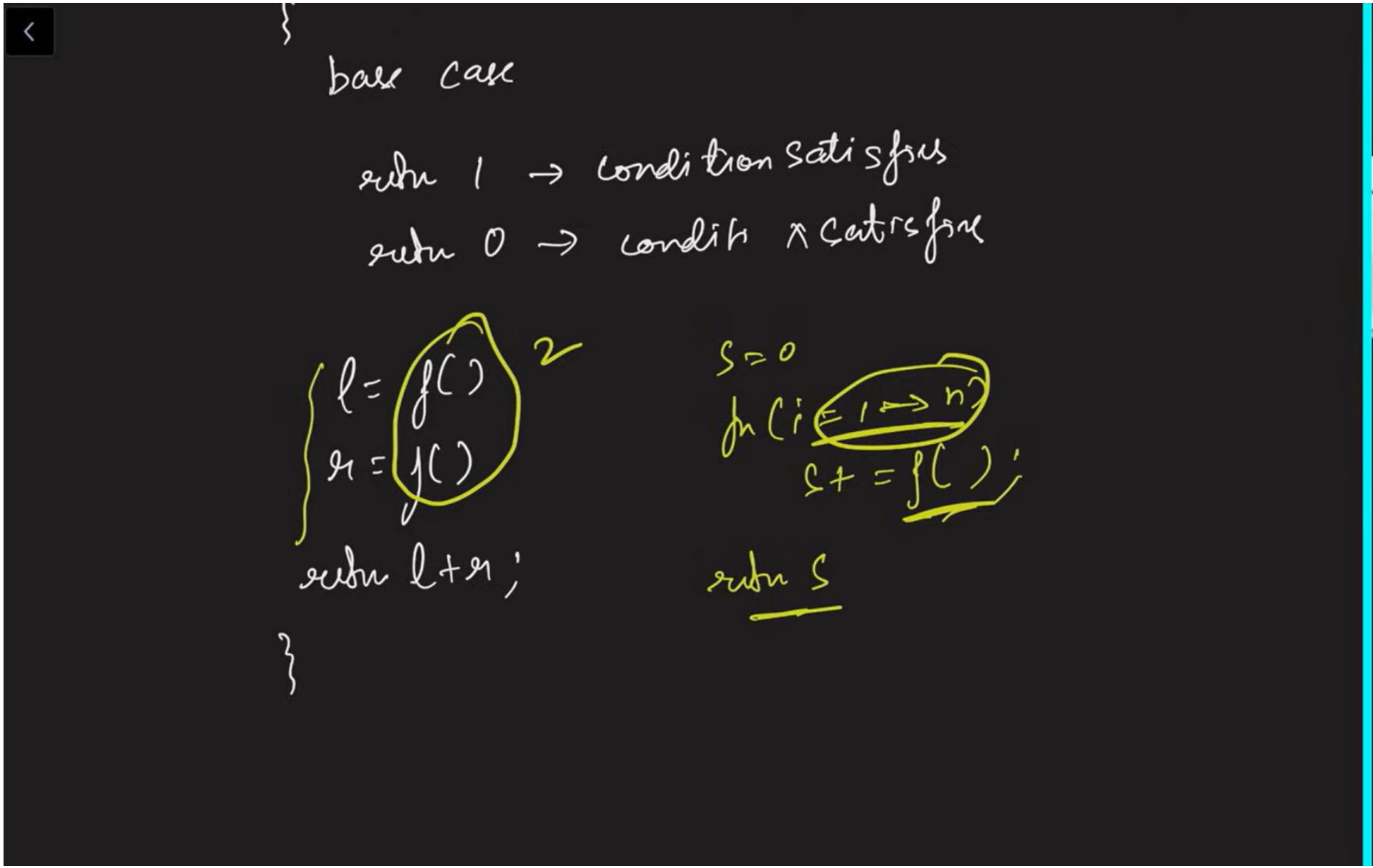
```
int f()
{
    base case

    return 1 → condition satisfies
    return 0 → condition not satisfies

    l = f()
    r = f()
    return l+r;
}
```

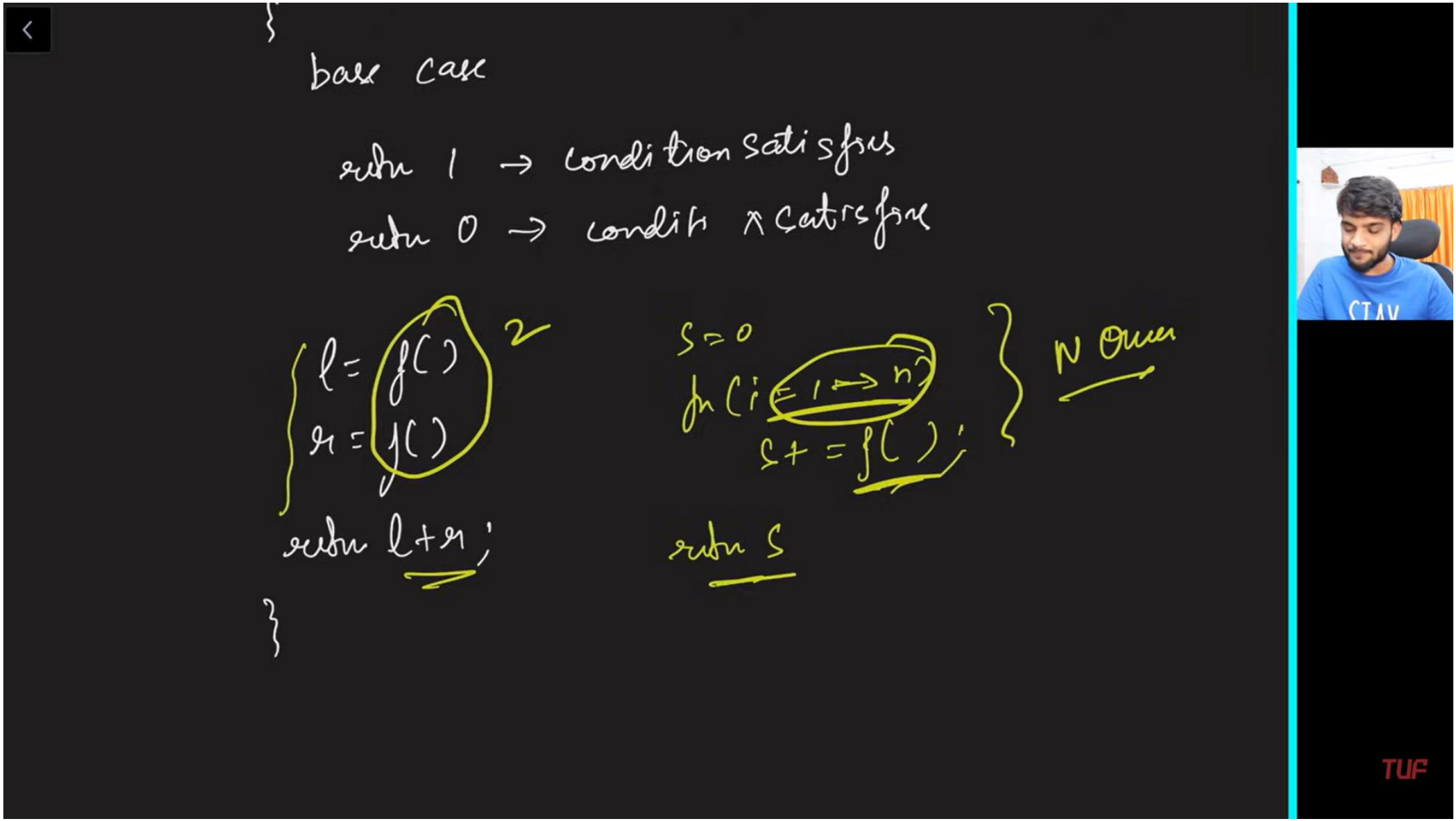
TUF

```
int f()
{
    base case

    return 1 -> condition satisfies
    return 0 -> condition not satisfies

    l = f()
    r = f()
    return l+r;
}
```

TUF


base case
return 1 → condition satisfies
return 0 → condition not satisfies
l = f()
r = f()
return l+r;
s = 0
for(i = 1 → n)
s += f();
return s


base case
return 1 → condition satisfies
return 0 → condition not satisfies
l = f()
r = f()
return l+r;
2
s = 0
for(i = 1 → n)
s += f();
return s


base case
return 1 → condition satisfies
return 0 → condition not satisfies
l = f()
r = f()
2
return l+r;
s = 0
for(i = 1 → n)
s += f();
N Queen
return s

code.cpp

```cpp
int printS(int ind, vector<int> &ds, int s, int sum, int arr[], int n) {
    if(ind == n) {
        // condition satisfied
        if(s == sum) {
            return 1;
        }
        // condition not satisfied
        else return 0;
    }

    ds.push_back(arr[ind]);
    s += arr[ind];

    int l = printS(ind+1, ds, s, sum, arr, n) ;

    s-= arr[ind];
    ds.pop_back();

    // not pick
    int r = printS(ind+1,ds, s, sum, arr, n);

    return l + r;
}
int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif
    int arr[] = {1, 2, 1};
    int n = 3;
    int sum = 2;
    vector<int> ds;
    printS(0, ds, 0, sum, arr, n);

    return 0;
```

input.txt

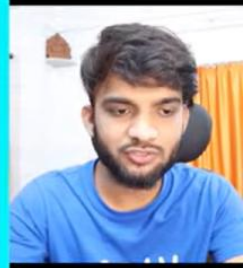
```
5
1 2 3 4 5
```

output.txt

```
1 1
```

```
[Finished in 2.2s]
```

Line 8, Column 23 Tab Size: 4 C++



TUF

code.cpp

```cpp
using namespace std;


int printS(int ind, int s, int sum, int arr[], int n) {
    if(ind == n) {
        // condition satisfied
        if(s == sum) return 1;
        // condition not satisfied
        else return 0;
    }


    s += arr[ind];

    int l = printS(ind+1,  s, sum, arr, n) ;

    s-= arr[ind];


    // not pick
    int r = printS(ind+1, s, sum, arr, n);

    return l + r;
}
int main() {
    #ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif
    int arr[] = {1, 2, 1};
    int n = 3;
    int sum = 2;
    cout << printS(0, 0, sum, arr, n);

    return 0;
}
```

input.txt

```
5
1 2 3 4 5
```

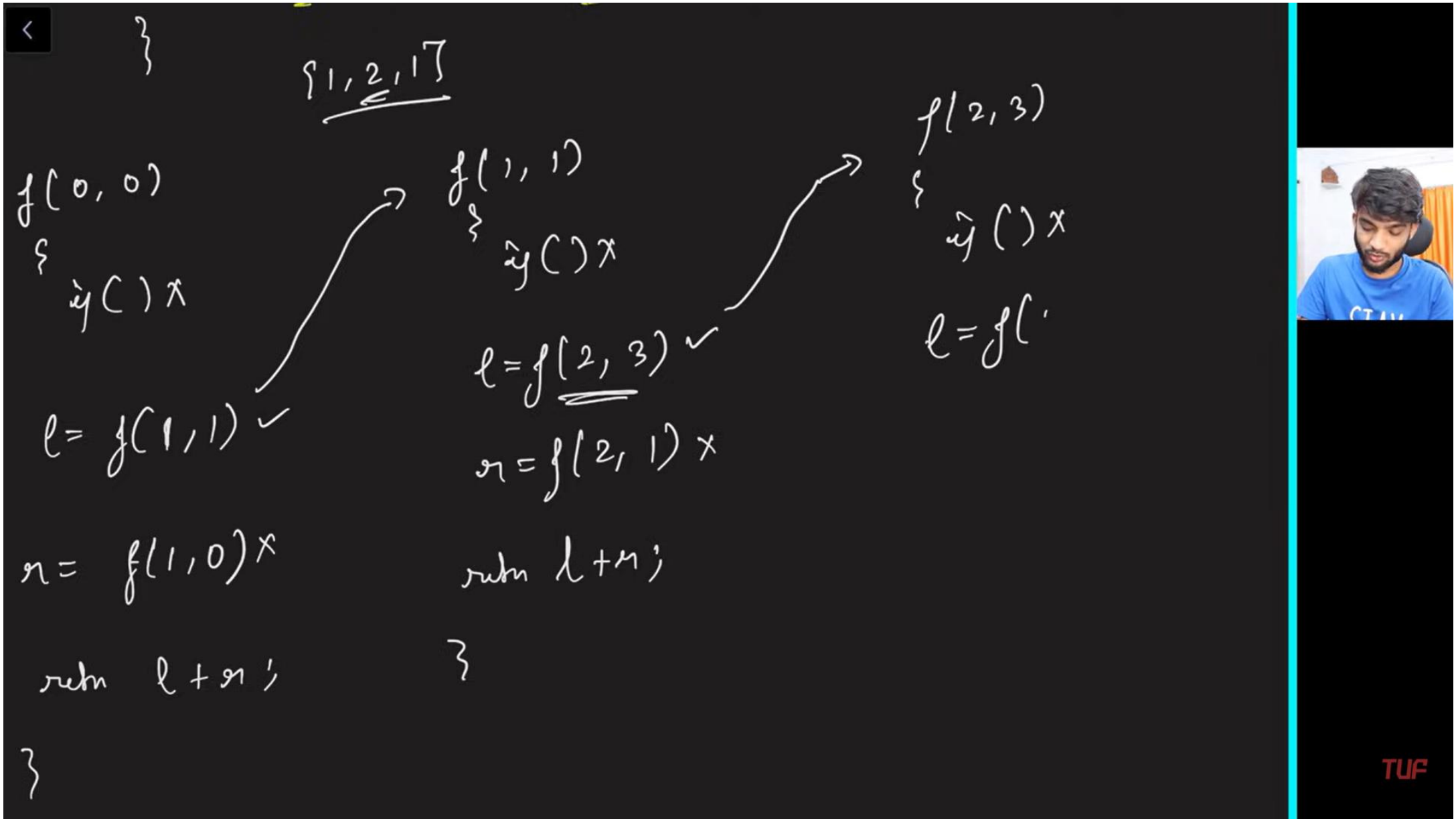
output.txt

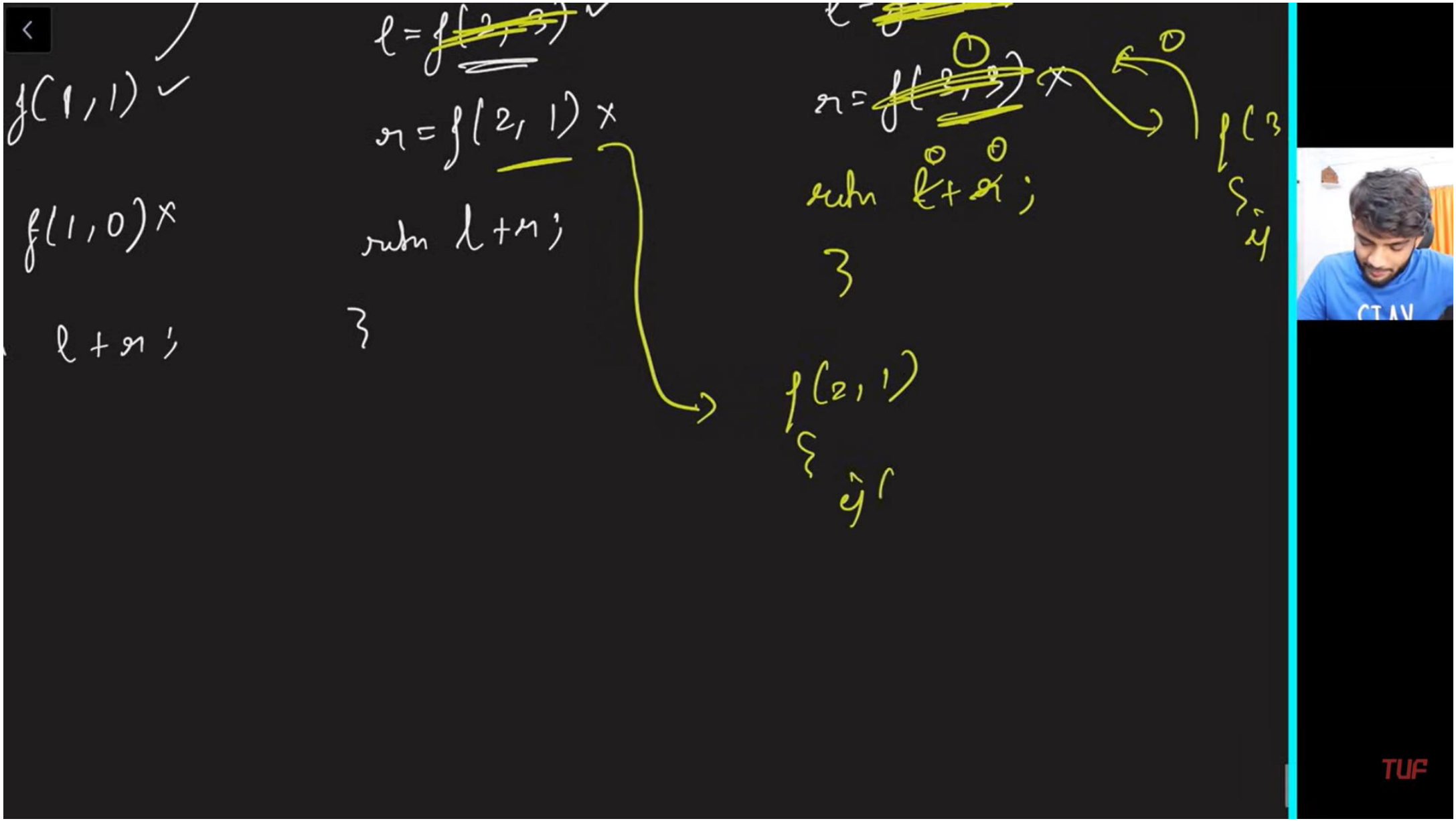
```
2
```

```
[Finished in 1.5s]
```

Line 19, Column 5; Reloading ~/Desktop/SublimeFolder/output.txt Tab Size: 4 C++

TUF


}
[1, 2, 1]
f(0, 0)
{
ij()x
l = f(1, 1)
r = f(1, 0)x
return l + r;
}
f(1, 1)
{
ij()x
l = f(2, 3)
r = f(2, 1)x
return l + r;
}
f(2, 3)
{
ij()x
l = f(


f(1,1)
f(1,0)
l+r;
l=f(2,2)
r=f(2,1)
return l+r;
}
f(2,1)
{
r=f(2,2)
return l+r;
}
f(3
{
y

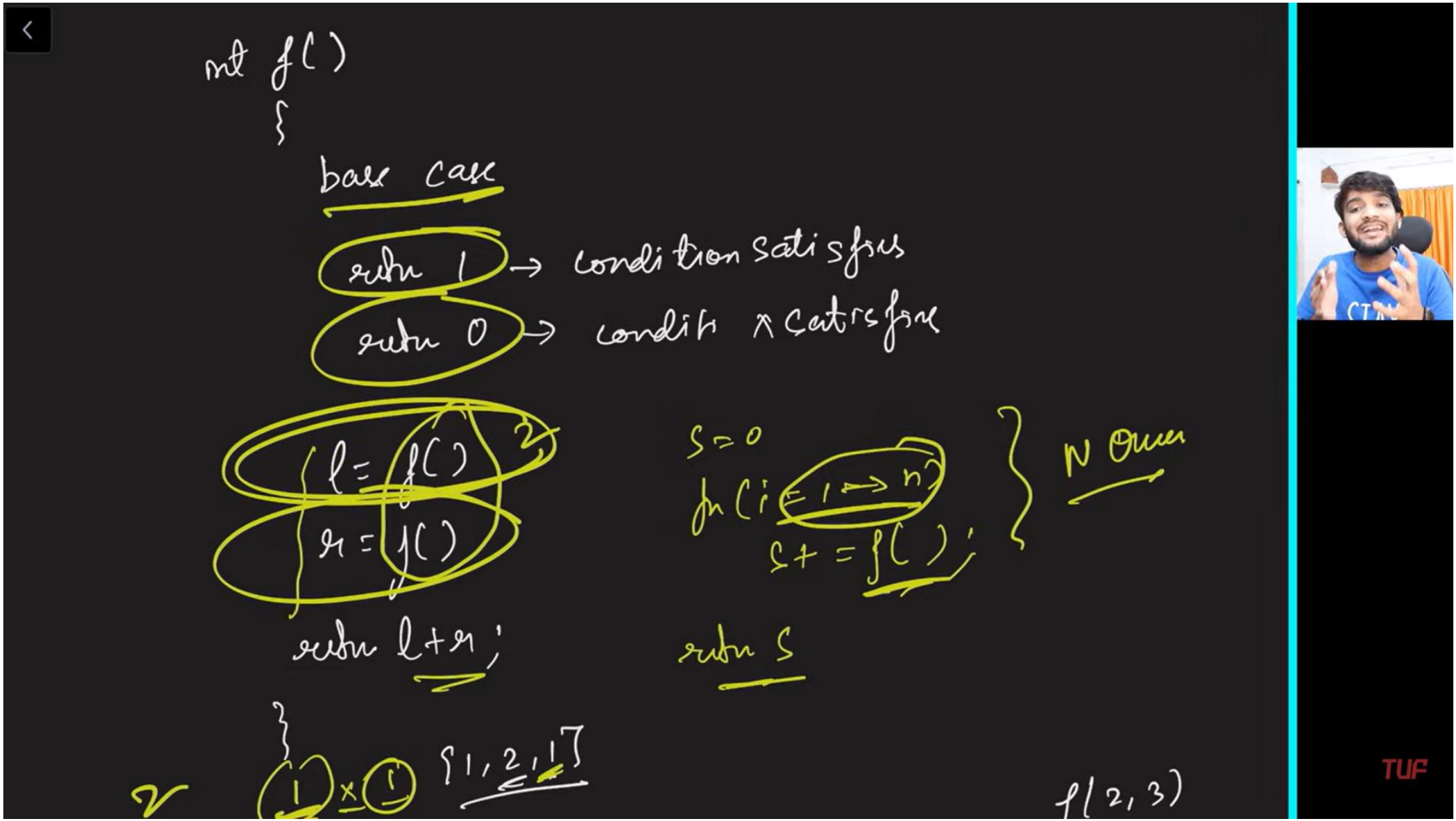
```
int f()
{
    base case
    return 1 -> condition satisfies
    return 0 -> condition not satisfies

    l = f()
    r = f()
    return l+r;

    s = 0
    for(i = 1 -> n)
        s += f();
    return s
}
```

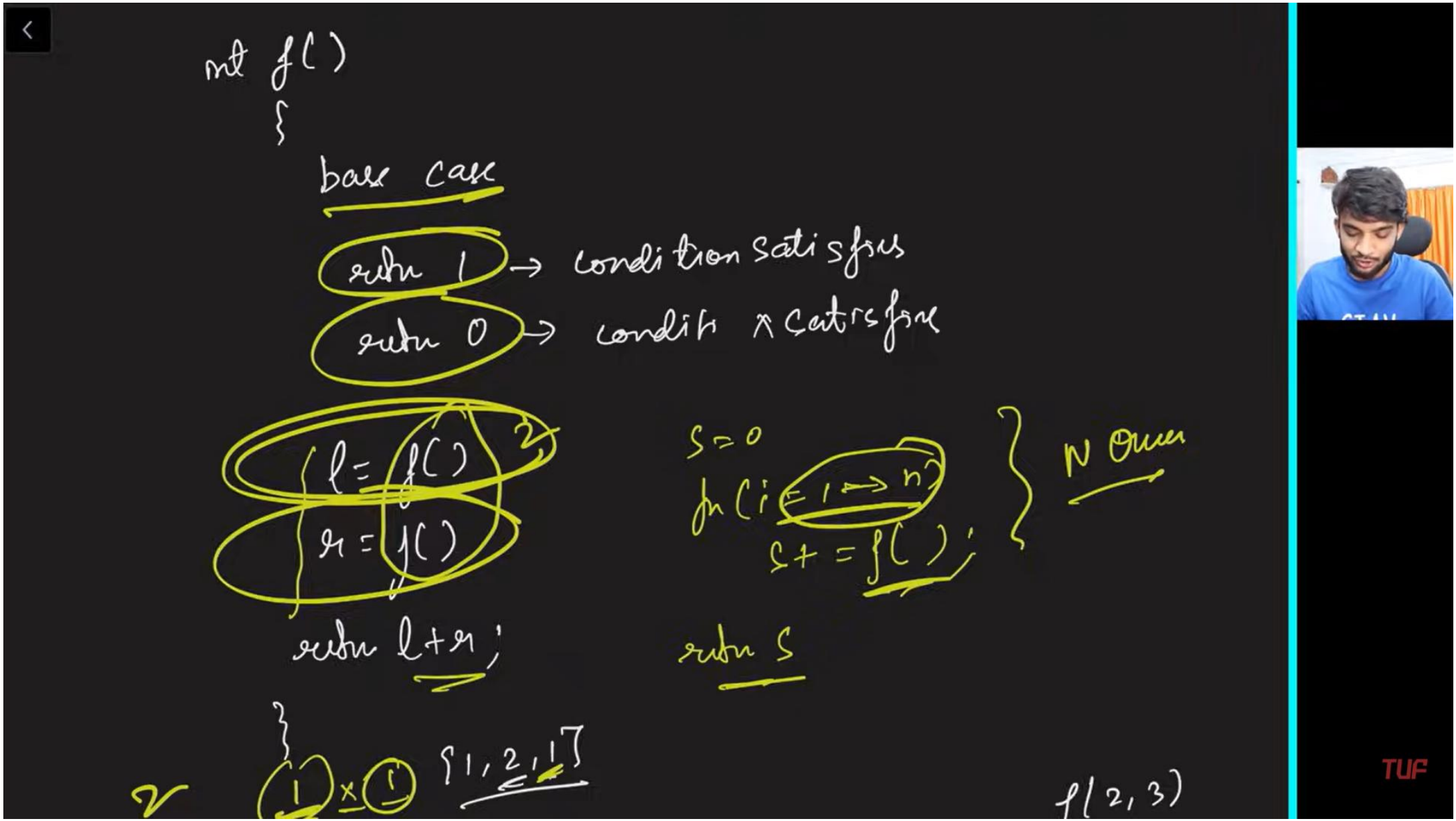
2

N Queen

2 (1) × (1) {1, 2, 1}

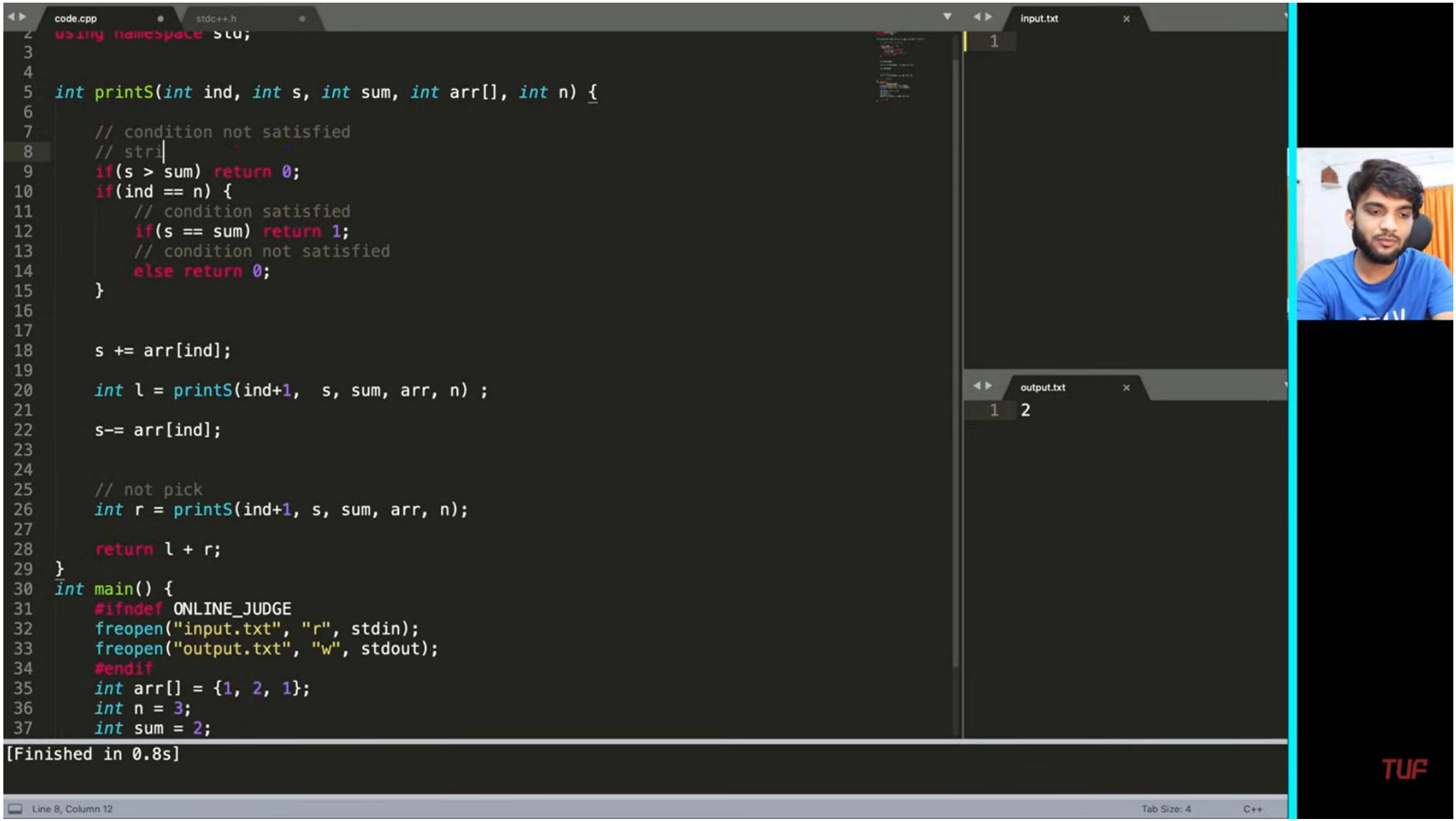
f(2, 3)

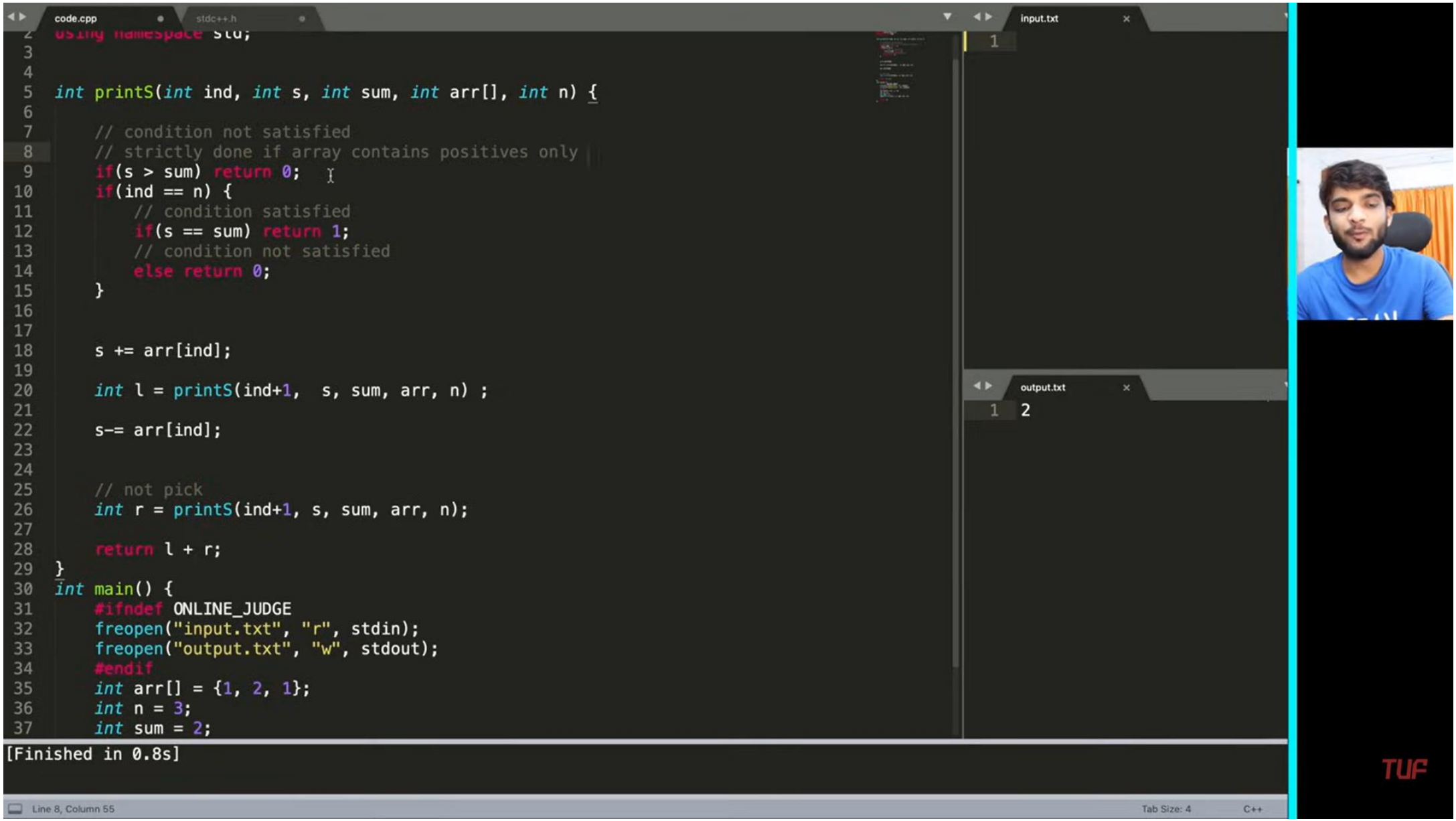

int f()
{
base case
return 1 → condition satisfies
return 0 → condition satisfies
l = f()
r = f()
2
return l+r;
s = 0
for(i = 1 → n)
s += f();
N times
return s
f(2,3)

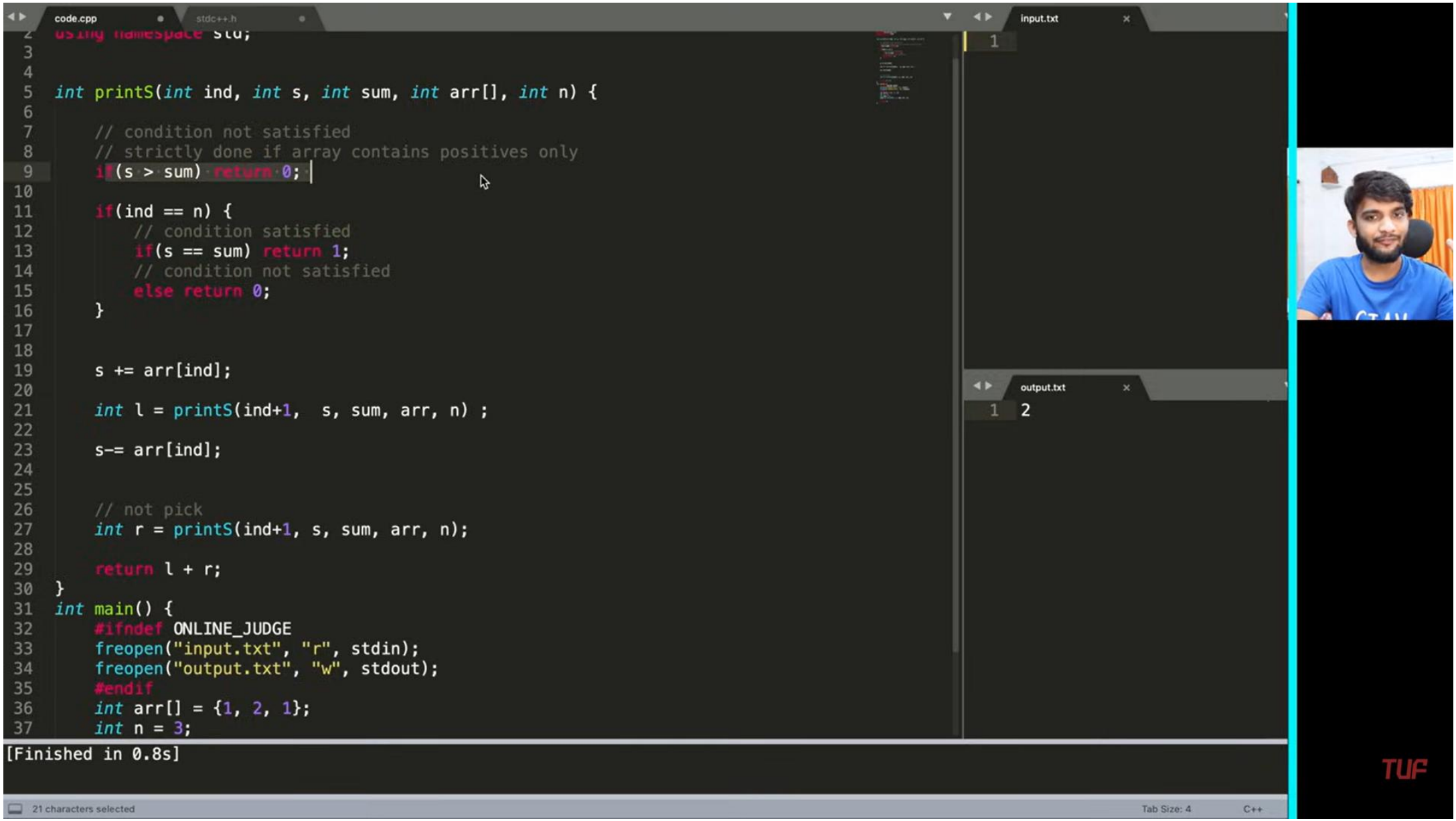

int f()
{
base case
return 1 → condition satisfies
return 0 → condition satisfies
l = f()
r = f()
return l+r;
s=0
for(i=1→n)
s+=f();
return s
N Queen
f(2,3)
TUF

$2 \quad 2 \quad 2 \quad 2 \quad 2 \quad \ldots = 2^n$

TUF


code.cpp
stdc++.h
using namespace std;
int printS(int ind, int s, int sum, int arr[], int n) {
// condition not satisfied
// stri
if(s > sum) return 0;
if(ind == n) {
// condition satisfied
if(s == sum) return 1;
// condition not satisfied
else return 0;
}
s += arr[ind];
int l = printS(ind+1, s, sum, arr, n) ;
s-= arr[ind];
// not pick
int r = printS(ind+1, s, sum, arr, n);
return l + r;
}
int main() {
#ifndef ONLINE_JUDGE
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
#endif
int arr[] = {1, 2, 1};
int n = 3;
int sum = 2;
input.txt
1
output.txt
1 2
[Finished in 0.8s]
Line 8, Column 12
Tab Size: 4
C++
TUF


code.cpp
stdc++.h
using namespace std;
int printS(int ind, int s, int sum, int arr[], int n) {
// condition not satisfied
// strictly done if array contains positives only
if(s > sum) return 0;
if(ind == n) {
// condition satisfied
if(s == sum) return 1;
// condition not satisfied
else return 0;
}
s += arr[ind];
int l = printS(ind+1, s, sum, arr, n) ;
s-= arr[ind];
// not pick
int r = printS(ind+1, s, sum, arr, n);
return l + r;
}
int main() {
#ifndef ONLINE_JUDGE
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
#endif
int arr[] = {1, 2, 1};
int n = 3;
int sum = 2;
input.txt
1
output.txt
1 2
[Finished in 0.8s]
Line 8, Column 55
Tab Size: 4
C++
TUF


code.cpp
stdc++.h
using namespace std;
int printS(int ind, int s, int sum, int arr[], int n) {
// condition not satisfied
// strictly done if array contains positives only
if(s > sum) return 0;
if(ind == n) {
// condition satisfied
if(s == sum) return 1;
// condition not satisfied
else return 0;
}
s += arr[ind];
int l = printS(ind+1, s, sum, arr, n) ;
s-= arr[ind];
// not pick
int r = printS(ind+1, s, sum, arr, n);
return l + r;
}
int main() {
#ifndef ONLINE_JUDGE
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
#endif
int arr[] = {1, 2, 1};
int n = 3;
input.txt
1
output.txt
1 2
[Finished in 0.8s]
21 characters selected
Tab Size: 4
C++
TUF

2 2 2 2 2 = 2

print → param

print 1 → ret T/F & accord func Recursion calls if you get done

p

2 2 2 2 2 = 2

print → param

print 1 → ret T/F & award fnctn Recursion calls if you get true

Count → ret 1
ret 0

print → param

print 1 → ret 1 if & word from Recursion calls if you get true

Count → ret 1 / ret 0 → add all f()'s & ret