Quick Compiler - CodeHelp La | SPOJ.com - Problem EKO

spoj.com/problems/EKO/

Gmail YouTube Maps

Press Esc to exit full screen

Sphere online judge

PROBLEMS STATUS RANKS DISCUSS CONTESTS sign in

Problems / classical / Eko Status Ranking

EKO - Eko

#binary-search

Lumberjack Mirko needs to chop down **M** metres of wood. It is an easy job for him since he has a nifty new woodcutting machine that can take down forests like wildfire. However, Mirko is only allowed to cut a single row of trees.

Mirko"s machine works as follows: Mirko sets a height parameter **H** (in metres), and the machine raises a giant sawblade to that height and cuts off all tree parts higher than **H** (of course, trees not higher than **H** meters remain intact). Mirko then takes the parts that were cut off. For example, if the tree row contains trees with heights of 20, 15, 10, and 17 metres, and Mirko raises his sawblade to 15 metres, the remaining tree heights after cutting will be 15, 15, 10, and 15 metres, respectively, while Mirko will take 5 metres off the first tree and 2 metres off the fourth tree (7 metres of wood in total).

Mirko is **ecologically** minded, so he doesn"t want to cut off more wood than necessary. That"s why he wants to set his sawblade as high as possible. Help Mirko find the **maximum integer height** of the sawblade that still allows him to cut off **at least M** metres of wood.

Input

The first line of input contains two space-separated positive integers, **N** (the number of trees, $1 \leq N \leq 1\,000\,000$) and **M** (Mirko"s required wood amount, $1 \leq M \leq 2\,000\,000\,000$).

The second line of input contains **N** space-separated positive integers less than 1 000 000 000, the heights of each tree (in metres). The sum of all heights will exceed **M**, thus Mirko will always be able to obtain the required amount of wood.

Output

The first and only line of output must contain the required height setting.

Example

```
Input:
4 7
20 15 10 17

Output:
15
```

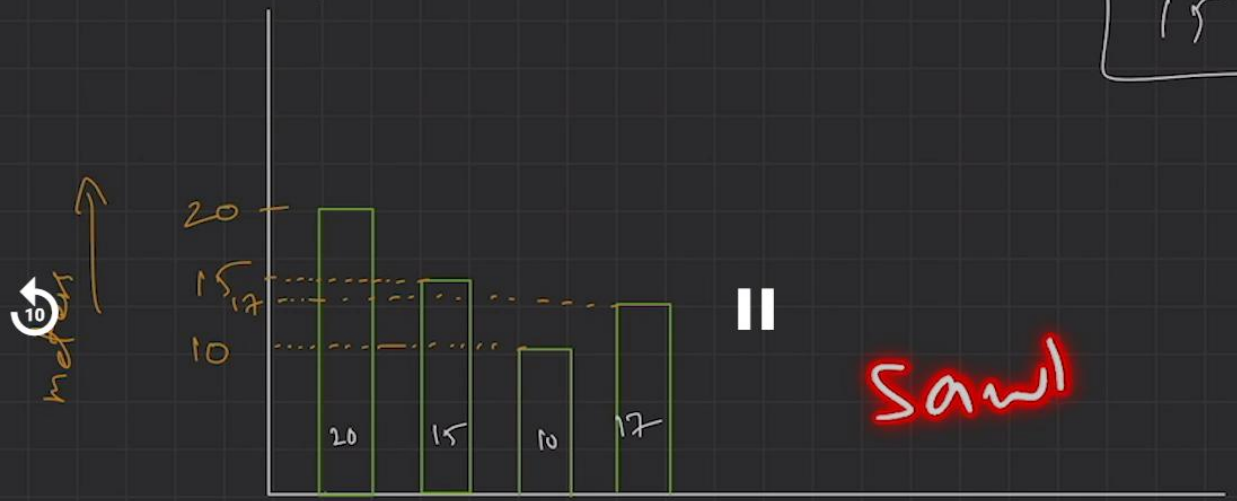
```
Input:
5 20
4 42 40 26 46
```

Submit solution!

Added by:	ghorvat
Date:	2012-07-02
Time limit:	1s
Source limit:	50000B
Memory limit:	1536MB
Cluster:	Cube (Intel G860)
Languages:	All
Resource:	COCI 2011/2012

EKO - SPOJ

20 | 15 | 10 | 17 meter



15 m → sawblade

Sawl

M = 7m

⇒ Sawblade ⇒ 15m

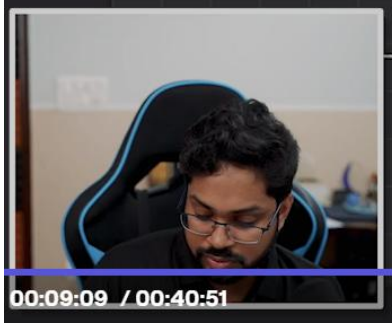
T_1 → 20 − 15 = 5m

Total wood = 5 + 2 = 7m,

T_2 = 0m,

T_3 = 0m,

T_4 = 17 − 15 = 2m



EKO - SPOJ

20 | 15 | 10 | 17 meter

15 m → sawblade

M = 7m

⇒ Sawblade ⇒ 15m

T_1 → 20 − 15 = 5m

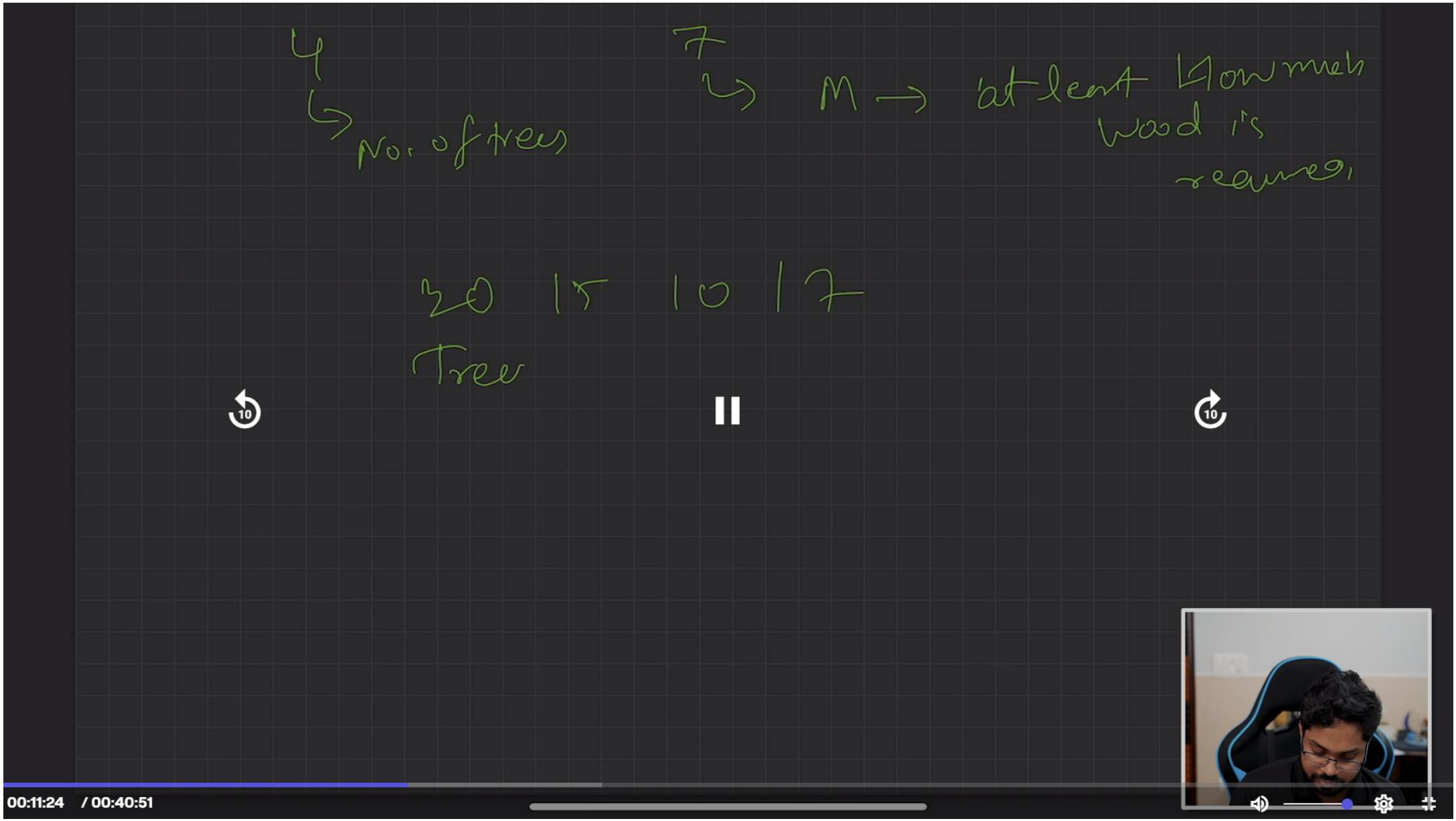
Total wood = 5 + 2 = 7m,

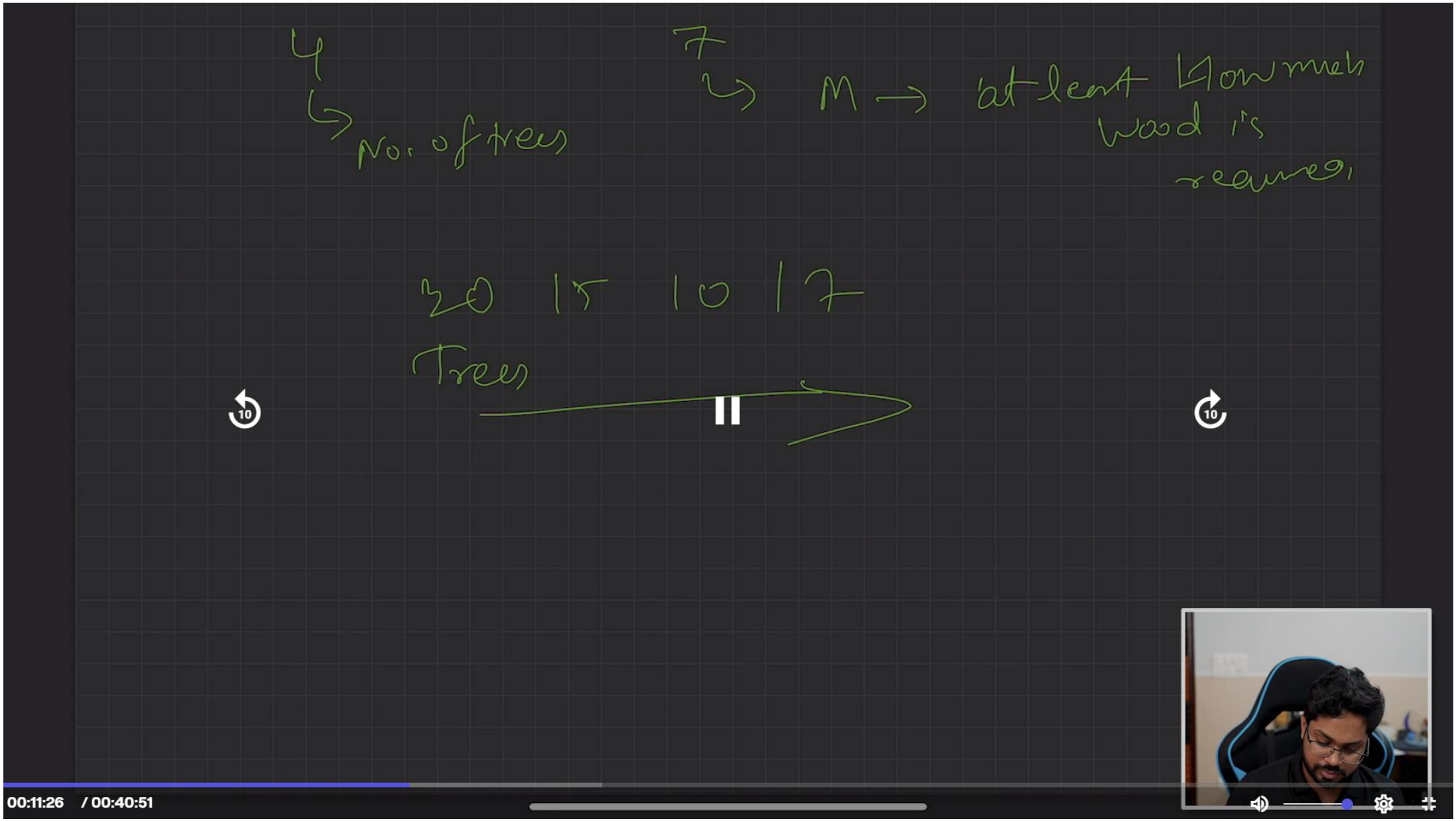
T_2 = 0m,

T_3 = 0m,

T_4 = 17 − 15 = 2m

00:09:24 / 00:40:51


4
No. of trees
7
M → atleast How much wood is required
20 15 10 17
Tree
10
10
00:11:24 / 00:40:51


4
No. of trees
7
M → atleast How much wood is required
20 15 10 17
Trees
10
10
00:11:26 / 00:40:51

Q → 20 | 15 | 10 | 17 M = 7

⇒

To Find → sawblade's height

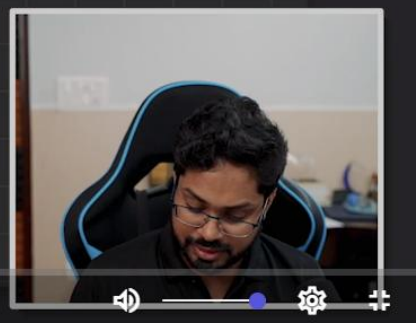
Brute force →

Sawblade height → 1m

⇒ (20 − 1) + (15 − 1) + (10 − 1) + (17 − 1)

= 19 + 14 + 9 + 16

= 58 m.

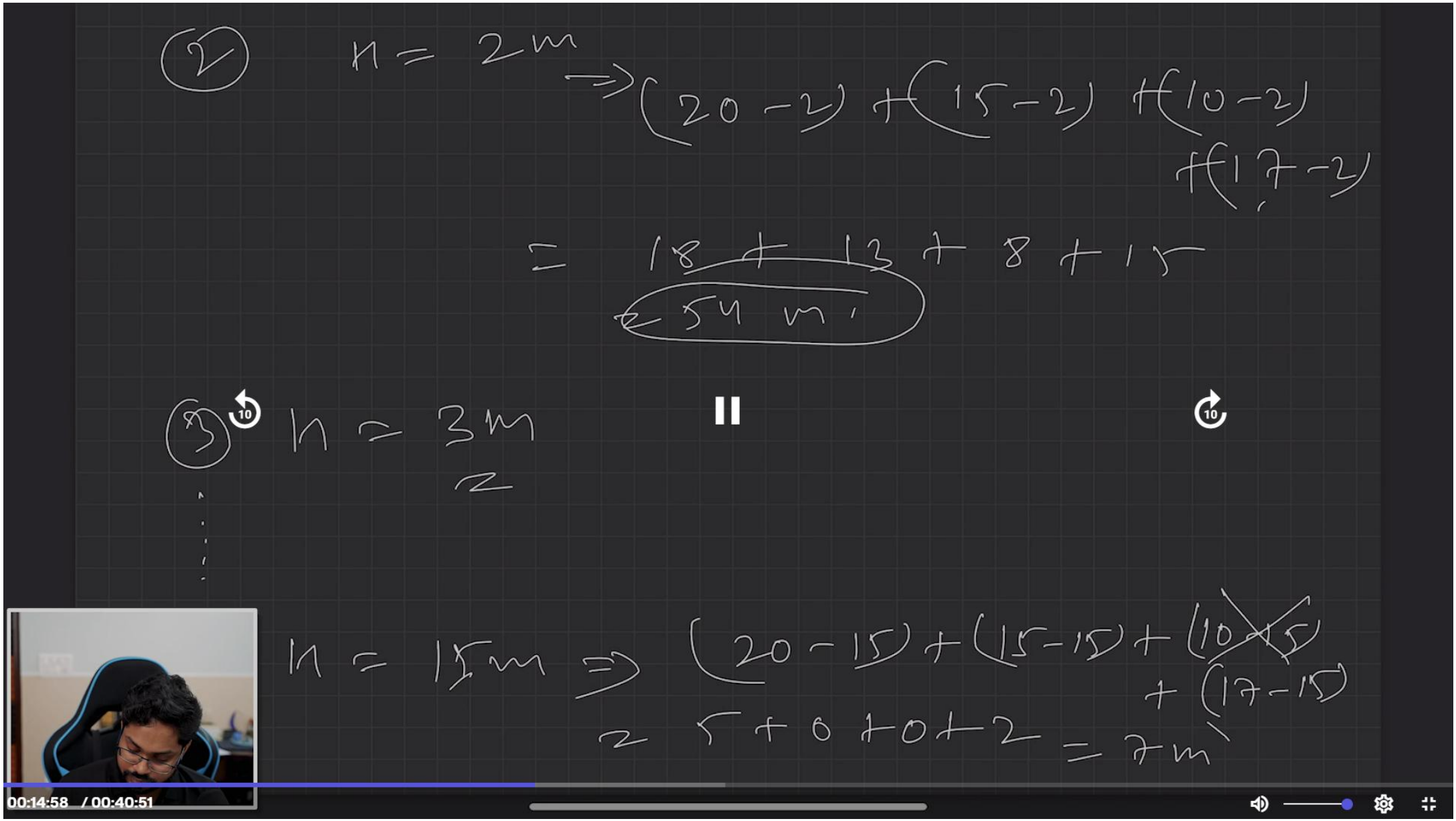


(2) $n = 2m$

$\Rightarrow (20-2) + (15-2) + (10-2) + (17-2)$

$= 18 + 13 + 8 + 15$

$= 54$ m


(2) n = 2m ⇒ (20 − 2) + (15 − 2) + (10 − 2) + (17 − 2)
= 18 + 13 + 8 + 15
= 54 m.
(3) n = 3m
n = 15m ⇒ (20 − 15) + (15 − 15) + (10 − 5) + (17 − 15)
= 5 + 0 + 0 + 2 = 7m

(5) $n = 16$

$\Rightarrow 20 - 16 + (15 - 16) + (10 - 10) + (17 - 16$

$= 4 + 0 + 0 + 1$

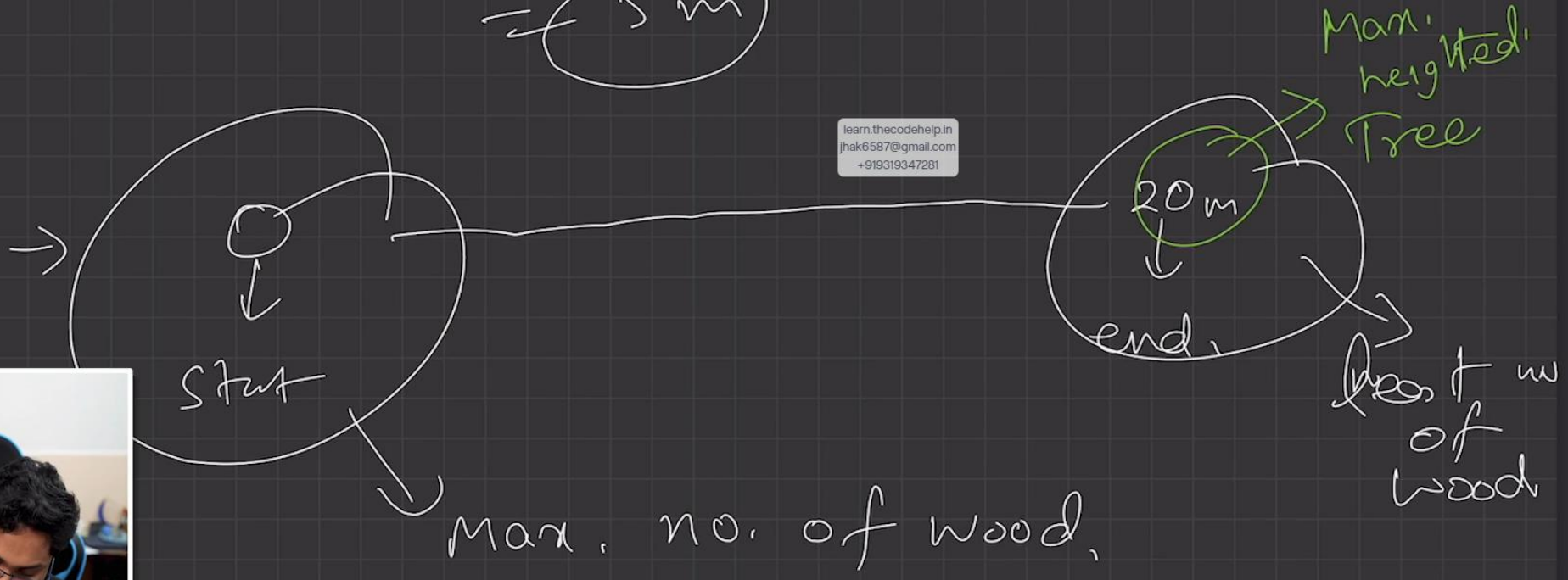
$= 5$ m

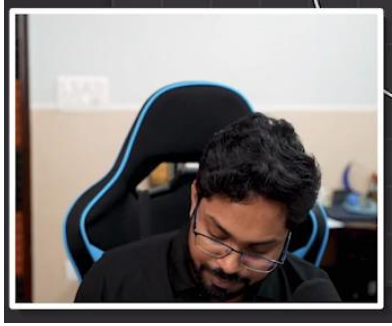
(5) H = 16

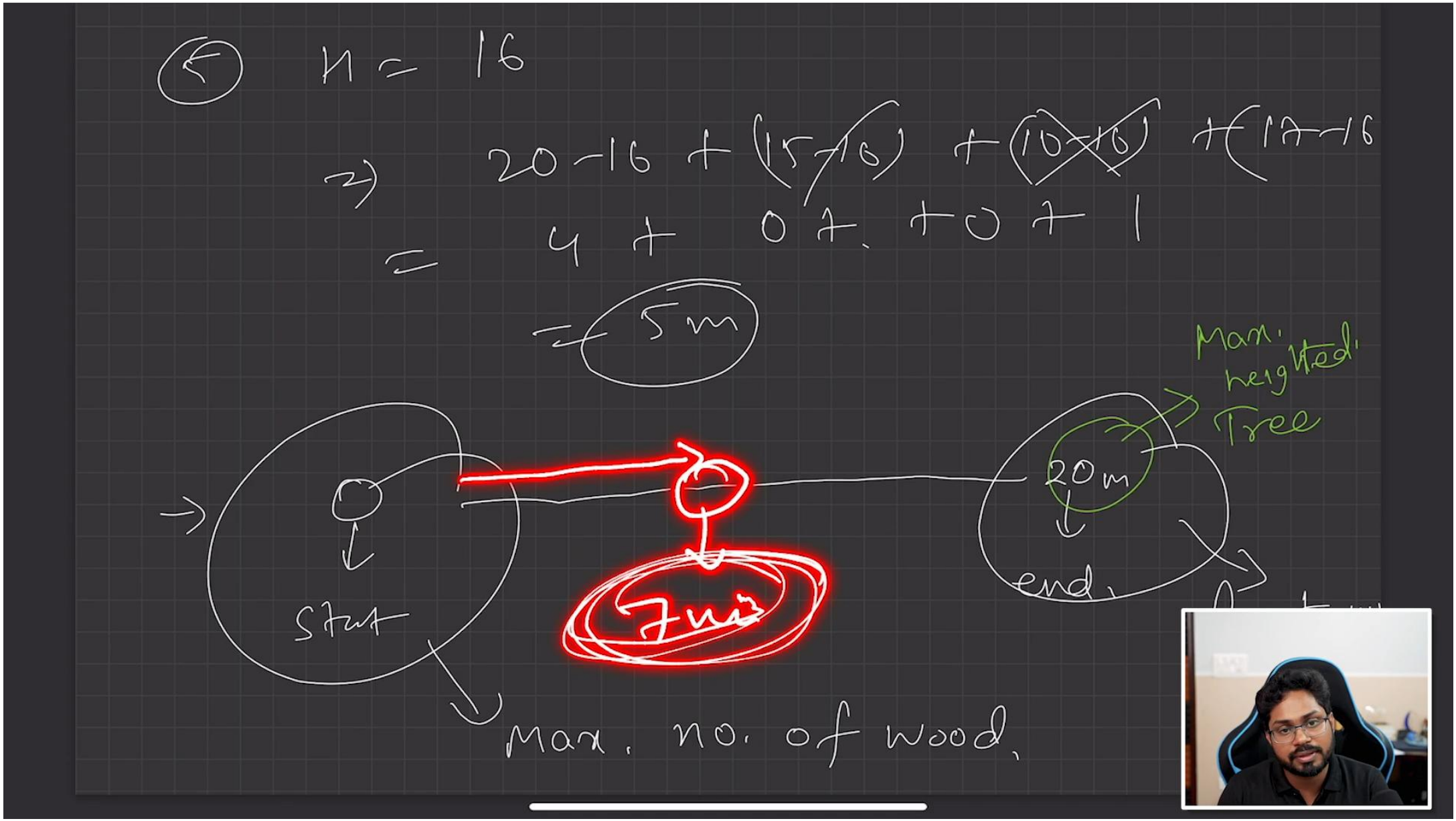
⇒ 20 − 16 + (15 − 10) + (10 − 10) + (17 − 16

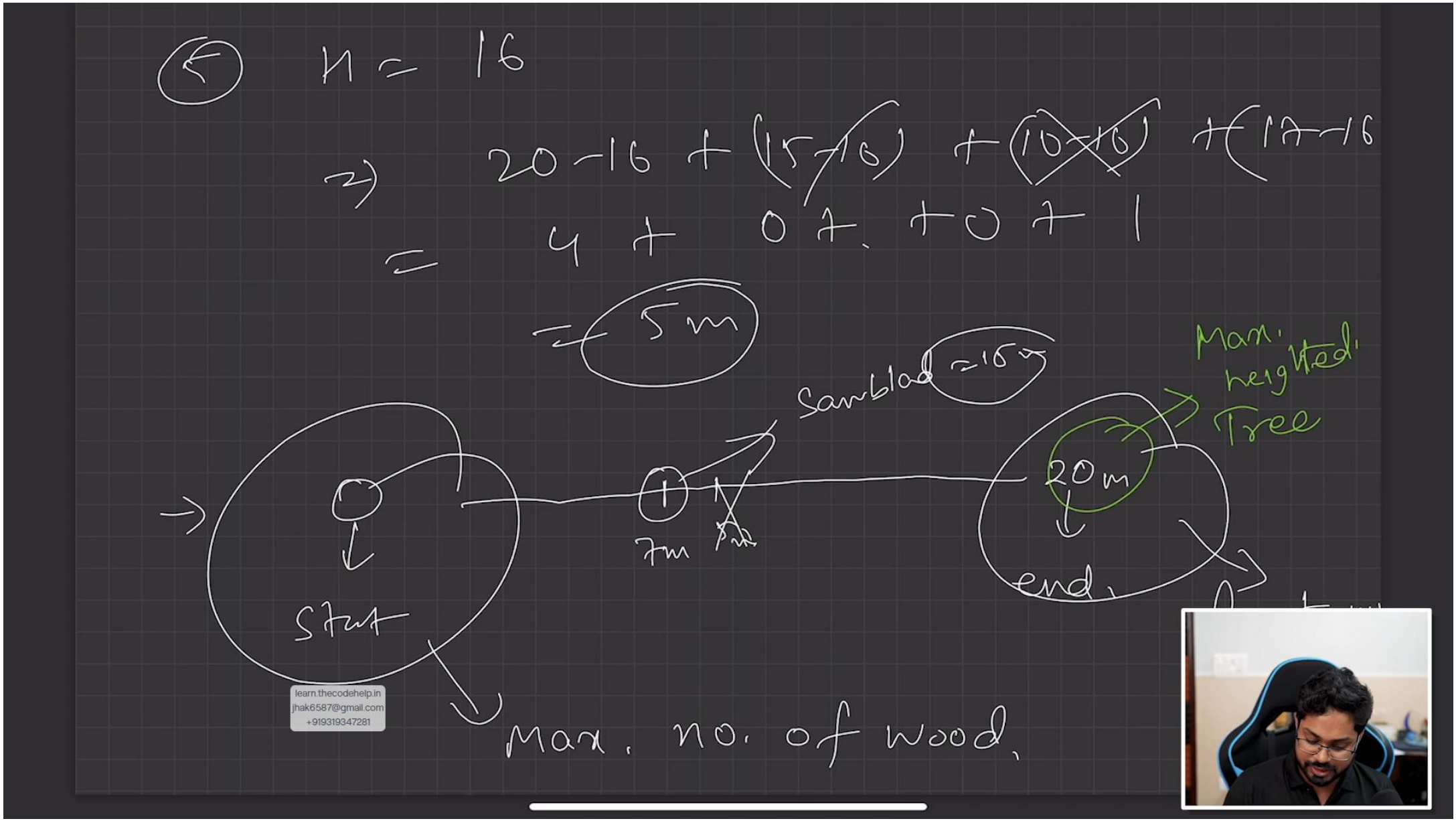
= 4 + 0 + 0 + 1

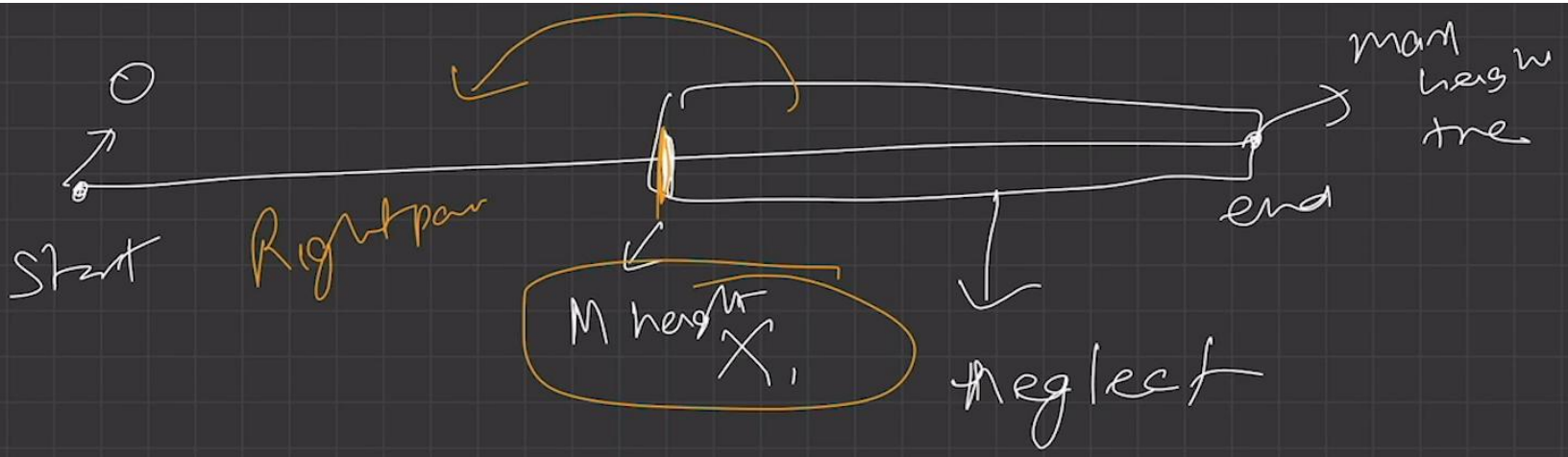
= 5 m

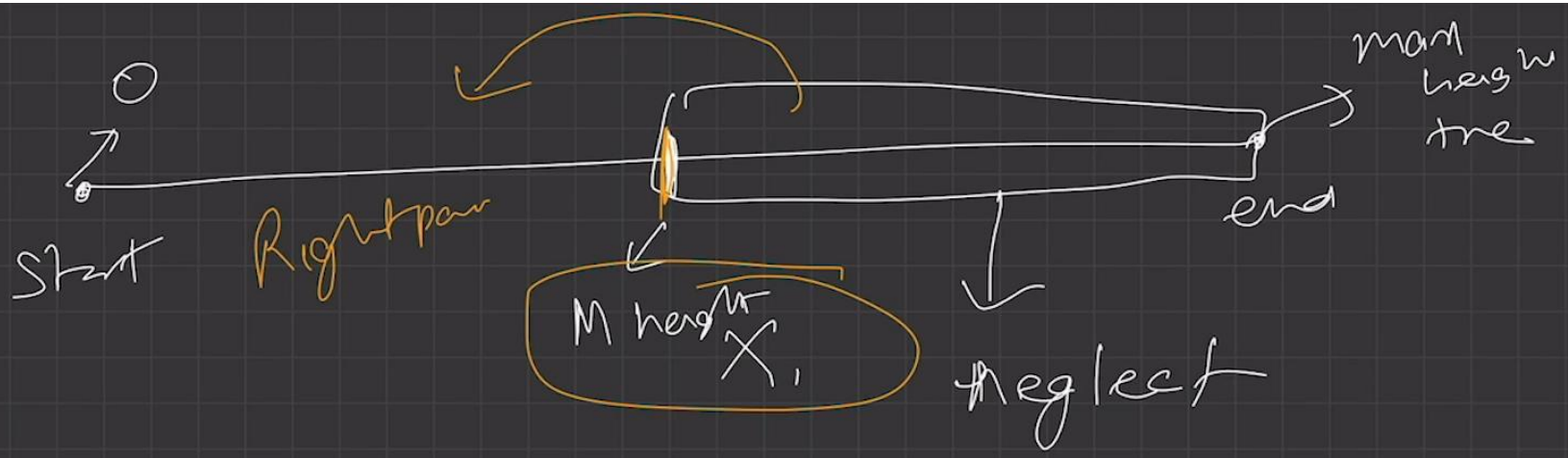





H = 16
20-16 + (15-10) + (10-10) + (17-16
4 + 0 + 0 + 1
5m
Max. Weighted Tree
20m
start
end
Max. no. of wood.



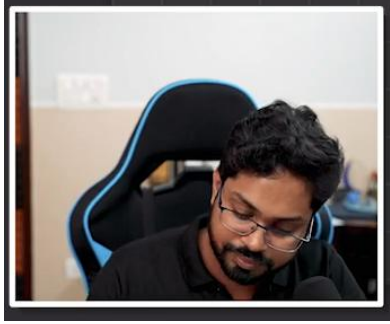


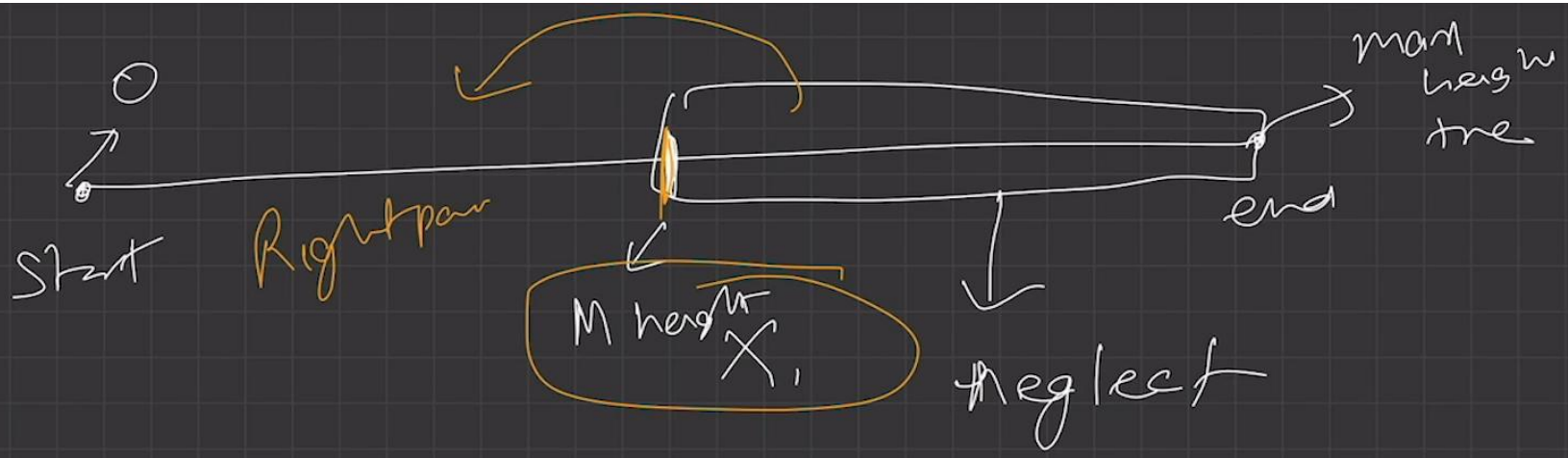


2) M = 7 Trees array → 20 | 15 | 10 | 17.

① start → 0 end = 20.

mid → $\frac{20+0}{2}$ = 10m, 22 ≥ 7m wood. mil payegi.





2) M = 7 Trees array → 20 | 15 | 10 | 17.

① start → 0 end = 20.

mid → $\frac{20+0}{2}$ = 10m. 22 ≥ 7m wood. mil payegi

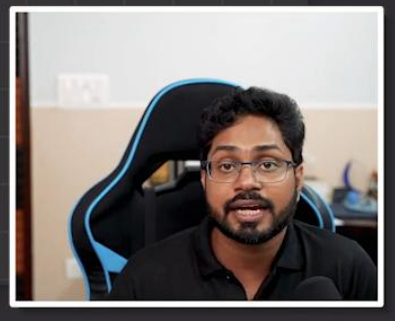
-> 20 | 15 | 10 | 17

(20-10) (15-10) (10-0) (17-10)

⇒ 10 + 5 + 0 + 7 = 22m

(22m >= 7) ? 2

Possible soln



-2) 20 | 15 | 10 | 17

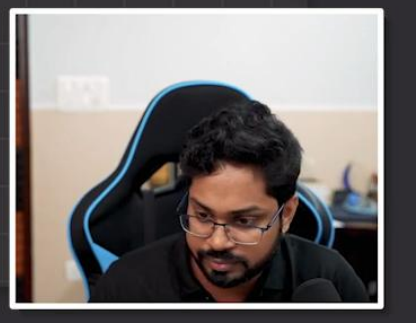
(20-10) (15-10) (10-0) (17-10)

$\Rightarrow 10 + 5 + 0 + 7 = 22m$

(22m >= 7) ? 2

Possible soln


```
if (possible soln)
{
    ans = mid;
    start = mid + 1;
}
```



-2) 20 | 15 | 10 | 7

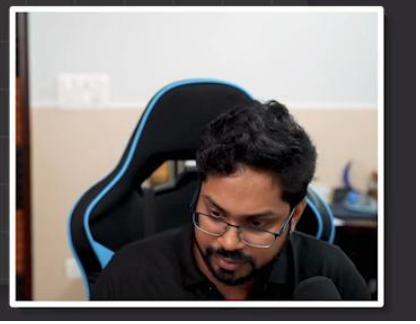
(20-10) (15-10) (10-0) (17-10)

$\Rightarrow 10 + 5 + 0 + 7 = 22m$

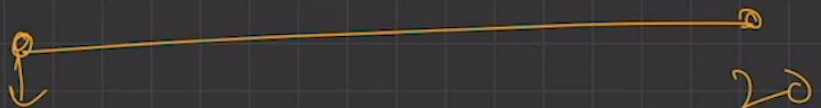
Possible soln

(22m >= 7) ? 2

if (possible soln)
{
  ans = mid;
  start = mid + 1;
}



(2)



⇒ mid = $\frac{11+20}{2}$ = (15)

20 | 15 | 10 | 17

⇒ 15 ⇒

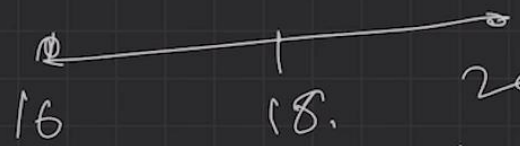
(20 − 15) (15 − 15) (10 − 15) (17 − 15)

⇒ 7m

if (possible)

{ ans = mid;
start = mid + 1;
}

(3)



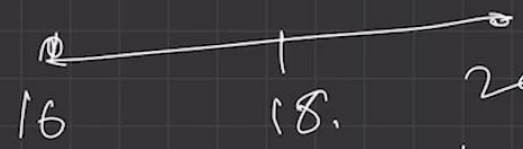
saw blade → 18m →

20 | 15 | 10 | 17

(20 − 18) + (15 − 18) + (10 − 18) + 17 − 1

= 2m

(3)



20 | 15 | 10 | 17.

saw blade → 18m →

(20 − 18) + (15 ~~− 18~~) + (10 ~~− 18~~) + 17 ~~− 1~~

= 2m

2 ~~≥ 7~~

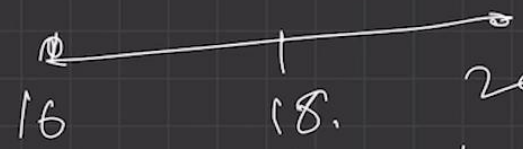
if(not possible soln.)

{

end = mid − 1 ;

}.

(3)



20 | 15 | 10 | 17.

Saw blade → 18m →

(20-18) + (15-18) + (10-18) + (17-18)

= 2m

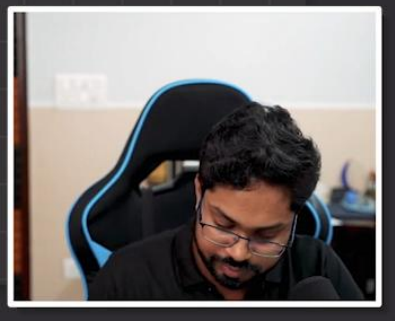
if(not possible soln.)

{

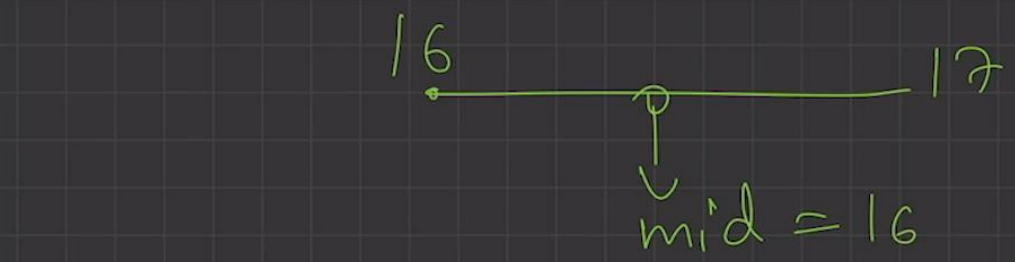
end = mid - 1 ;

}.

2 ~~2~~ 7.



(4)



⇒ $(20-16) + (17-16) + (13-16) + (17-16)$

$= 4 + 0 + 7 + 0 + 1$

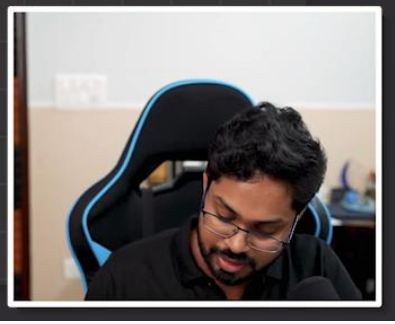
$= 5m$

~~$5m \geq 7m$~~

⇒ end = mid − 1

end = 16 − 1

= 15.



4

$\Rightarrow (20-16) + (17-16) + (17-16) + (17-16)$

$= 4 + 0 + 0 + 1$

$= 5$ m

5m ≯ 7m

$\Rightarrow$ end = mid − 1

end = 16 − 1

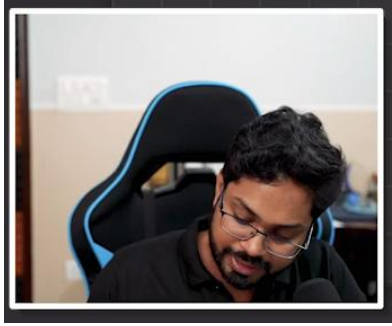
= 15.

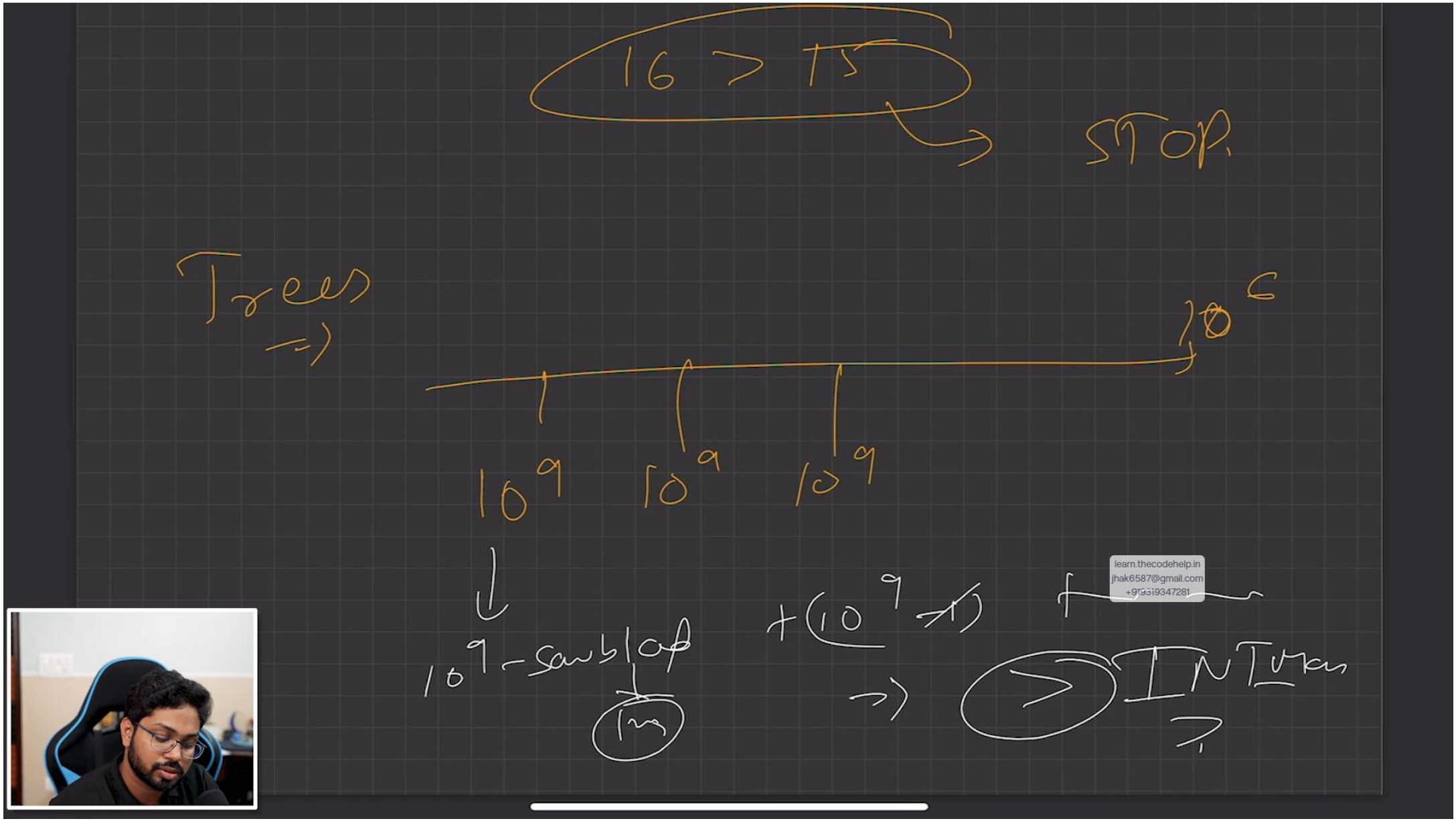
start = 16

end = 15

while (start > end)

stop





Quick Compiler - CodeHelp La | SPOJ.com - Problem EKO | main.cpp - week3class3 - Rep

labs.thecodehelp.in/quick-compiler

Gmail YouTube Maps

CODE HELP <LABS/> Quick Compiler Beta Fruitbox Flex

RUN C++ File Settings SAVE SHARE

SOURCE - main.cpp

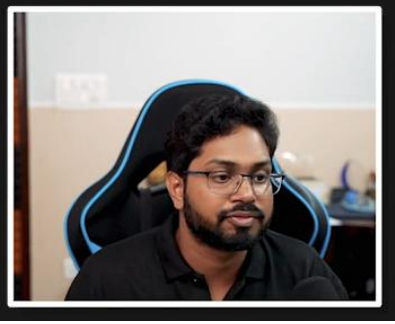
```
30      return ans;
31  }
32
33  int main() {
34    long long int n, m;
35    cin >> n >> m;
36    vector<long long int> trees;
37    while(n--){
38      long long int height;
39      cin >> height;
40      trees.push_back(height);
41    }
42
43    cout << maxSawBladeHeight(trees, m) << endl;
44
45    return 0;
46  }
```

STDIN

```
4 7
20 15 10 17
```

STDOUT

```
15
```



Quick Compiler - CodeHelp La | SPOJ.com - Problem EKO | main.cpp - week3class3 - Rep

labs.thecodehelp.in/quick-compiler

Gmail YouTube Maps

CODE HELP <LABS/> Quick Compiler Beta Fruitbox Flex

RUN C++ File Settings SAVE SHARE

SOURCE - main.cpp

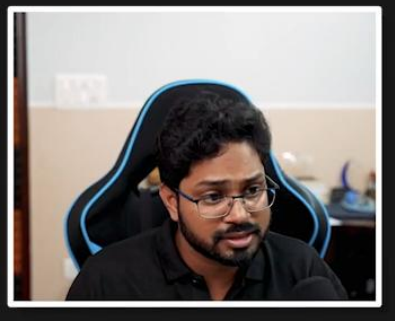
```
13    return woodCollected >= m;
14  }
15
16  long long int maxSawBladeHeight( vector<long long int> trees, long
    long int m){
17    long long int start = 0, end, ans = -1;
18    end = *max_element(trees.begin(), trees.end());
19
20    while(start<=end){
21      long long int mid = start + (end - start) / 2;
22      if(isPossibleSolution(trees, m, mid)) {
23        ans = mid;
24        start = mid + 1;
25      }
26      else{
27        end = mid - 1;
28      }
29    }
30      return ans;
31  }
32
```

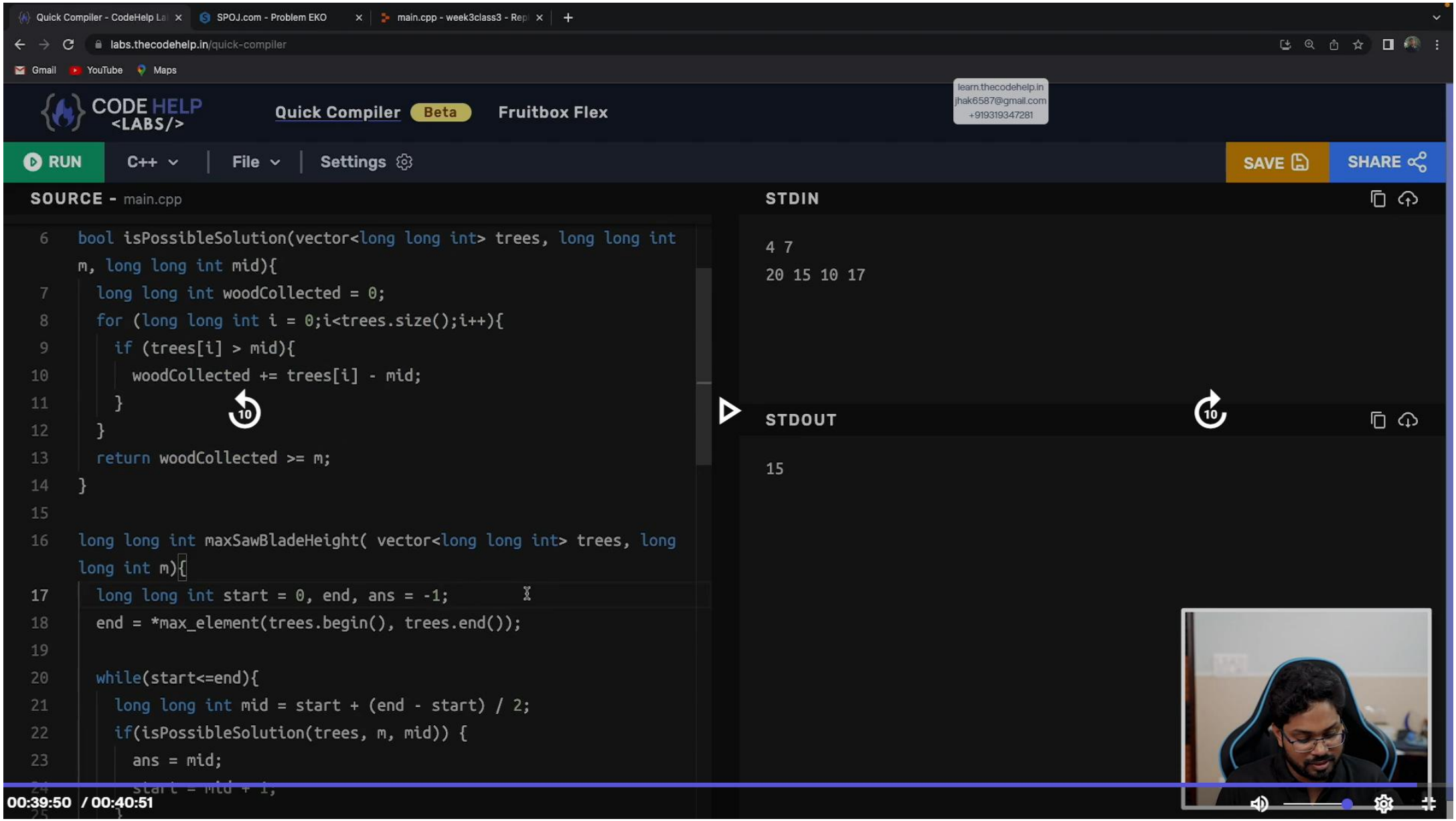
STDIN

```
4 7
20 15 10 17
```

STDOUT

```
15
```




Quick Compiler - CodeHelp La
SPOJ.com - Problem EKO
main.cpp - week3class3 - Rep
labs.thecodehelp.in/quick-compiler
Gmail
YouTube
Maps
CODE HELP <LABS/>
Quick Compiler
Beta
Fruitbox Flex
learn.thecodehelp.in
jhak6587@gmail.com
+919319347281
RUN
C++
File
Settings
SAVE
SHARE
SOURCE - main.cpp
6 bool isPossibleSolution(vector<long long int> trees, long long int m, long long int mid){
7 long long int woodCollected = 0;
8 for (long long int i = 0;i<trees.size();i++){
9 if (trees[i] > mid){
10 woodCollected += trees[i] - mid;
11 }
12 }
13 return woodCollected >= m;
14 }
15
16 long long int maxSawBladeHeight(vector<long long int> trees, long long int m){
17 long long int start = 0, end, ans = -1;
18 end = *max_element(trees.begin(), trees.end());
19
20 while(start<=end){
21 long long int mid = start + (end - start) / 2;
22 if(isPossibleSolution(trees, m, mid)) {
23 ans = mid;
24 start = mid + 1;
25 }
STDIN
4 7
20 15 10 17
STDOUT
15
00:39:50 / 00:40:51

Quick Compiler - CodeHelp La | SPOJ.com - Problem PRATA

spoj.com/problems/PRATA/

Gmail YouTube Maps

Sphere online judge

PROBLEMS STATUS RANKS DISCUSS CONTESTS PROFILE

Problems / classical / Roti Prata

My status Status Ranking

PRATA - Roti Prata

no tags

IEEE is having its AGM next week and the president wants to serve cheese prata after the meeting. The subcommittee members are asked to go to food connection and get P (P<=1000) pratas packed for the function. The stall has L cooks (L<=50) and each cook has a rank R (1<=R<=8). A cook with a rank R can cook 1 prata in the first R minutes 1 more prata in the next 2R minutes, 1 more prata in 3R minutes and so on(he can only cook a complete prata) (For example if a cook is ranked 2.. he will cook one prata in 2 minutes one more prata in the next 4 mins an one more in the next 6 minutes hence in total 12 minutes he cooks 3 pratas in 13 minutes also he can cook only 3 pratas as he does not have enough time for the 4th prata). The webmaster wants to know the minimum time to get the order done. Please write a program to help him out.

Input

The first line tells the number of test cases. Each test case consist of 2 lines. In the first line of the test case we have P the number of prata ordered. In the next line the first integer denotes the number of cooks L and L integers follow in the same line each denoting the rank of a cook.

Output

Print an integer which tells the number of minutes needed to get the order done.

Example

```
Input:
3
10
4 1 2 3 4
8
1 1
8
8 1 1 1 1 1 1 1 1

Output:
12
36
1
```

Submit solution!

Submit solution!

Added by:	Saransh Bansal
Date:	2011-05-14
Time limit:	1s
Source limit:	50000B
Memory limit:	1536MB
Cluster:	Cube (Intel G860)
Languages:	All except: ASM64
Resource:	Own problem- NTU IEEE codejam 2011

Vote requirements

- ✔ be spoj user for at least 5 days
- ✖ solved 1 from 15 needed problems
- ✔ solve this problem

Own tags

Tag name Add

PRATA SPOJ

2) P → Prata ≤ 1000

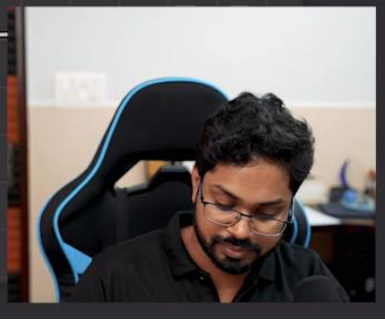
L cooks ⇒ ≤ 50

R → [1, 8]

Ci → 1st 2nd 3rd 4th ... 1000

R 2R 3R 4R ... 1000R

C1 → 1 (1m, 2m, 3m)



① Brute force

nP = 10

4 cooks

R ⇒ 1/2/3/4.

① for(i = 0m ; i < (max) ; i++)

{

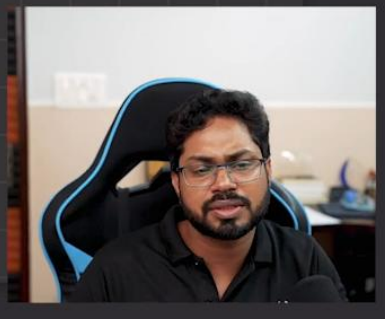
0m ✓

1m ✓

} 2m ✓

3m ✓

4m ✓



$R \rightarrow$ 1 | 2 | 3 | 4

$C_1\ C_2\ C_3\ C_4$

$P = 10$

$C_4 \Rightarrow$ all 10 pratai

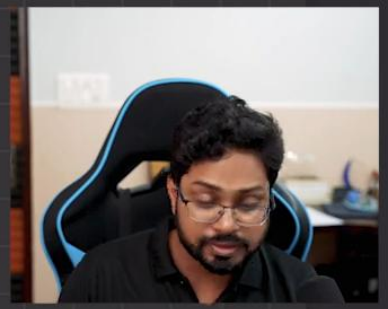
$1 \times 4 + 2 \times 4 + 3 \times 4 + 4 \times 4 + 5 \times 4 + 6 \times 4$

$+ 7 \times 4 + 8 \times 4 + 9 \times 4 + 10 \times 4$

$R \rightarrow 1 | 2 | 3 | (4)$

$C_1\ C_2\ C_3\ C_4$

$P = 10$



$C_4 \Rightarrow$ all 10 pratai,

$1 \times 4 + 2 \times 4 + 3 \times 4 + 4 \times 4 + 5 \times 4 + 6 \times 4$

$+ 7 \times 4 + 8 \times 4 + 9 \times 4 + 10 \times 4,$

$\Rightarrow 4 \times [1 + 2 + 3 + \dots + 10]$

$4 \times \left[\frac{10 \times (10+1)}{2}\right] = 20 \times 11$

$= 220$ m.

$[1 + 2 + 3 + \dots + n]$ → Sum,

$\frac{n \times (n+1)}{2}$ ✗,

$R \rightarrow 1 | 2 | 3 | (4)$

$C_1\ C_2\ C_3\ C_4$

$P = 10$

$C_4 \Rightarrow$ all 10 pratas.

$1 \times 4 + 2 \times 4 + 3 \times 4 + 4 \times 4 + 5 \times 4 + 6 \times 4$

$+ 7 \times 4 + 8 \times 4 + 9 \times 4 + 10 \times 4.$

$\Rightarrow 4 \times [1 + 2 + 3 + \dots + 10]$

$4 \times \left[\frac{10 \times (10+1)}{2}\right] = 20 \times 11$

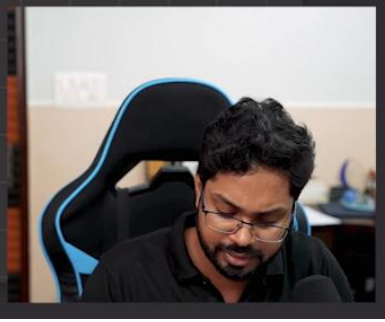
$= 220$ ans.

$[1 + 2 + 3 + \dots + (n)]$ → sum, → 10

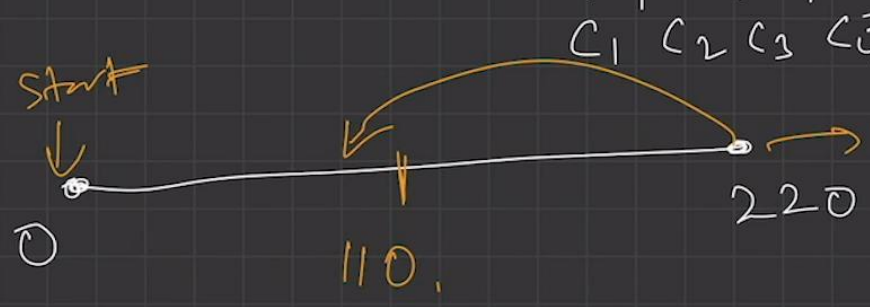
$\frac{n \times (n+1)}{2}$ ✗

0

220 (Time taken by cook with highest rank when all the parata are given to him)



Q 10-P 1|2|3|4 Rankes.



① Can Cooks Cook order of 10 Parata in mid=110 min?

$C_1 \Rightarrow 1 + 2\times1 + 3\times1 + 4\times1 - - - -$

$= 1 + 2 + 3 + 4 + - - - - - + 10$

$= 55$ min

$55 \leq 110$

⇒ if (Possible sol)
{ ans = mid;
end = mid-1;
}

(2)

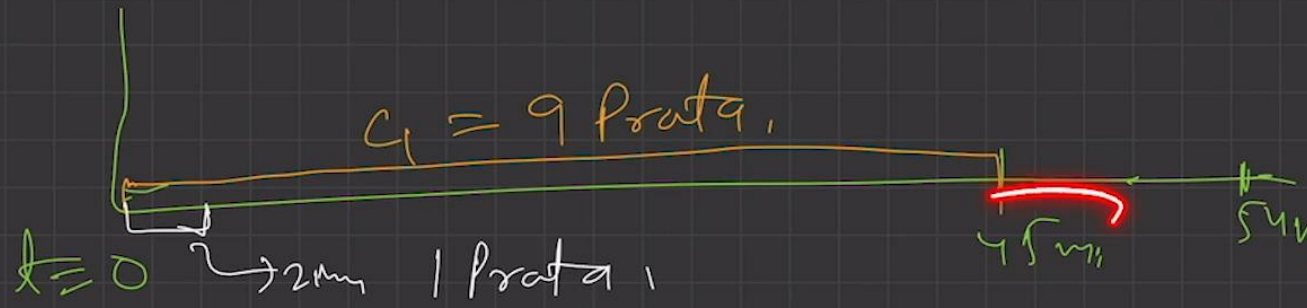
10 - P.

mid = 54.

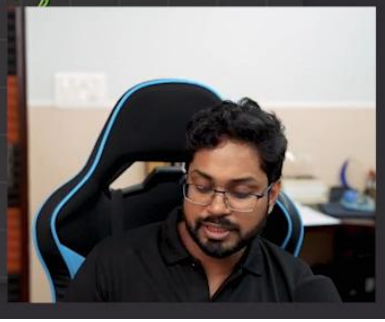
1|2|3|4

$C_1 \rightarrow 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9$

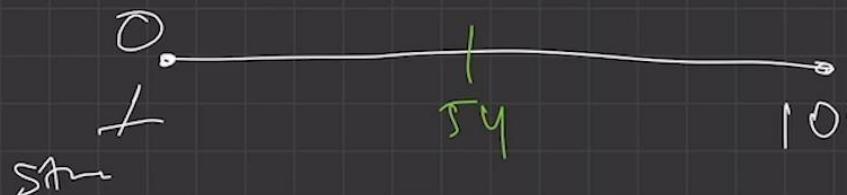
$= 45$ min.



$C_2 \Rightarrow$ $1 \times 2 = 2$ min



(2)



10 - P.

mid = 54.

1|2|3|4

C4 → 1+2+3+4+5+6+7+8+9

= 45 min.

1×2 = 2mins

```
if (Possible
{ ans = mid;
  end = mid-1;
```

(3)

26 min? Order =

6 Prata.

$C_1 \Rightarrow$ 1 + 2 + 3 + 4 + 5 + 6 → 21 min.

$C_2 \Rightarrow$ 2 + 2×2 + 2×3

↳ 2 + 4 + 6 + 8 ⇒ 20 min

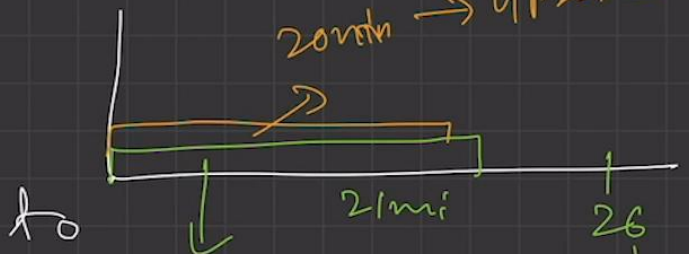
4 Prata

→ 10 ₹

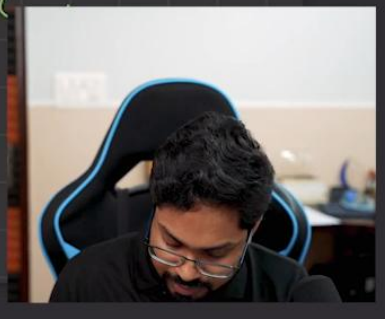
10 - ₹

1 | 2 | 3 | 4.

C_1 C_2 C_3 C_4



to 6 Prata



(3)

10-P

1 | 2 | 3 | 4.

C_1 C_2 C_3 C_4

26 min? Order =

$C_1 \Rightarrow$ 6 Prata.

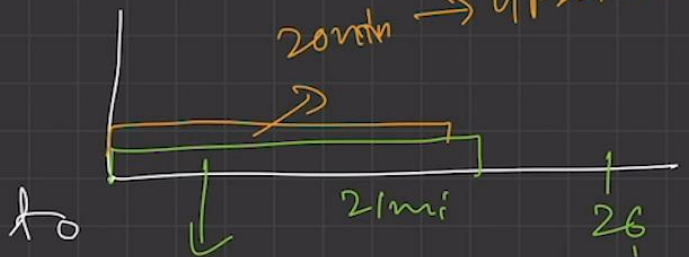
$1 + 2 + 3 + 4 + 5 + 6$

21 min.

$C_2 \Rightarrow$ $2 + 2\times2 + 2\times3$

$\hookrightarrow$ $2 + 4 + 6 + 8 \Rightarrow$ 20 min

4 Prata



6 Prata

10 Prata

(4)

1|2|3|4.

4P ✓

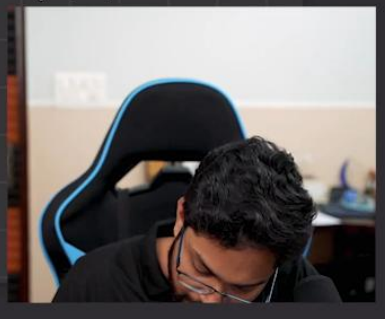
$C_1 \Rightarrow$ 1 + 2 + 3 + 4 + 5

10min.

$C_2 \Rightarrow$ 2 + 4 + 6 = 3 Prod

12min.

C1 4P.



(4)

1|2|3|4.

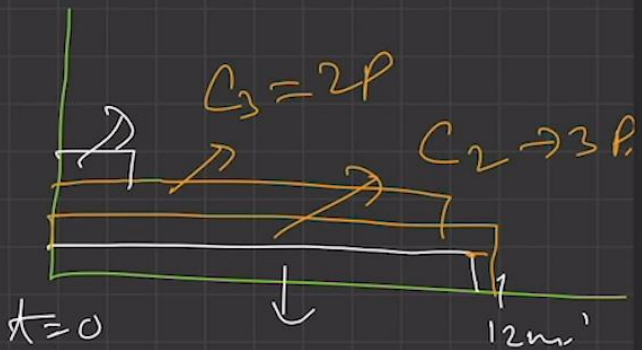
4P ✓

$C_1 \Rightarrow$ 1 + 2 + 3 + 4 + 5

10min.

$C_2 \Rightarrow$ 2 + 4 + 6 = 3 Prod

12min.

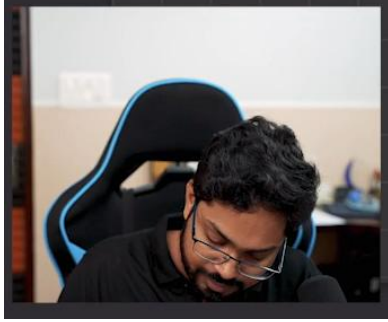


C_1 4P.

3 + 6 + ~~4~~ → 2P.

9 min

C_4 → 4 ✓ (P.)



4

1/2/3/4.

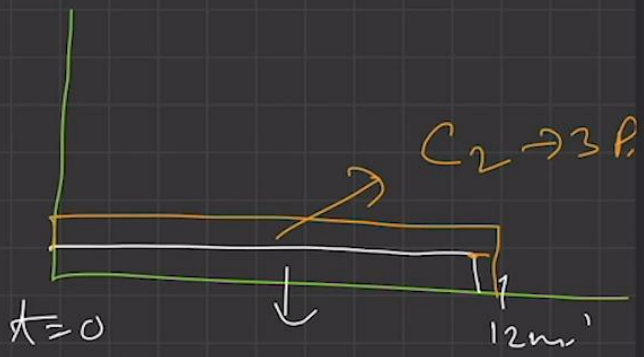
4P ✓

$C_1 \Rightarrow$ 1 + 2 + 3 + 4 + 5

10min.

$C_2 \Rightarrow$ 2 + 4 + 6 = 3 Prod

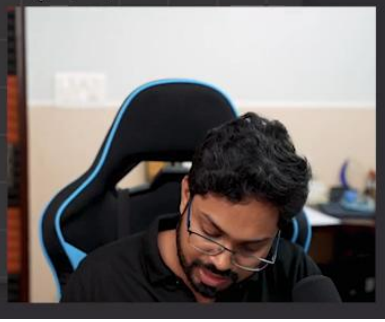
12min.



$C_3 \Rightarrow$ 3 + 6 → 2P.

9 min

$C_4 \Rightarrow$



(4)

1|2|3|4.

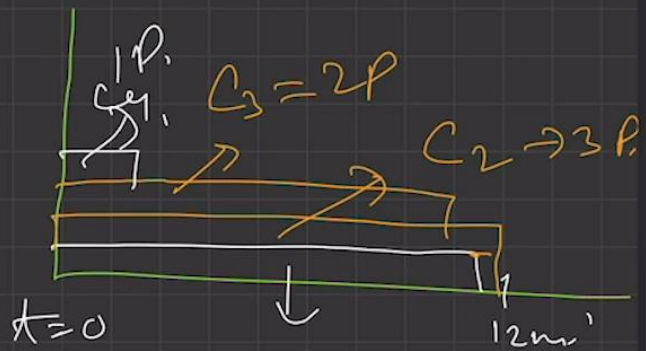
4P ✓

$C_1 \Rightarrow$ 1 + 2 + 3 + 4 + 5

10min.

$C_2 \Rightarrow$ 2 + 4 + 6 = 3 Prod

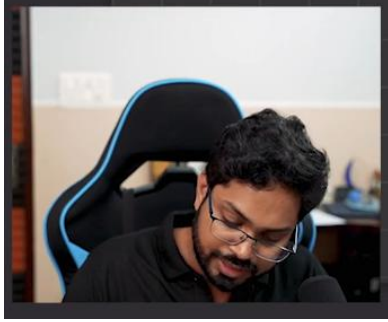
12min.



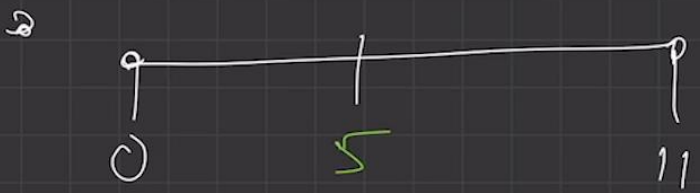
3 + 6 + ~~4~~ → 2P.

9 min

C_4 → 4 ✓ (P.)



⑤



1|2|3|4.

$\lceil \frac{min}{} \rceil$

$C_1 \Rightarrow$ 1 + 2 + ~~3~~. 2P.

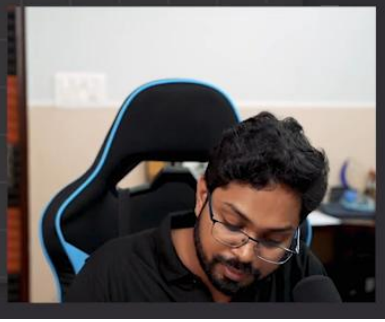
3min

$C_2 \Rightarrow$ 2 + ~~4~~ 1P.

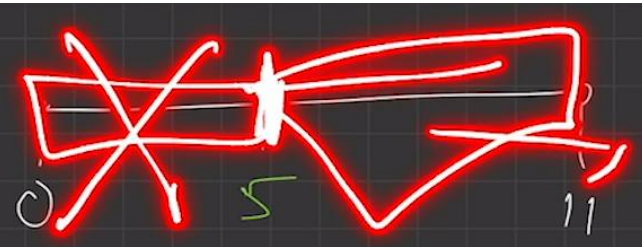
$C_3 \Rightarrow$ 3 + ~~6~~ = 1P.

$C_4 \Rightarrow$ 4 + ~~8~~ = 1P.

```
if(notPossible)
{
    start=mid+1
}
```



(5)



1|2|3|4.

[min]

$C_1 \Rightarrow$ 1 + 2 + ~~3~~. 2P.

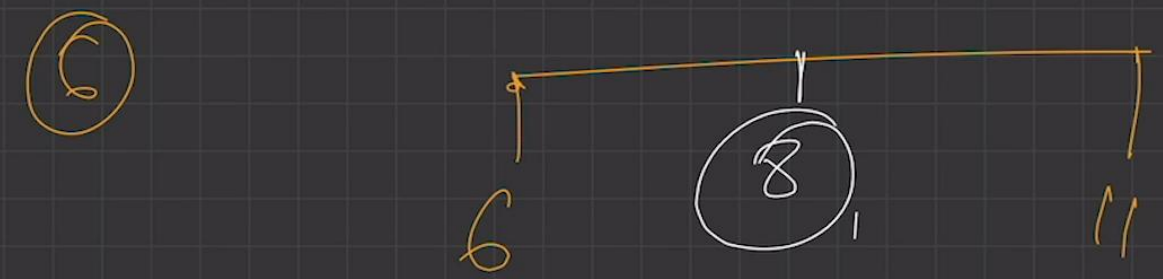
3min

$C_2 \Rightarrow$ 2 + ~~4~~ 1P.

$C_3 \Rightarrow$ 3 + ~~6~~ = 1P.

$C_4 \Rightarrow$ 4 + ~~8~~ = 1P.

```
if(notPossible)
{
    start=midt+1
}
```



8 min ??

1 | 2 | 3 | 4.

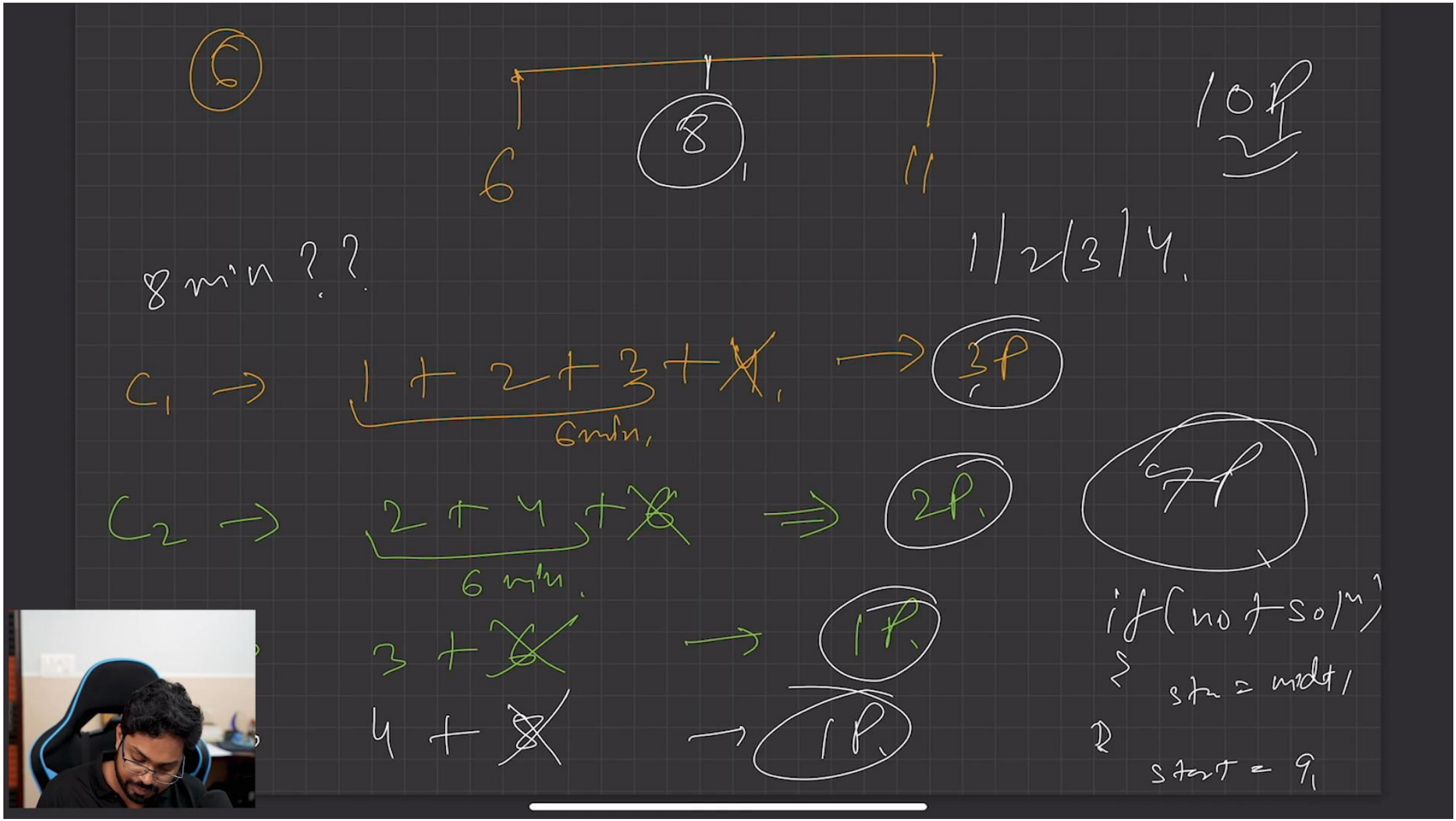
learn.thecodehelp.in
jhak6587@gmail.com
+919319347281

$C_1 \rightarrow$ 1 + 2 + 3 + ~~4~~ → 3P (6 min)

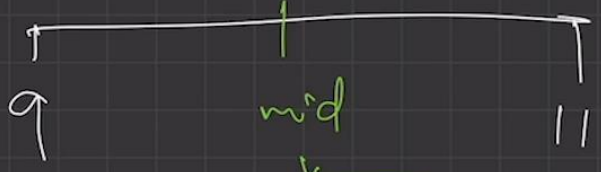
$C_2 \rightarrow$ 2 + 4 + ~~8~~ ⇒ 2P (6 min)

$C_3 \rightarrow$ 3 + ~~8~~ → 1P

$C_4 \rightarrow$ 4 + ~~8~~ → 1P


6
8
11
10 P
8 min ??
1|2|3|4.
C1 → 1 + 2 + 3 + X → 3P
6 min.
C2 → 2 + 4 + X ⇒ 2P.
6 min.
3 + X → 1P.
4 + X → 1P.
if(not solve)
{
start = mid+1;
}
start = 9.

(7)



10 P.

10 min ?

10 min.

1 | 2 | 3 | 4

$C_1 \rightarrow$ 1 + 2 + 3 + 4 + ✗ = 4 P. C.

9 min

$C_2 \Rightarrow$ 2 + 4 + ✗ $\Rightarrow$ 2 P.

6 m

3 + 6 + ✗ = 2 P.

9 min

$\Rightarrow$ (4) + ✗ $\rightarrow$ 1 P.

Not ✗

Start = 10

Start = 11

(8)

Minimum

2)

$C_1 \rightarrow 1+2+3+4 \rightarrow 4P$

$C_2 \rightarrow 2+4+6 = 2P$

$C_3 \rightarrow 3+6+9 = 2P$

$C_4 \Rightarrow 4+8 = 1P$

9P

(8)

11 11 11

2) 11 min.

$C_1 \rightarrow 1+2+3+4 \rightarrow 4P$

$C_2 \rightarrow 2+4+6 = 2P$

$C_3 \rightarrow 3+6+9 = 2P$

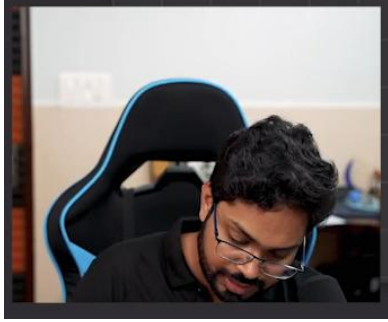
$C_4 \Rightarrow 4+8 = 1P$

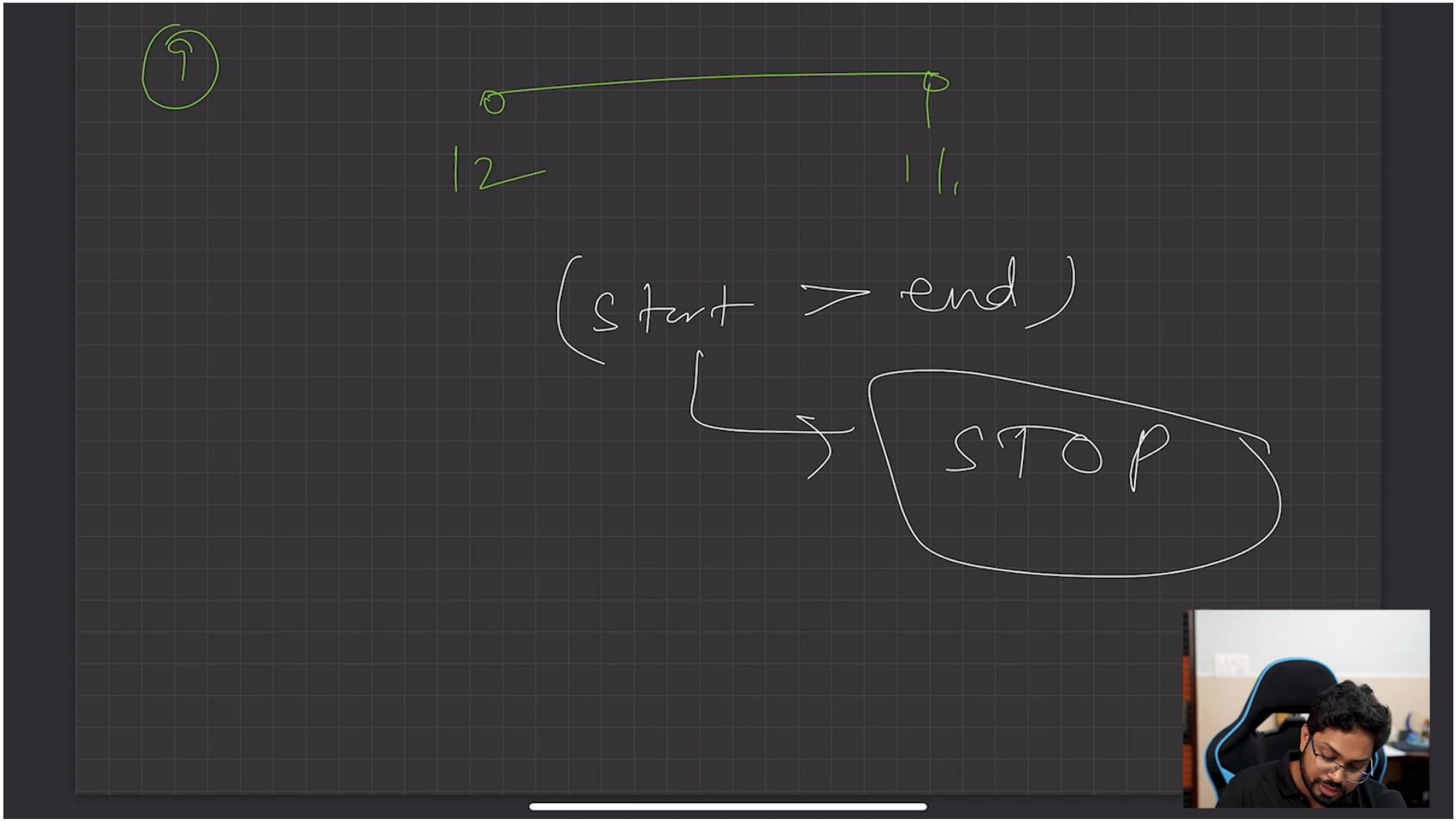
9P

2) Not a soln

start = mid+1

start = 12




1
12
11
(start > end)
STOP

Quick Compiler - CodeHelp La | SPOJ.com - Problem PRATA

labs.thecodehelp.in/quick-compiler

Gmail YouTube Maps

CODE HELP <LABS/>

Quick Compiler Beta Fruitbox Flex

RUN | C++ | File | Settings | SAVE | SHARE

SOURCE - main.cpp

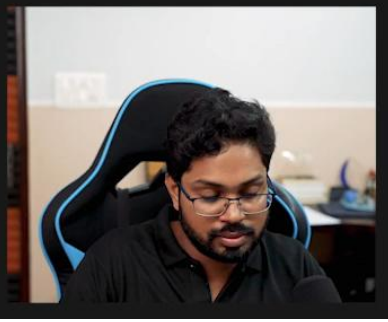
```cpp
#include <iostream>
using namespace std;


int main() {
  int T; cin >> T;
  while(T--){
    int nP, nC;
    cin >> nP >> nC;
    vector<int>cooksRanks;
    while(nC--){
      int R; cin>>R;
      cooksRanks.push_back(R);
    }

    cout<<minTimeToCompleteOrder(cooksRanks, nP)<<endl;
  }
  return 0;
}
```

STDIN

Your Input Goes Here ...

STDOUT

Your Output Will Be Displayed Here ...



end ⇒ 1 | 2 | 3 | 4

C_1 C_2 C_3 C_4

— (1) man ranker cook = 4 C_4.

(2) $nP = 10$ ↳ $1R + 2R + 3R \ldots$

$1R + 2R + 3R + \cdots + nPR$

⇒ $R\left[\frac{nP \times (nP+1)}{2}\right]$ ⇒ $R_{C_4}\left[\frac{nP*(nP+1)}{2}\right]$

R num.

Quick Compiler - CodeHelp La | SPOJ.com - Problem PRATA

labs.thecodehelp.in/quick-compiler

Gmail YouTube Maps

CODE HELP <LABS/>

Quick Compiler Beta Fruitbox Flex

RUN | C++ | File | Settings | SAVE | SHARE

SOURCE - main.cpp

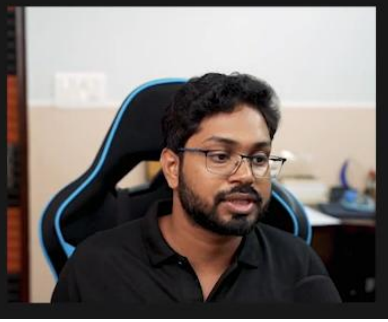
```cpp
#include <iostream>
using namespace std;

int minTimeToCompleteOrder(vector<int>cooksRanks, int nP){
  int start = 0;
  int highestRank = *max_element(cooksRanks.begin(), cooksRanks.end());
  int end = highestRank * (nP * (nP + 1) / 2);
  int ans = -1;

  while(start <= end){
    int mid = (start + end) >> 1;
    if(isPossibleSolution(cooksRanks, nP, mid)) {
      ans = mid;
      end = mid - 1;
    }
    else {
      start = mid + 1;
    }
  }
  return ans;
}

int main() {
  int T; cin >> T;
  while(T--){
```

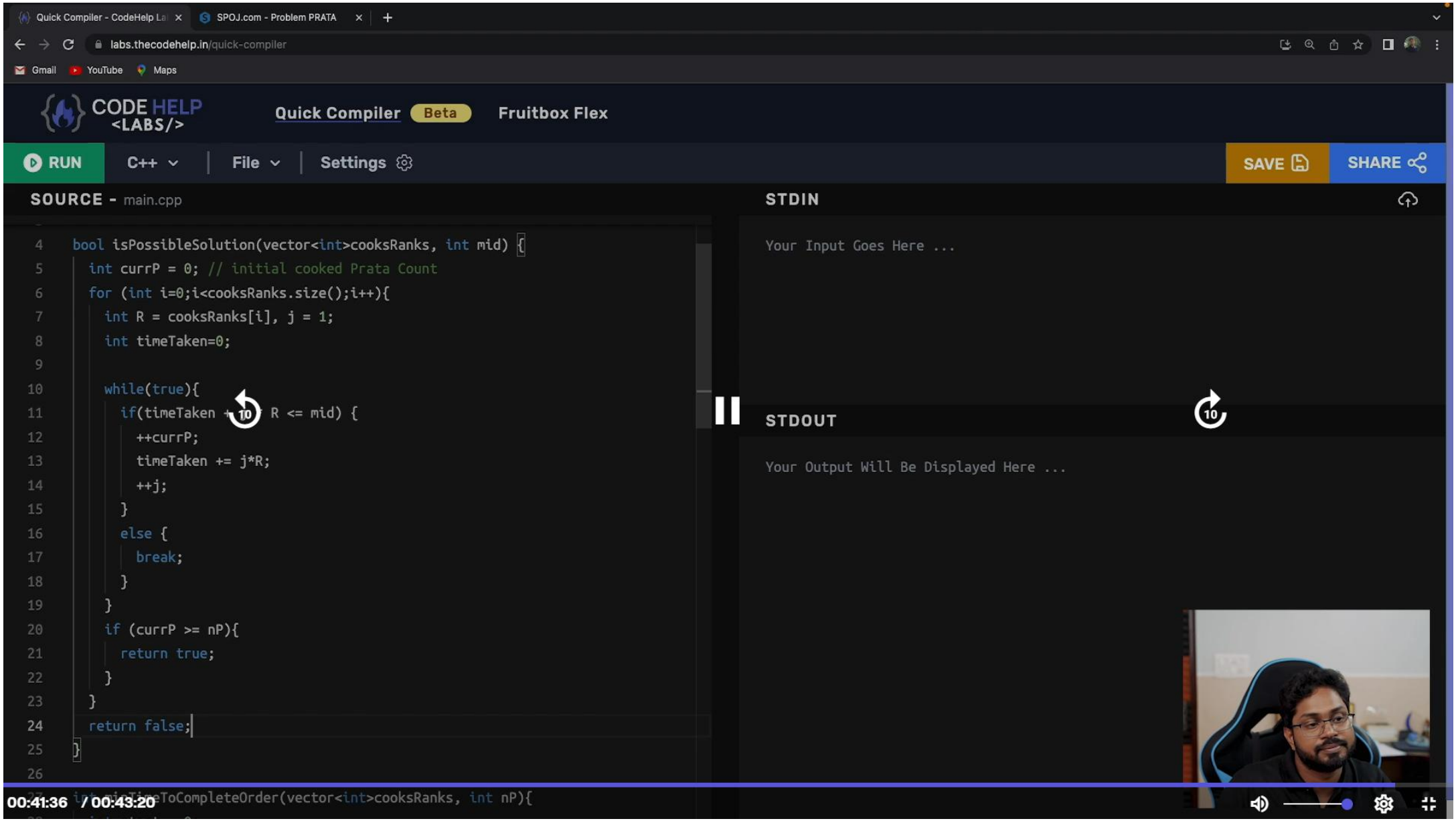
STDIN

Your Input Goes Here ...

STDOUT

Your Output Will Be Displayed Here ...




Quick Compiler - CodeHelp La
SPOJ.com - Problem PRATA
labs.thecodehelp.in/quick-compiler
Gmail
YouTube
Maps
CODE HELP <LABS/>
Quick Compiler
Beta
Fruitbox Flex
RUN
C++
File
Settings
SAVE
SHARE
SOURCE - main.cpp
STDIN
Your Input Goes Here ...
STDOUT
Your Output Will Be Displayed Here ...
bool isPossibleSolution(vector<int>cooksRanks, int mid) {
int currP = 0; // initial cooked Prata Count
for (int i=0;i<cooksRanks.size();i++){
int R = cooksRanks[i], j = 1;
int timeTaken=0;
while(true){
if(timeTaken + j*R <= mid) {
++currP;
timeTaken += j*R;
++j;
}
else {
break;
}
}
if (currP >= nP){
return true;
}
}
return false;
}
int minTimeToCompleteOrder(vector<int>cooksRanks, int nP){
00:41:36 / 00:43:20