

# Data Science & AI & NIC - Param

Python-For Data Science

Binary Tree

Lecture No.- 04

By- Pankaj Sharma Sir



# Recap of Previous Lecture



Topic

Trees Part 03





# Topics to be Covered



Topic

Trees Part 04





## Topic : Trees



Maths  
↗

1) File handling

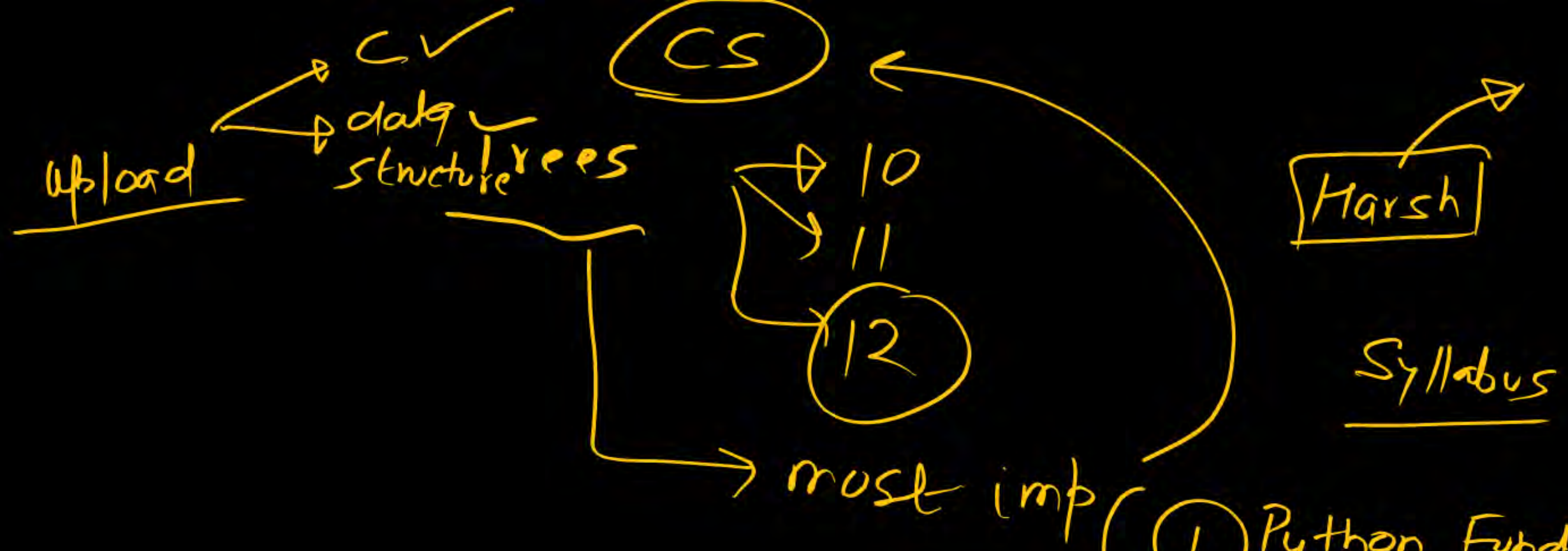
2) Numpy

3) Pandas

Numpy ✓✓





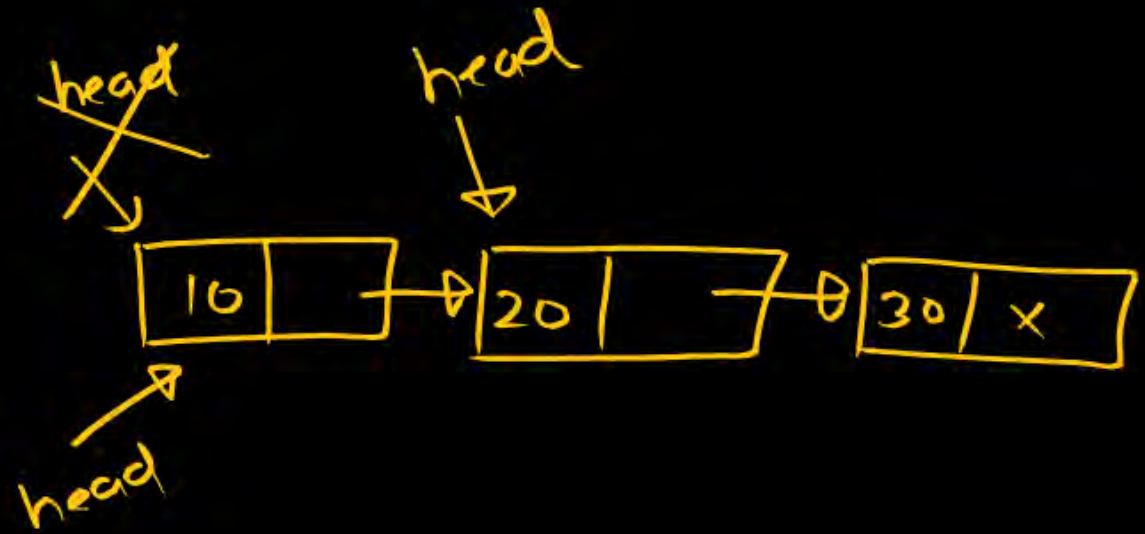


- CS Gate
- ① C ✓
  - ② DS ✓
- type of Questions
- ① Python Fundamentals
  - ② OOPS
  - ③ Data structure

head  
→ None

```
def delete_First(head):  
    if head is None:  
        return
```

Doubt



delete\_First(head)

bal factor  
↳ -1, 0, 1

Given a binary tree, check whether it is  
balanced or not?

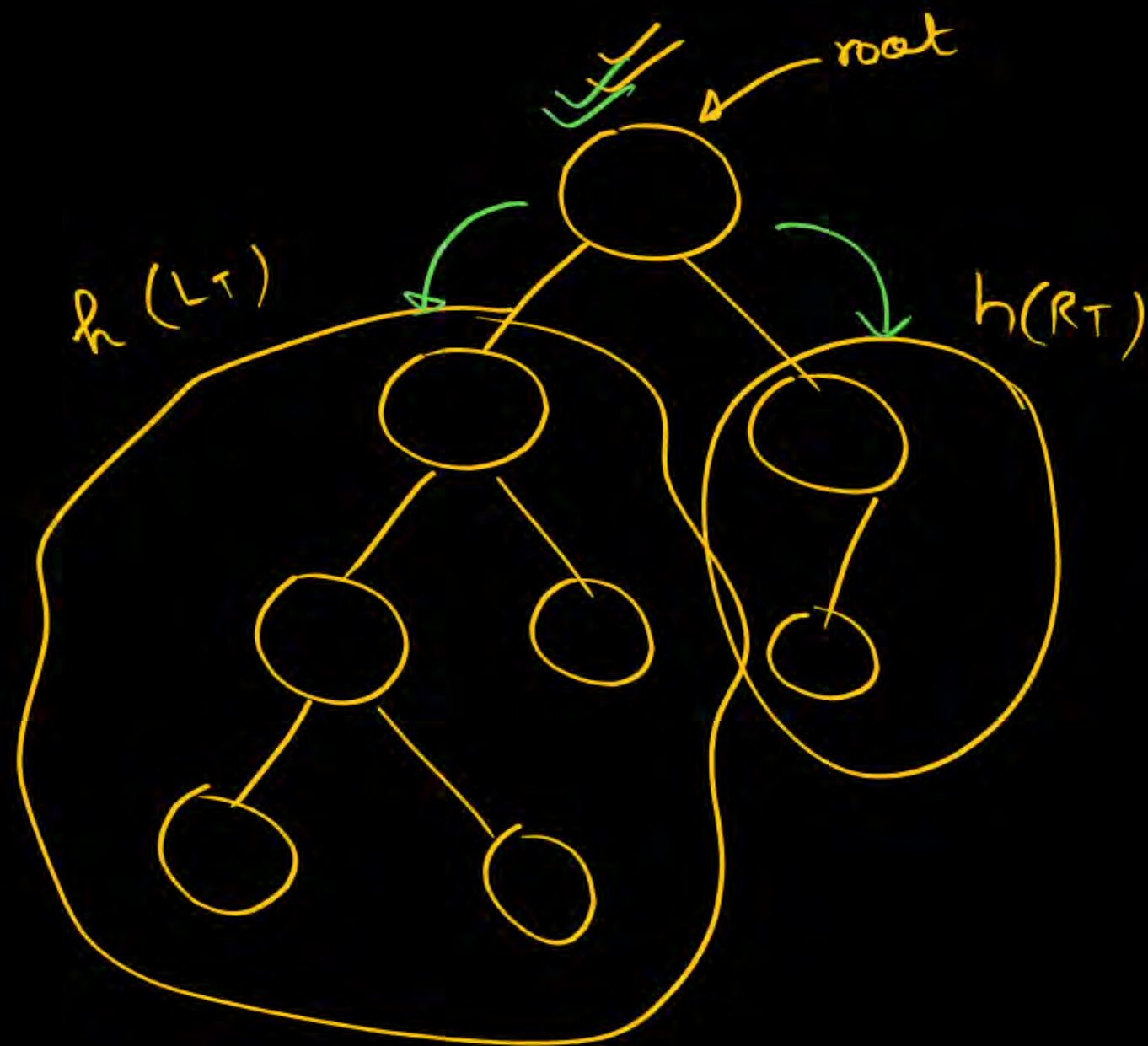
```
def height(root):
```

```
    if root is None:
```

```
        return 0
    if root is None:
```

```
        diff = height(root.left)
```

```
        - height(root.right)
    if diff in [-1, 0, 1]:
```

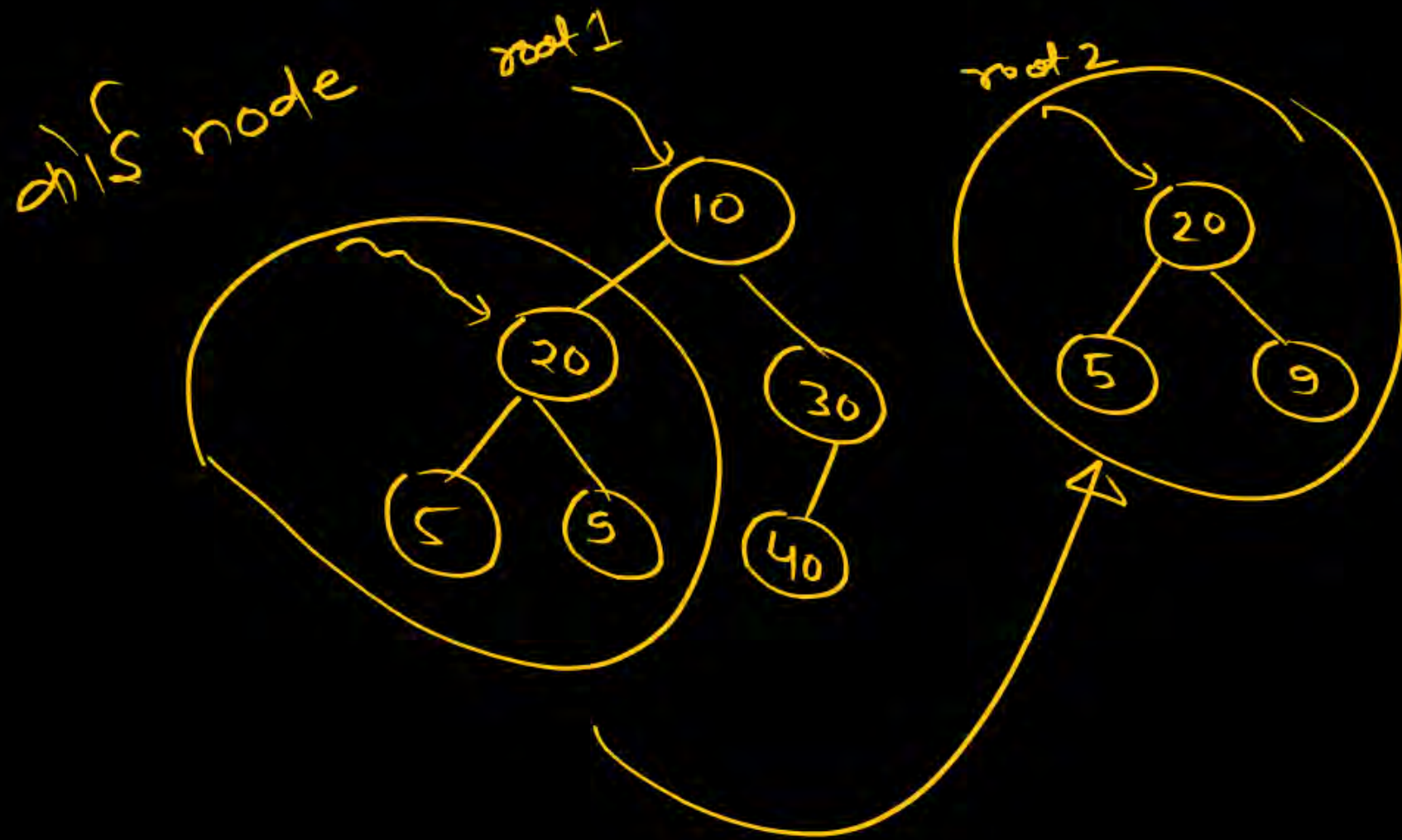




$T_1, T_2$

$root_1, root_2 \Rightarrow root_2$   
is a subtree  
in  
 $root_1$

Code X  $\leftarrow$   
method  $\checkmark$

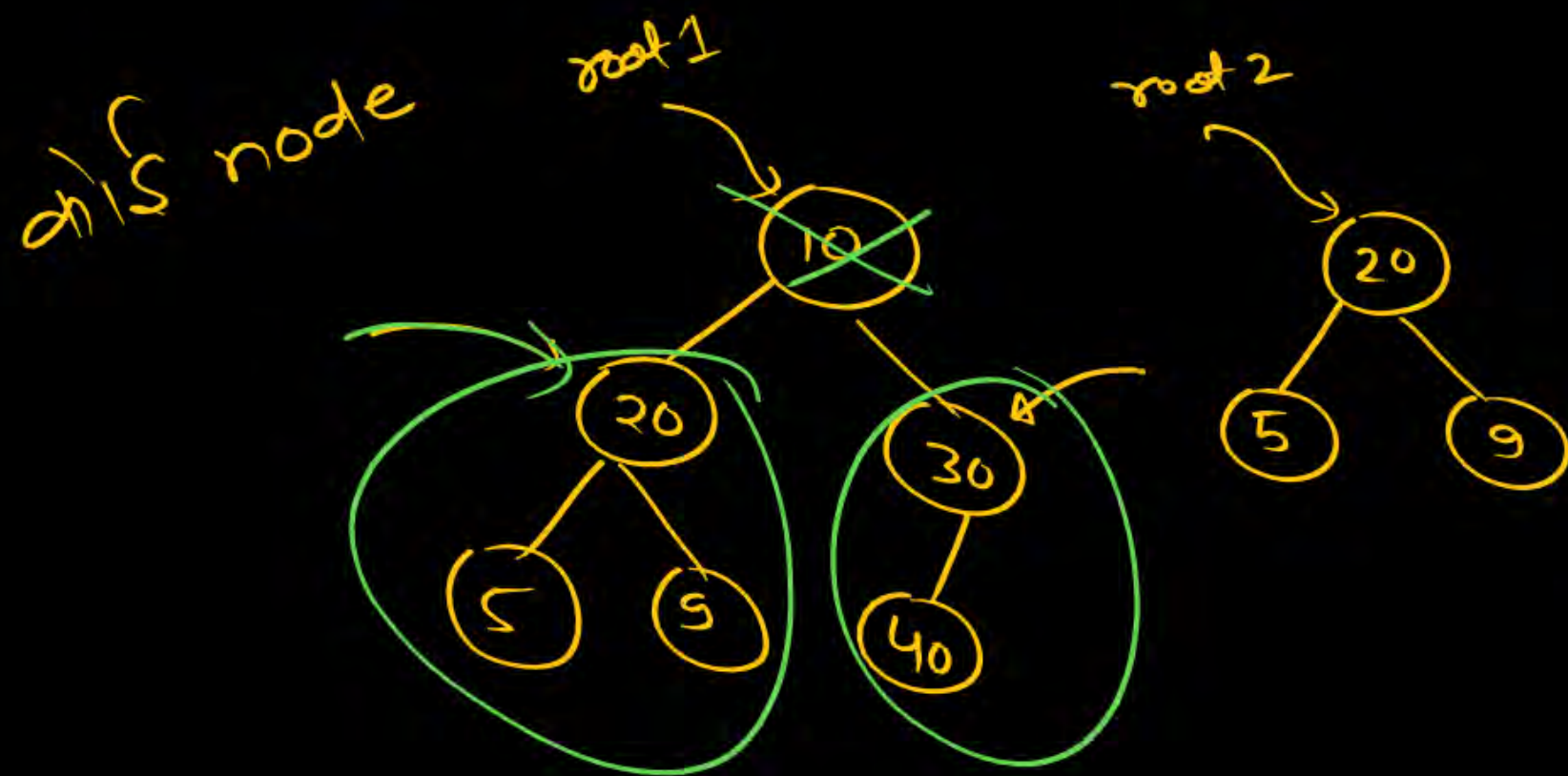


$\Rightarrow$  Yes



$T_1, T_2$

$root_1, root_2 \Rightarrow root_2$   
is a subtree  
in  
 $root_1$



```
def isSubtree(self, root1, root2):  
    if root2 is None:
```

```
        return True
```

```
    if root1 is None:
```

```
        return False
```

```
    if self.isIdentical(root1, root2):  
        return True
```

```
    Ans1 = self.isSubtree(root1.left, root2)  
    Ans2 = self.isSubtree(root1.right, root2)  
    return Ans1 or Ans2
```

# Python

## Fundamental types

- (i) bool
- (ii) int
- (iii) complex
- (iv) string
- (v) float

## Type-casting

bool()  
int()  
complex()  
str()  
float()

## built in

- (i) bytes
- (ii) bytearray } x
- (i) set
- (ii) dict
- (iii) List
- (iv) tuple
- (v) range }  
vi) frozenset x



decision

if

elif

else

loops

for i in range(10):



for

:

else :

while condition :



continue →

break →

Nested loop → pattern printing

List → methods  
programs

mutable

ordered

✓ slicing

concat

$l1 + l2$

len()

1-D list

$l =$

[ i for i in range(s) ]

List comprehension

2-D list



## Tuple

\* same as list but read only

seperate  $\Rightarrow$  \* mutability, Aliasing, cloning

copy

## string

\* slicing

\* -ve index

$s[:: -1]$

\*

filter  
map  
reduce

functions  
Anonymous func.  
lambda function

H.W

zip()

enumerate()

dict

\* freq. of a word  
in a string

\* Characters

get( )

d[ ] →

→ default

OOPS

Object  
class

reference variable

type of var.

4  
3

instance var.  
class level  
local var.

type of method

instance method  
static method  
class method

inheritance

① Public

② Private --

Polymorphism

method overriding

operator overloading

Exception handling

magic  
methods

Object class

-- new --

-- str --

-- init --

Encapsulation

Abstraction



Abstract class

→ import abc

↳ ABC

abstract method

## Linked list

- \* Create → Implementation
- \* Input
- \* Print
- \* Insertion 

at begin  
at end
- \* last\_node - data
- \* Sec. - last
- \* Reverse
- \* Search
- \* Types of linked list
- \* Deletion

- \* Stack → Implementation
- \* application
- \* TOH
- \* == balanced parenthesis

## Queue

- \* Implementation
- ## Tree

Binary Tree →

BST →

AVL →

Interviews  
DS

→ upload



**THANK - YOU**