CS 112

Week11: Trees

General Trees

General Trees

- This week we discuss one of the most important non linear data structures in computing, trees.

General Trees

- This week we discuss one of the most important non linear data structures in computing, trees.

- Trees are indeed a breakthrough in data organization, allowing us to implement a host of algorithms much faster than when using linear data structures such as arrays or lists.

General Trees

- This week we discuss one of the most important non linear data structures in computing, trees.

- Trees are indeed a breakthrough in data organization, allowing us to implement a host of algorithms much faster than when using linear data structures such as arrays or lists.

- Trees provide a natural organization for data, and are ubiquitous structures in file systems, graphical user interface, databases, web sites and other computer systems.

General Trees

- This week we discuss one of the most important non linear data structures in computing, trees.

- Trees are indeed a breakthrough in data organization, allowing us to implement a host of algorithms much faster than when using linear data structures such as arrays or lists.

- Trees provide a natural organization for data, and are ubiquitous structures in file systems, graphical user interface, databases, web sites and other computer systems.

-

Non-linear?

Non-linear?

- When we say that trees are non linear we are referring to an organizational relationship that is richer than the simple “before” and “after” relationships between object in sequences.

Non-linear?

- When we say that trees are non linear we are referring to an organizational relationship that is richer than the simple “before” and “after” relationships between object in sequences.
- The relationships in tree are hierarchical, with some objects being “above” and some “below” others.

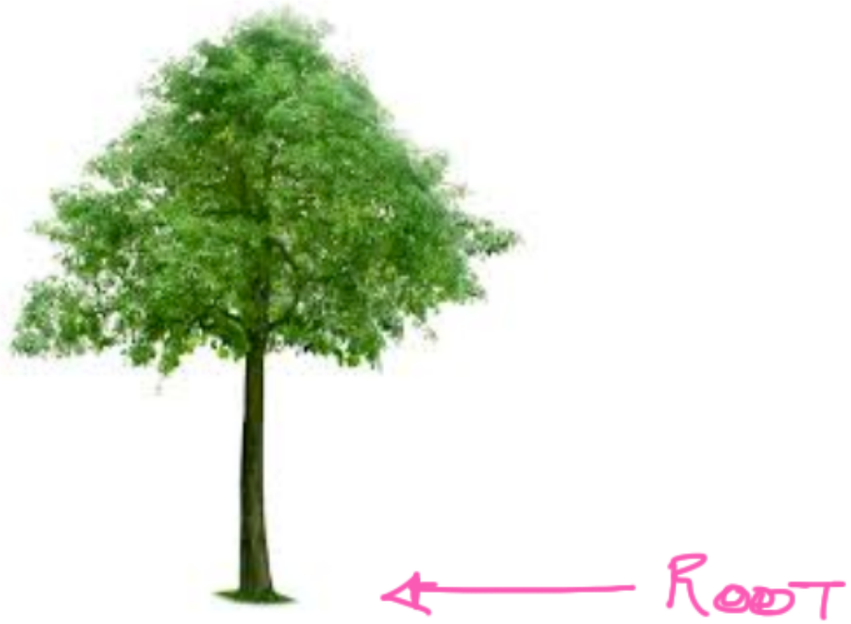
Non-linear?

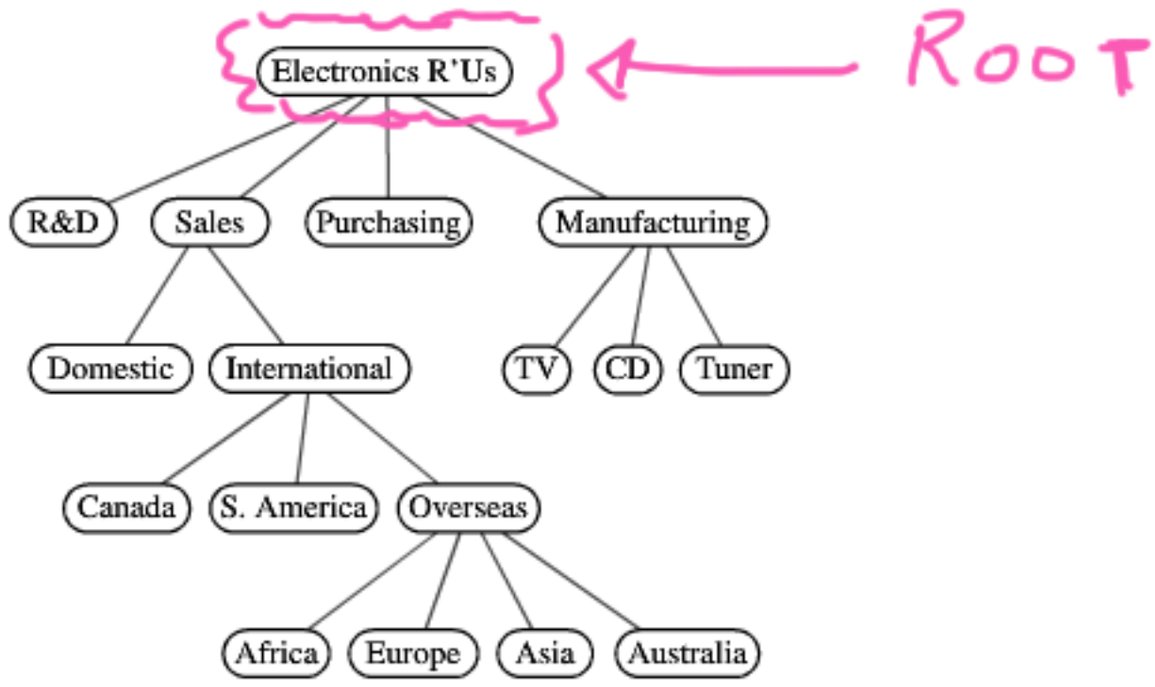
- When we say that trees are non linear we are referring to an organizational relationship that is richer than the simple “before” and “after” relationships between object in sequences.

- The relationships in tree are hierarchical, with some objects being “above” and some “below” others.

- Actually the main terminology for tree data structures comes from family trees, with the terms “parent”, “child”, “ancestor” and “descendants” being the most common words used to describe relationships.

Tree


Root


Electronics R'Us
Root
R&D
Sales
Purchasing
Manufacturing
Domestic
International
TV
CD
Tuner
Canada
S. America
Overseas
Africa
Europe
Asia
Australia

Tree definitions and properties

Tree definitions and properties

- A tree is an abstract data type that stores elements hierarchically.

Tree definitions and properties

- A tree is an abstract data type that stores elements hierarchically.

- With the exception of the top element, each element in a tree has a parent element and zero or more children.

Tree definitions and properties

- A tree is an abstract data type that stores elements hierarchically.

- With the exception of the top element, each element in a tree has a parent element and zero or more children.

- A tree is usually visualized by placing elements inside ovals or rectangles and by drawing the connections between parents and children with straight lines.

Tree definitions and properties

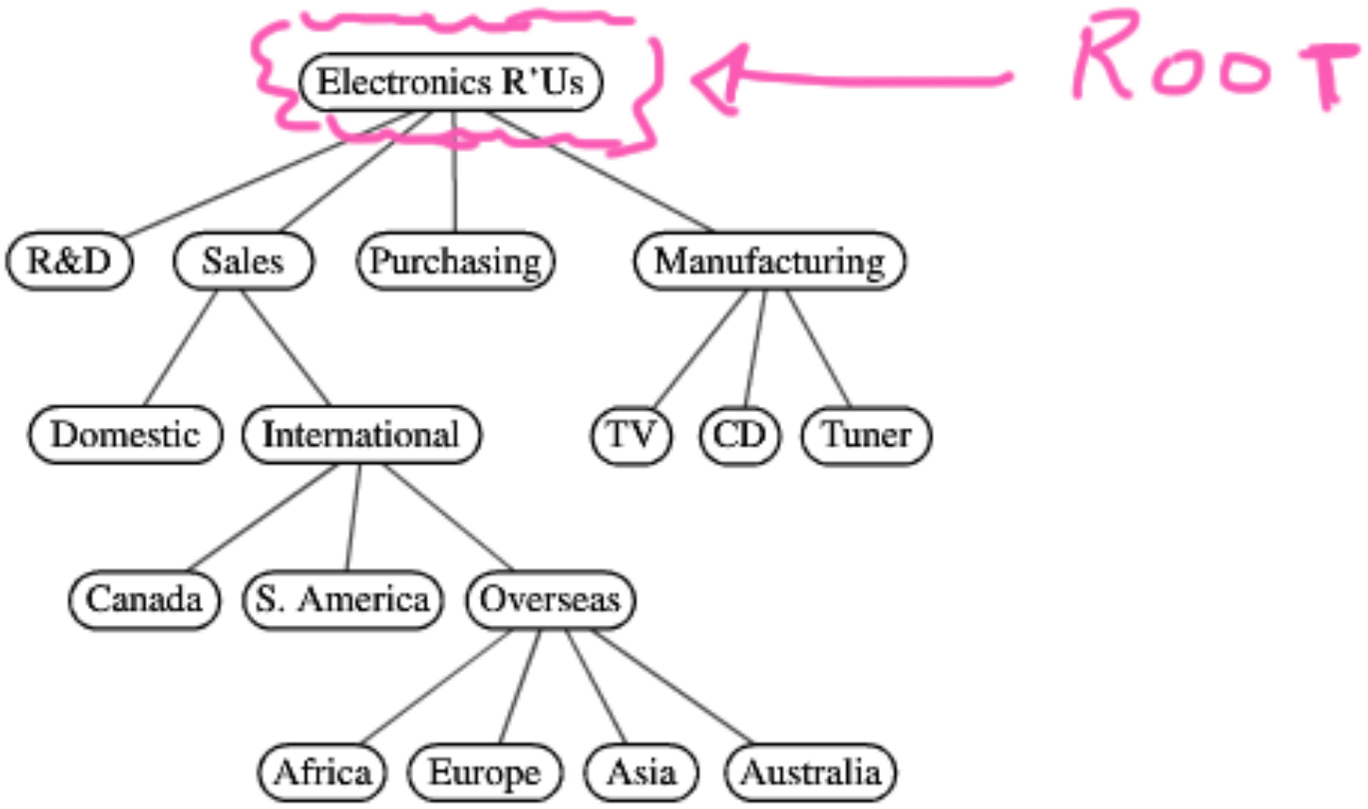
- A tree is an abstract data type that stores elements hierarchically.

- With the exception of the top element, each element in a tree has a parent element and zero or more children.

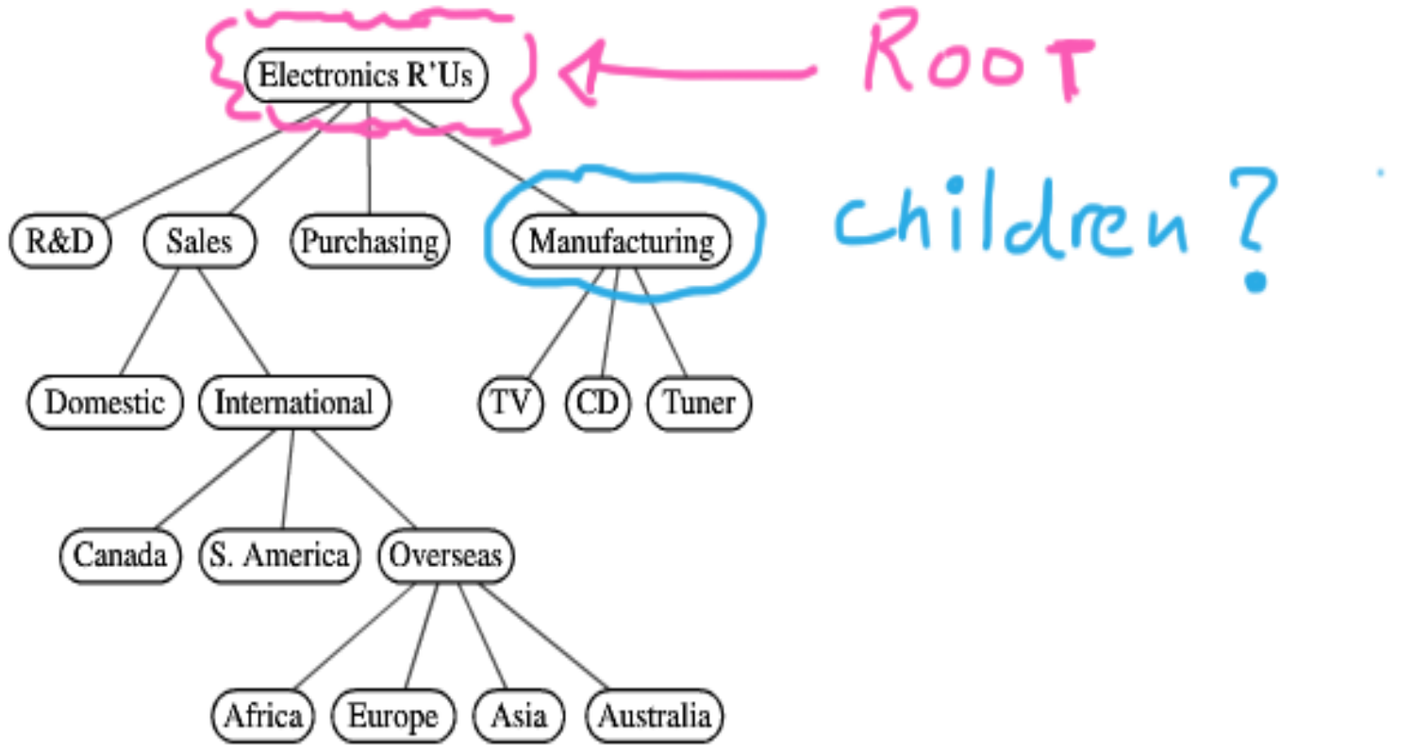
- A tree is usually visualized by placing elements inside ovals or rectangles and by drawing the connections between parents and children with straight lines.

- We typically call the top element the root of the tree, but it is drawn as the highest element, with the other elements being connected below (just the opposite of the real tree picture shown on the previous slide).

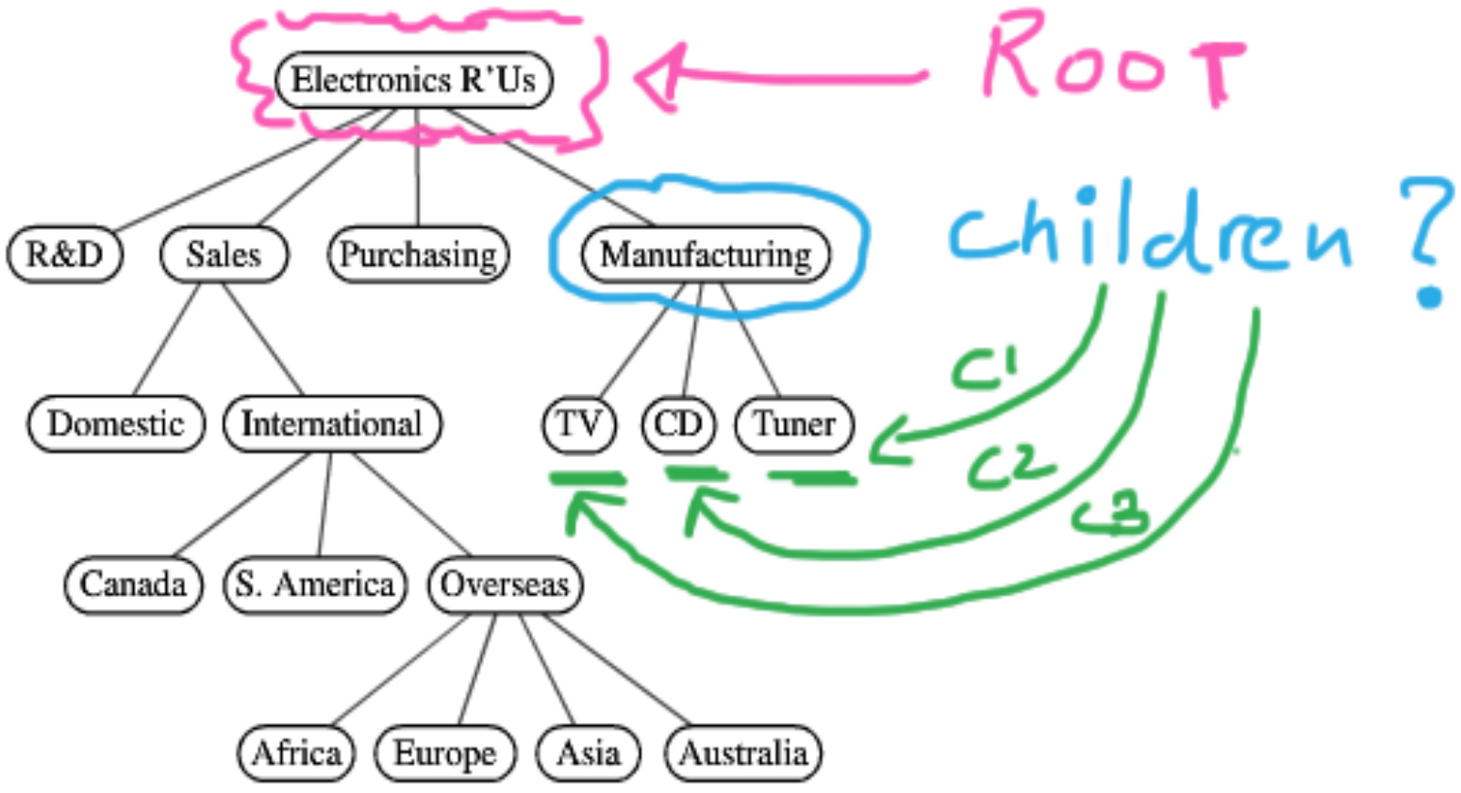
Tree definitions and properties



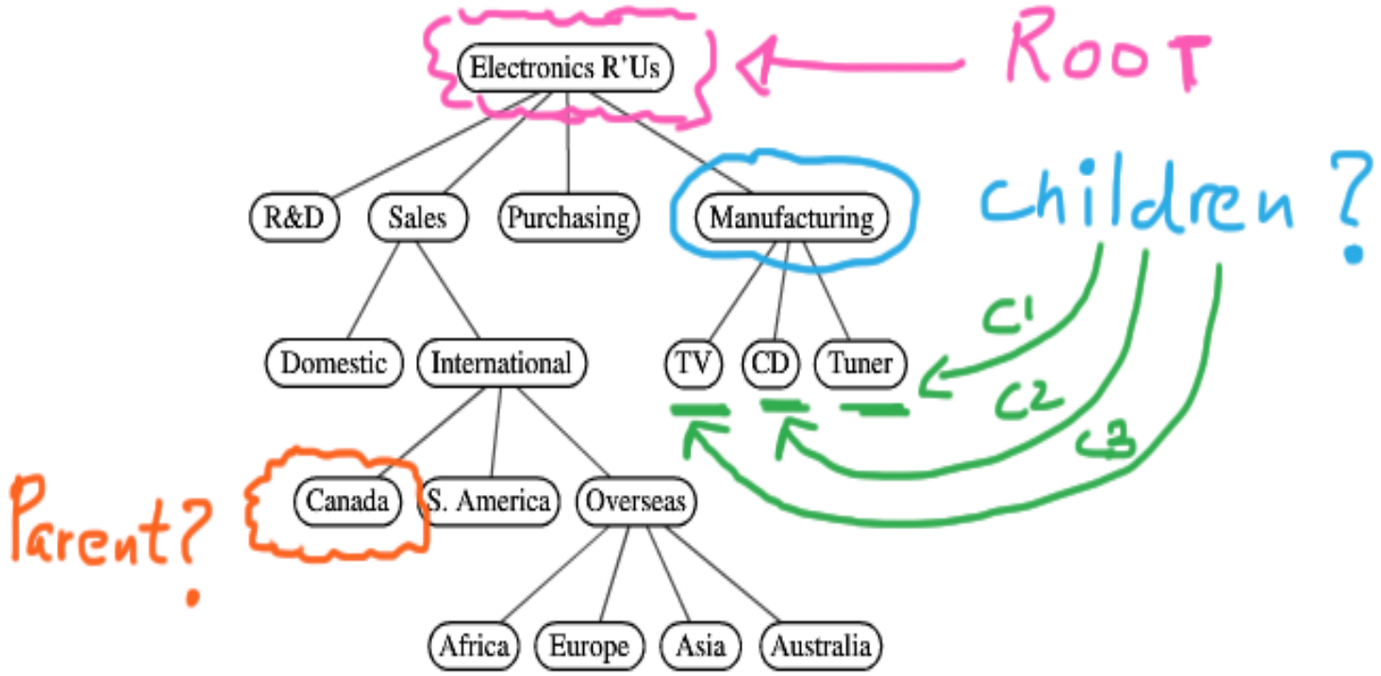
Tree definitions and properties



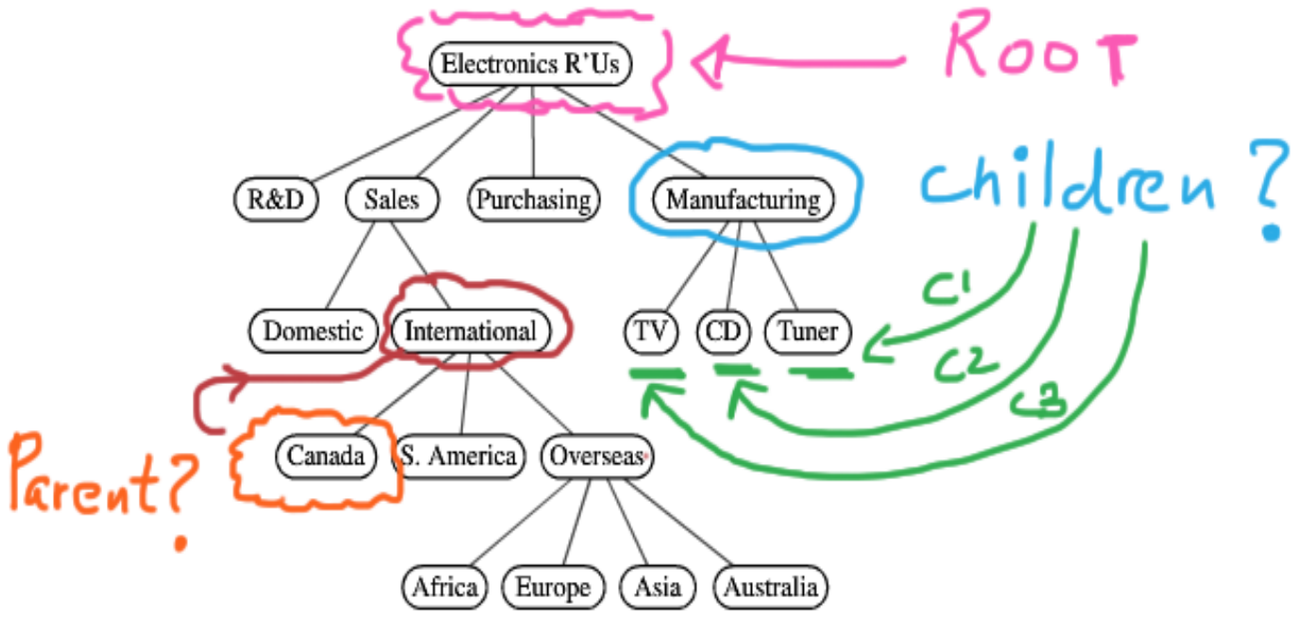
Tree definitions and properties



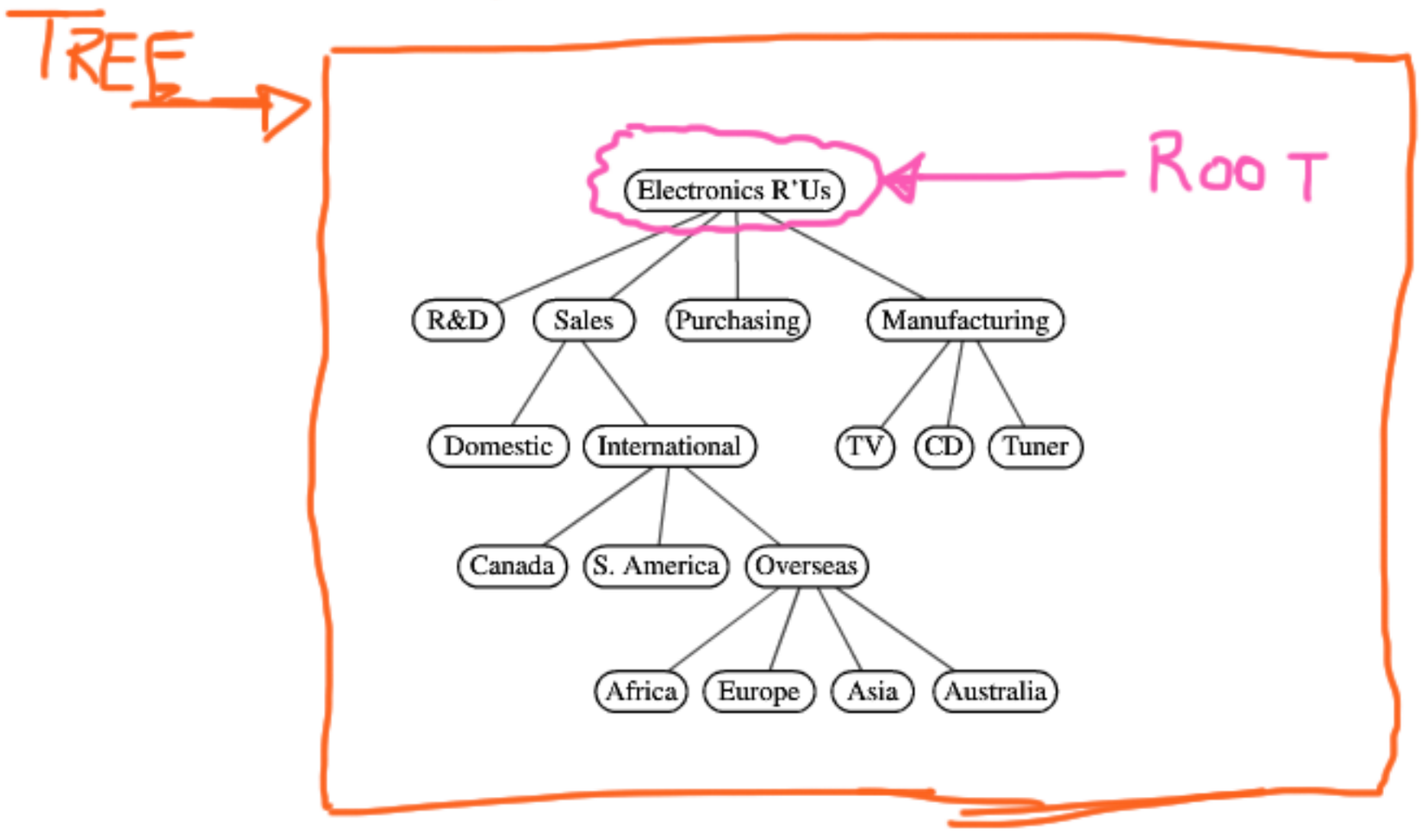
Tree definitions and properties



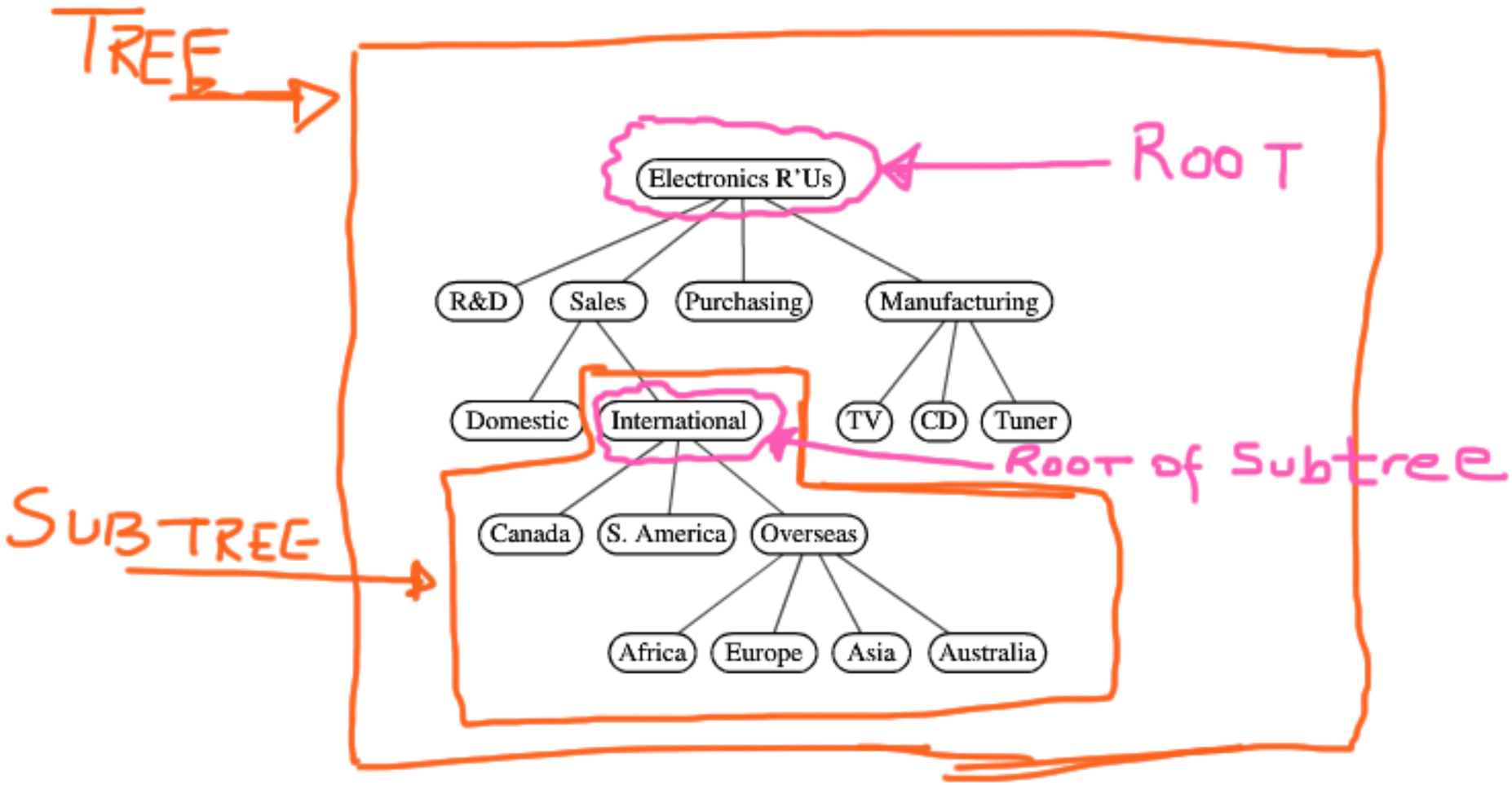
Tree definitions and properties



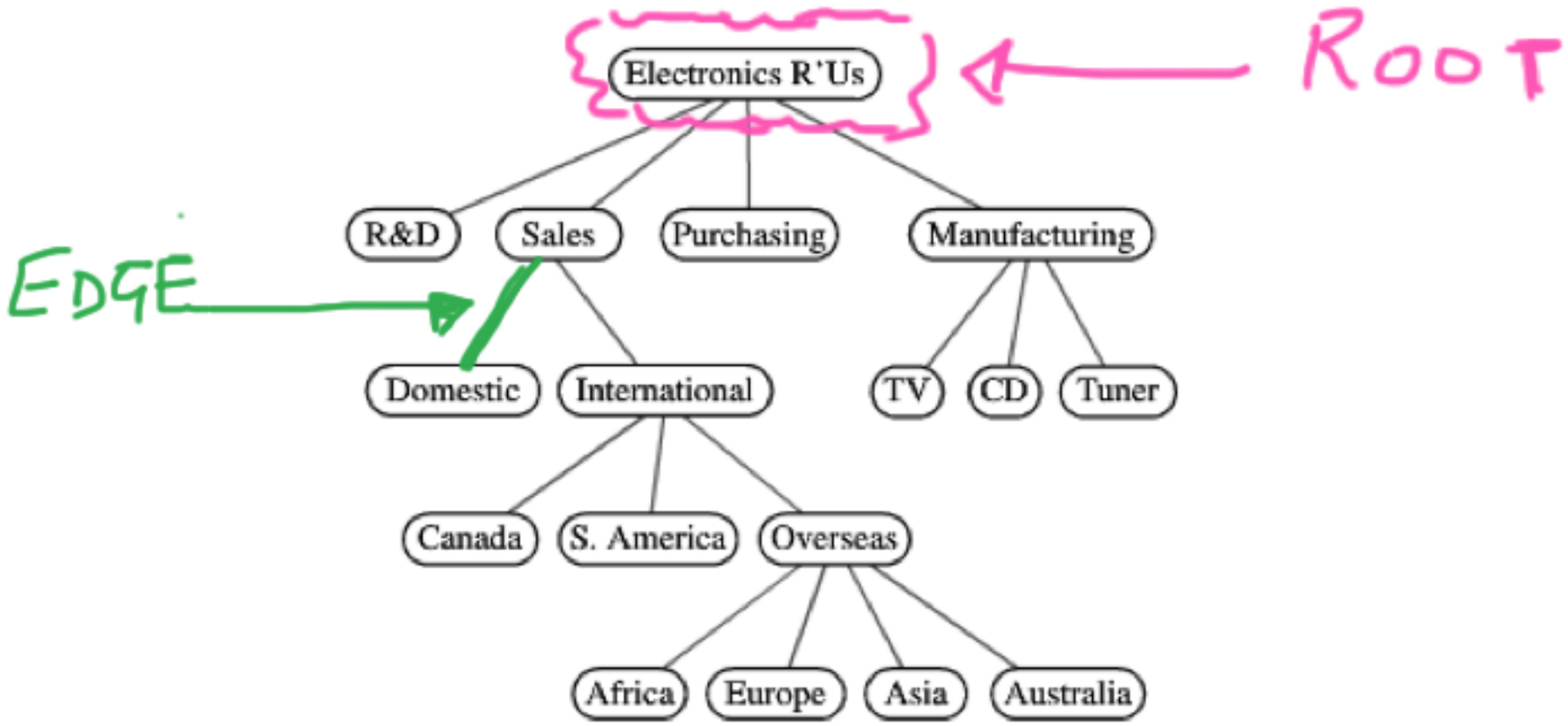
Tree is a recursive data structure.



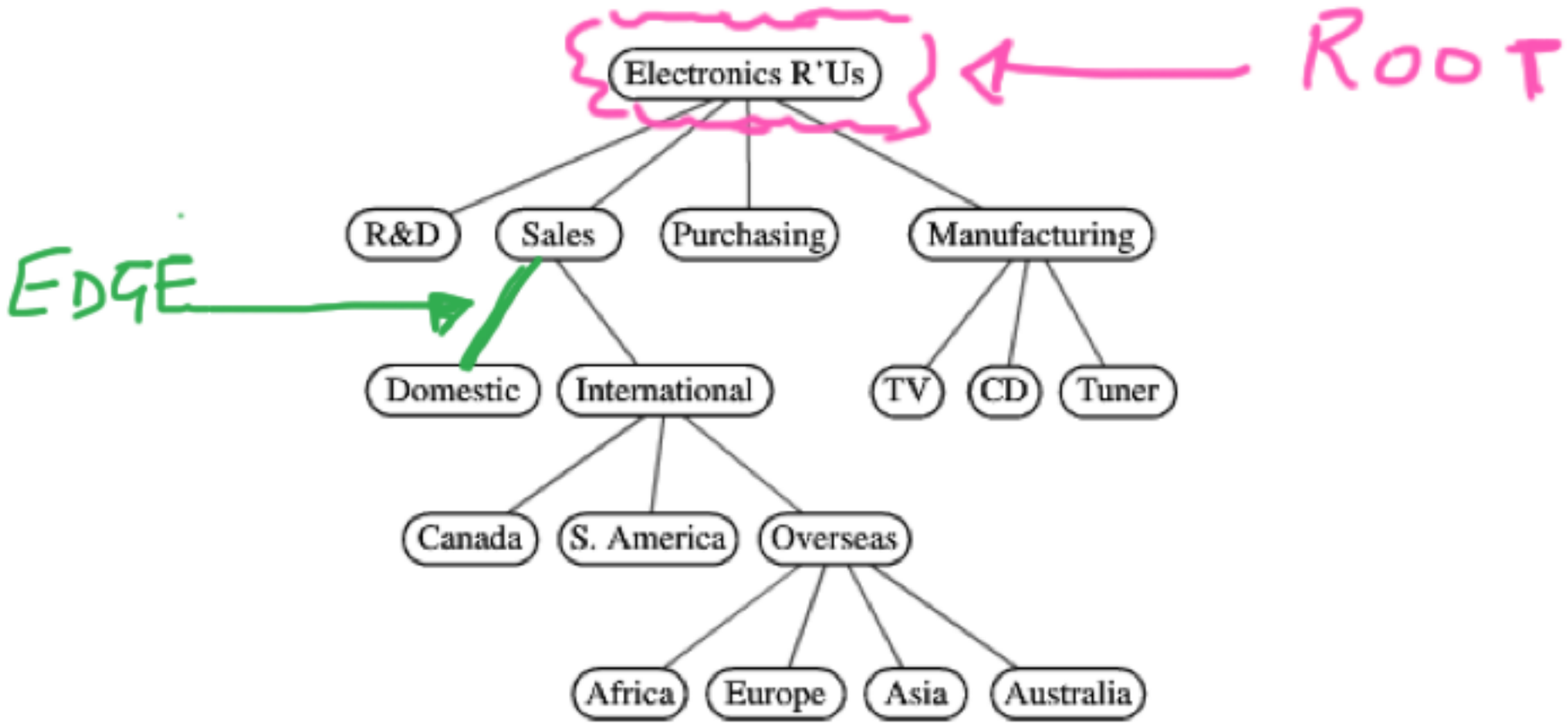
Tree is a recursive data structure.



Edge

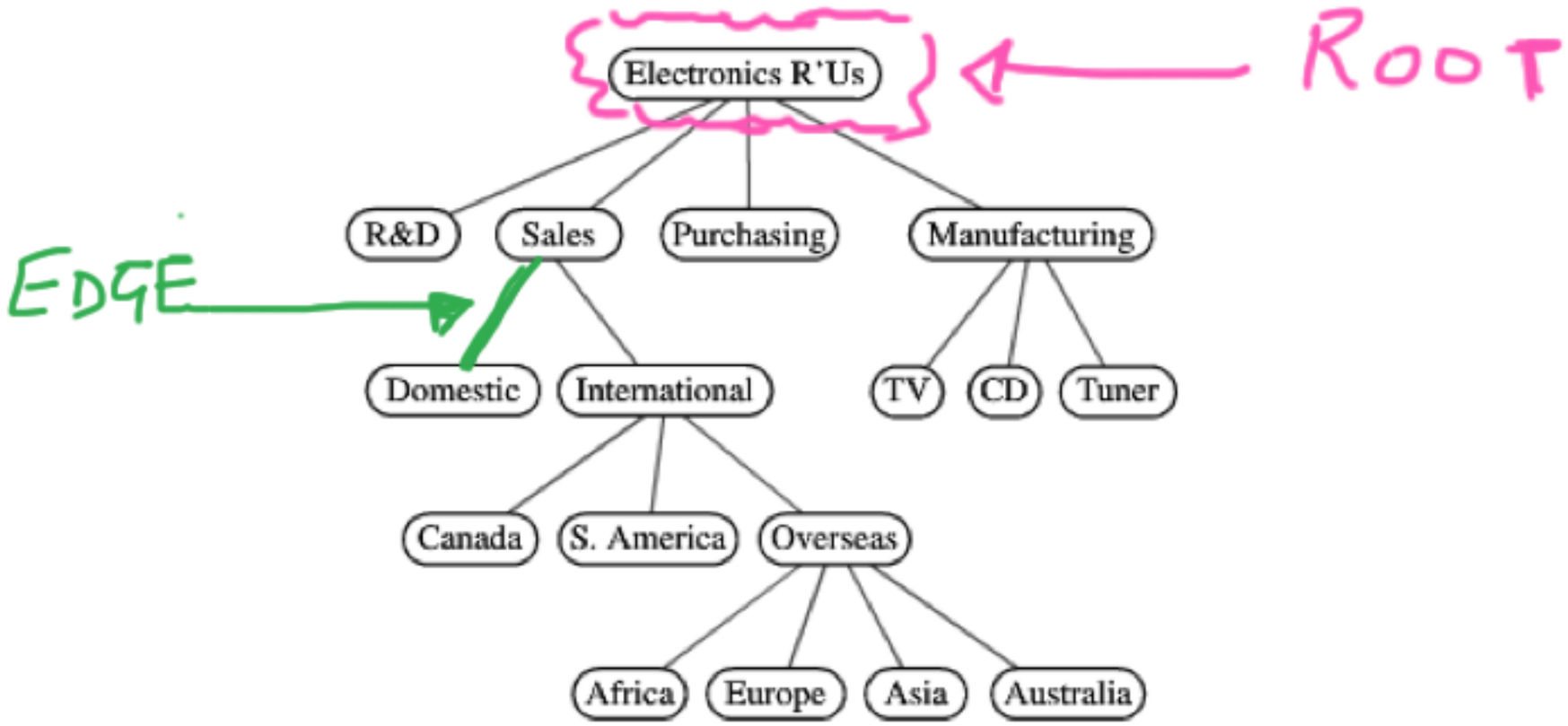

Electronics R'Us
ROOT
EDGE
R&D
Sales
Purchasing
Manufacturing
Domestic
International
TV
CD
Tuner
Canada
S. America
Overseas
Africa
Europe
Asia
Australia

Edge



- An edge of a tree is a pair of nodes (u,v) such that u is the parent of v or vice versa. In the above example, the green line is an edge.

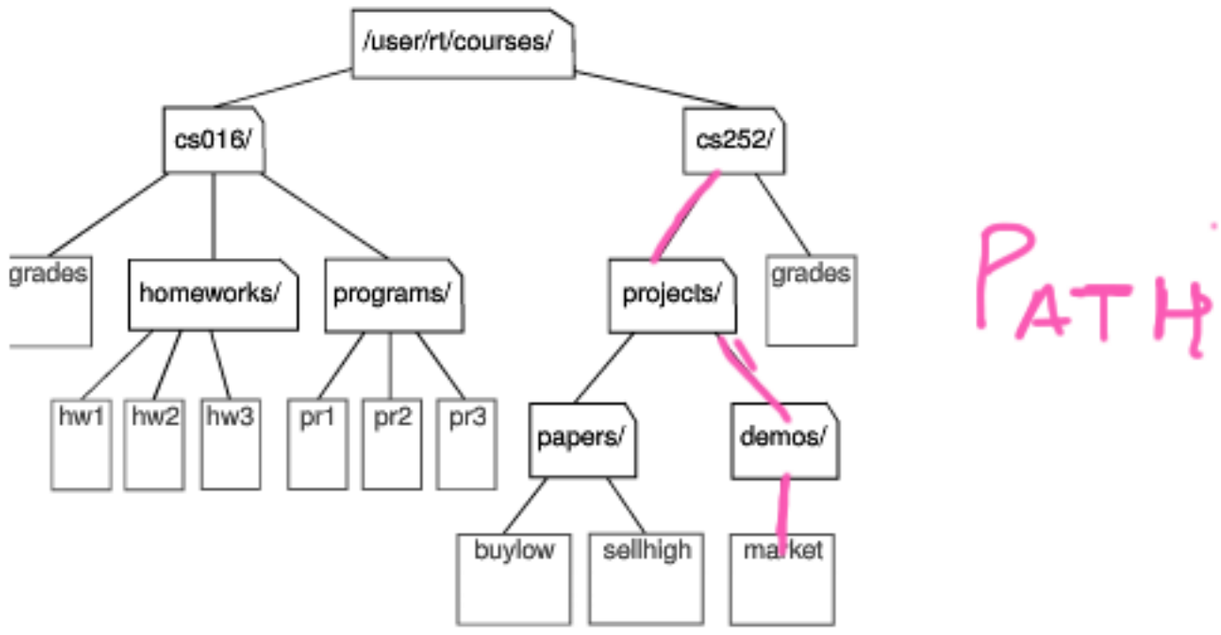
Edge



- An edge of a tree is a pair of nodes (u,v) such that u is the parent of v or vice versa. In the above example, the green line is an edge.

- It is an edge between (Sales, Domestic)

Path



PATH

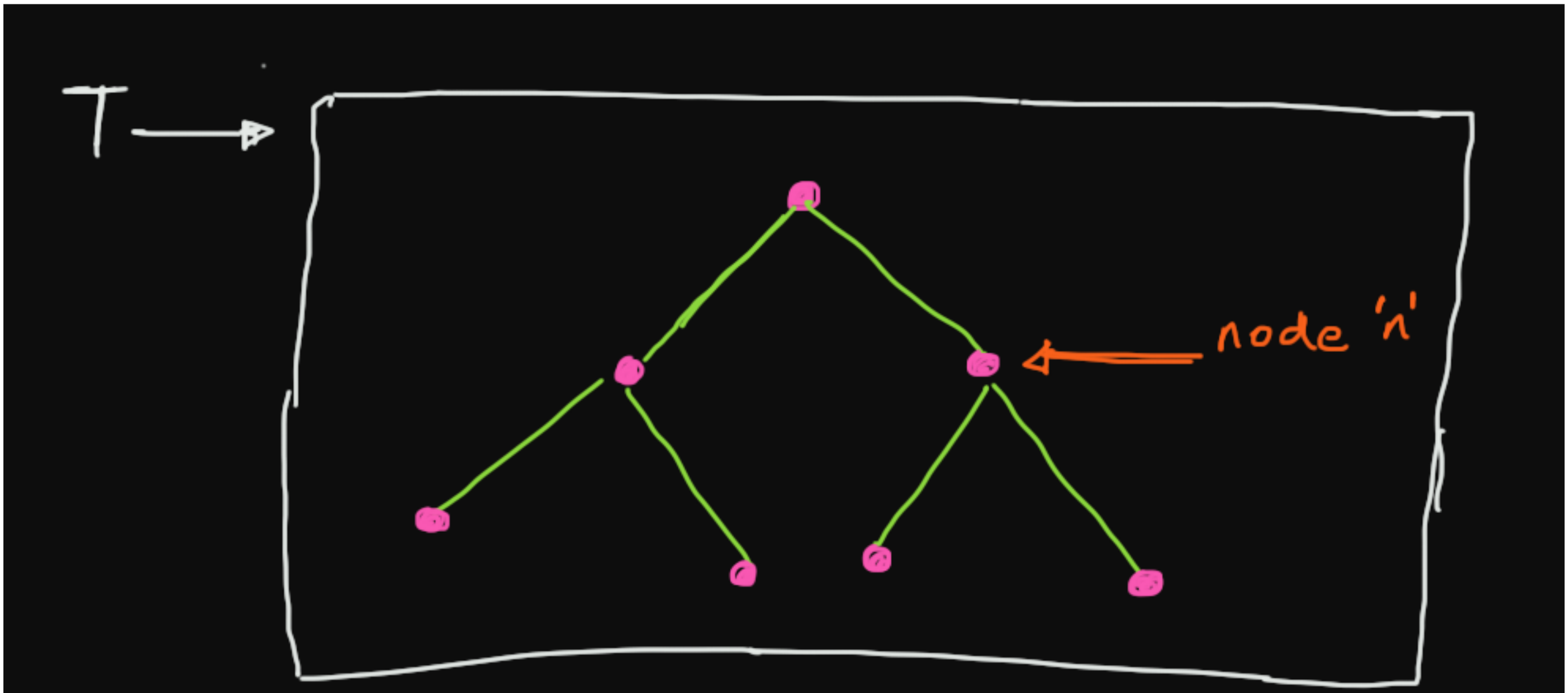
- A path of a tree is a sequence of nodes such that any two consecutive nodes in the sequence form an edge. For example in the picture above, the path in pink is highlighted.

- (CS252/,projects/,demos/,market).

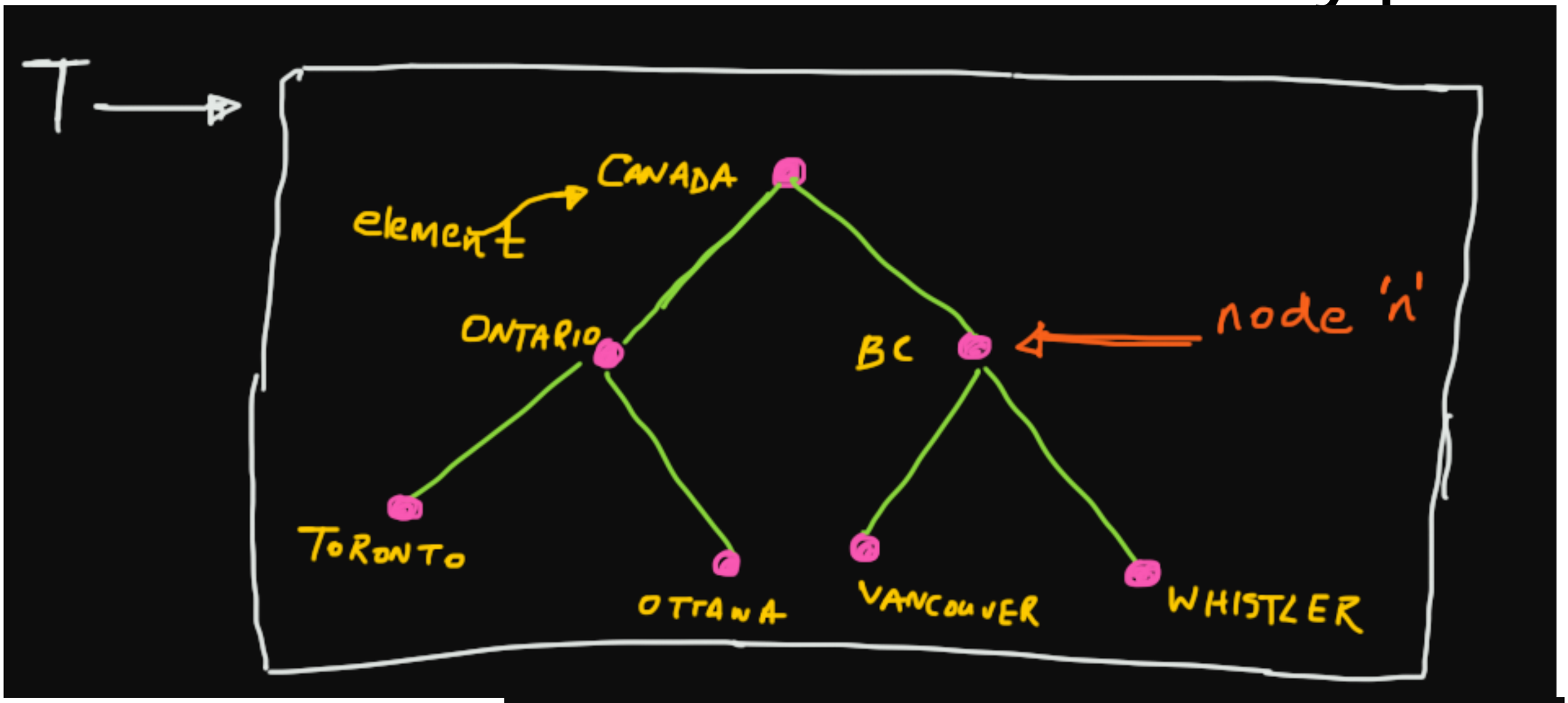
Tree Abstract Data Type

- Lets assume that ’T’ refers to some Tree.

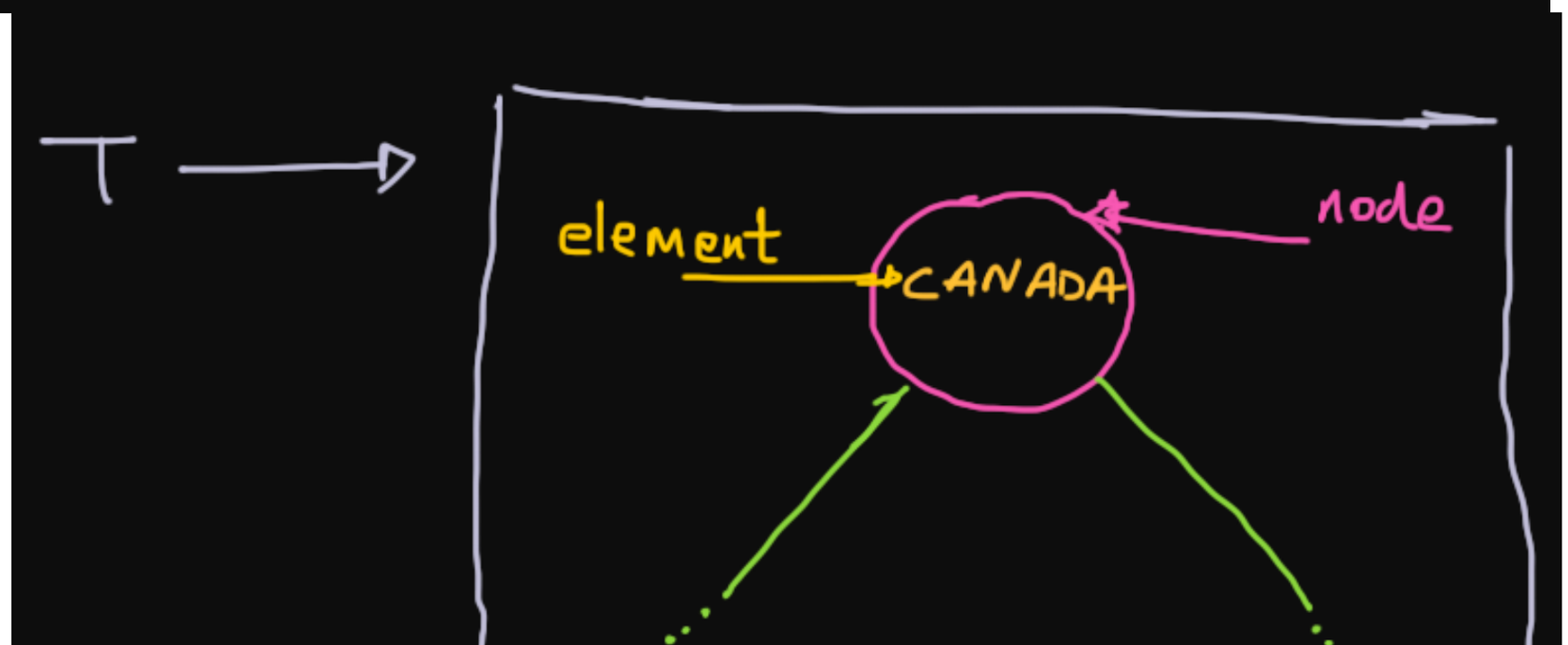
- Lets assume that ’n’ refers to some node in the Tree ’T’.



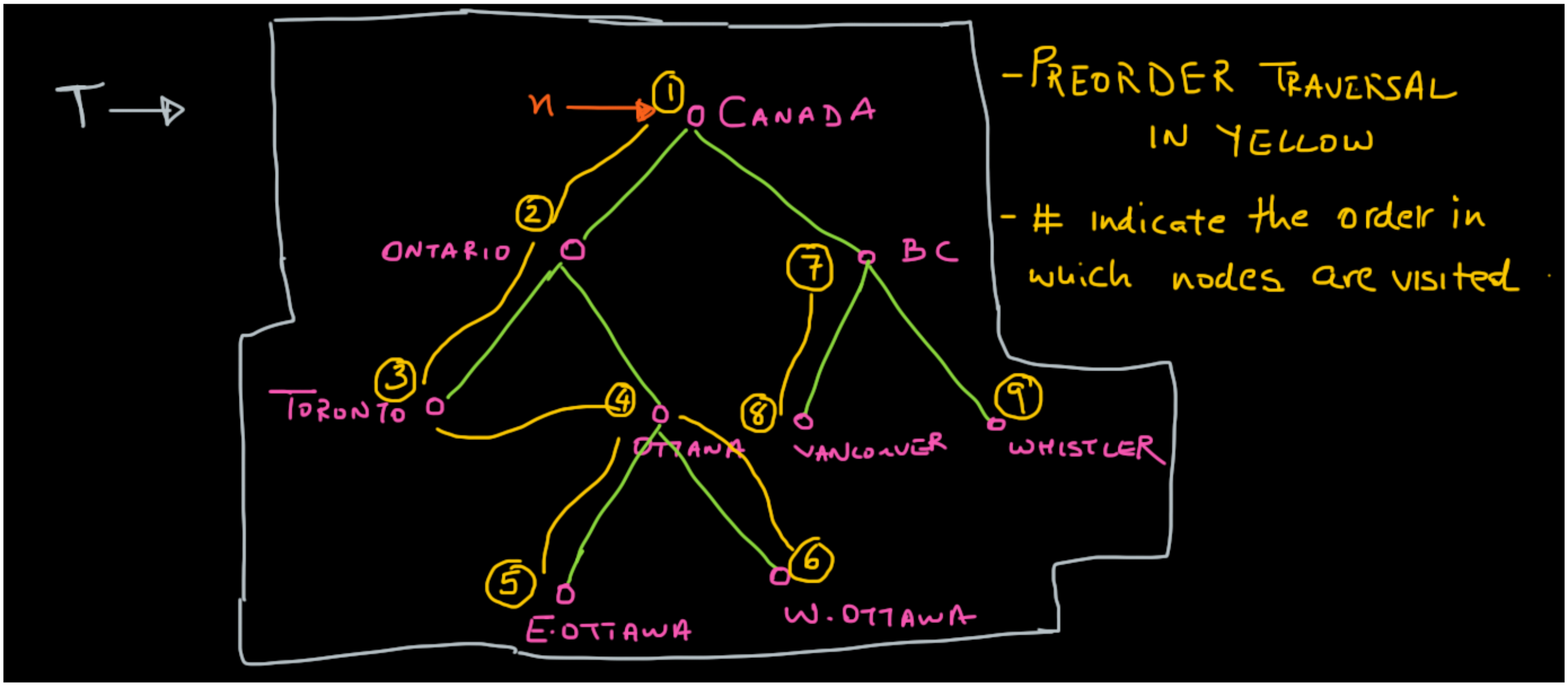
Tree Abstract Data Type



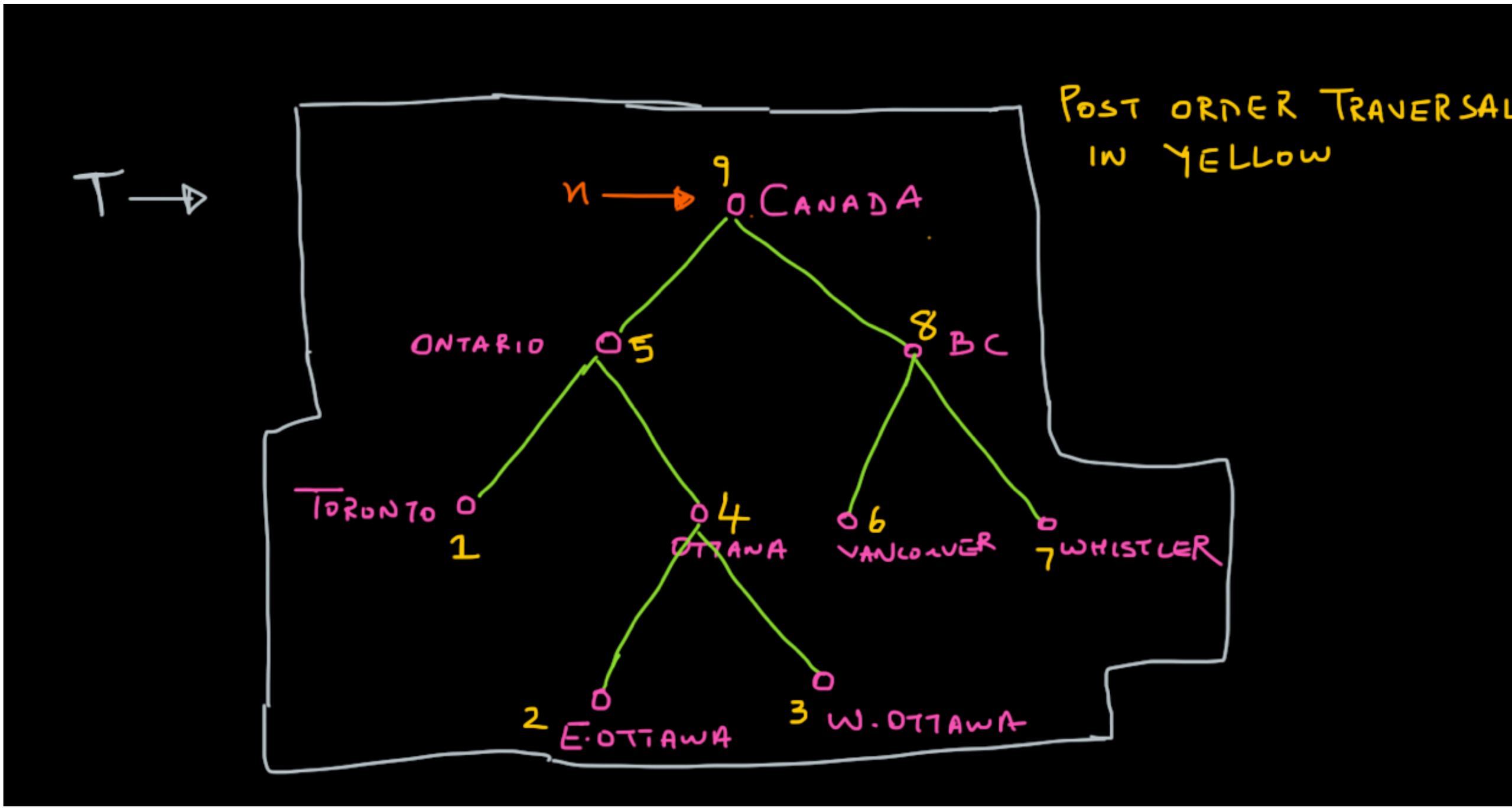
ZOOM IN VERSION



Preorder on binary tree

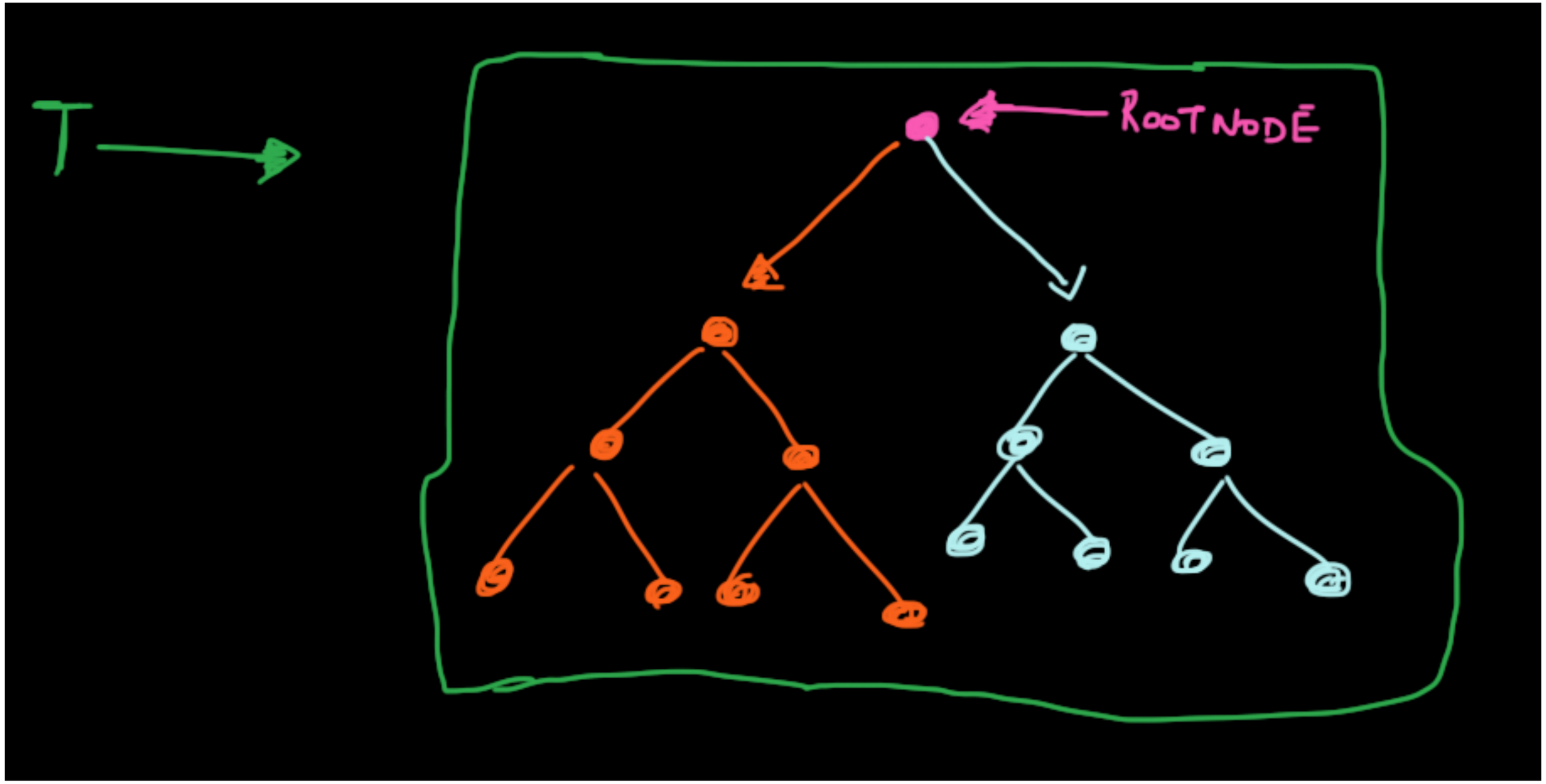


Post-order on binary tree that is represented as list of list.

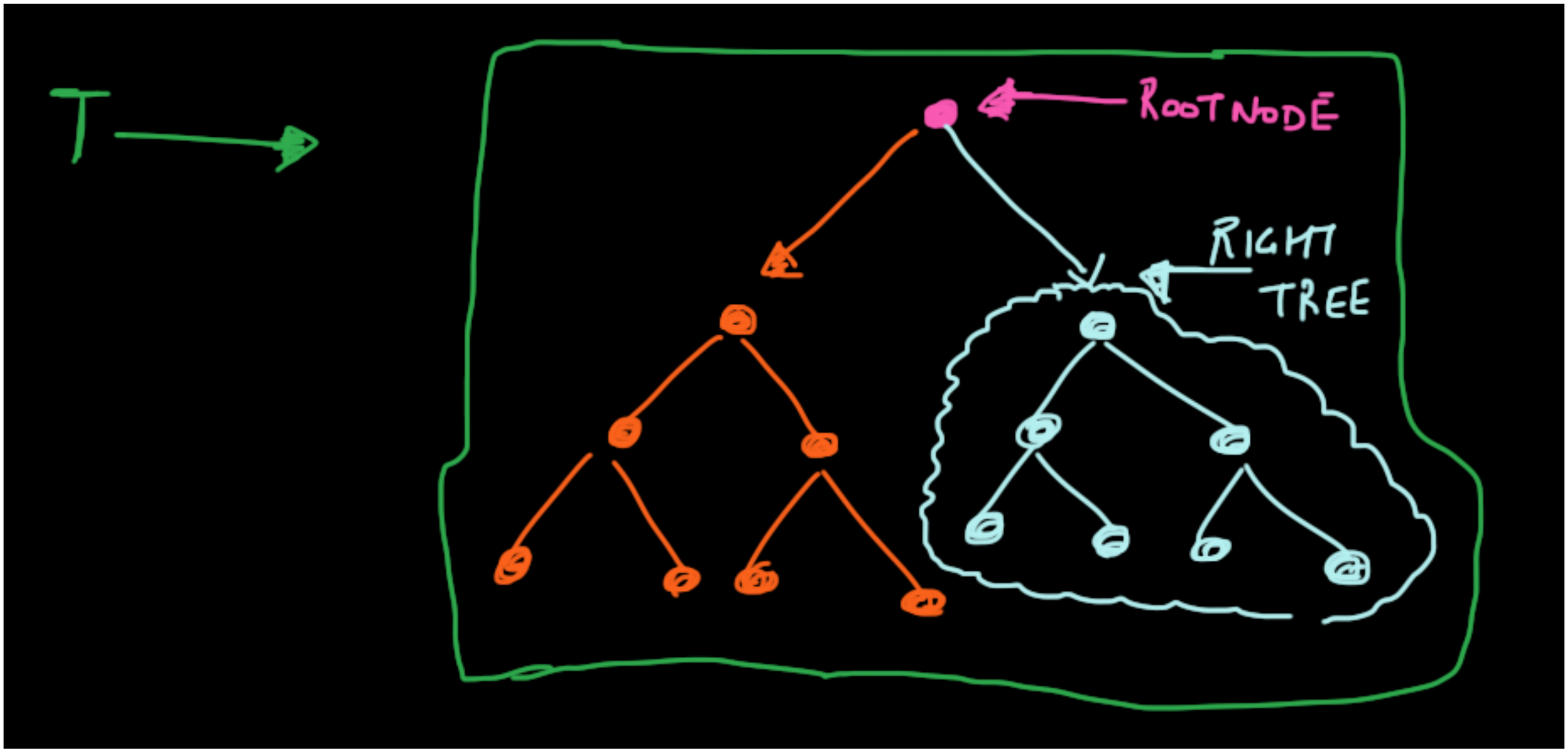


Lets get an intuition first…

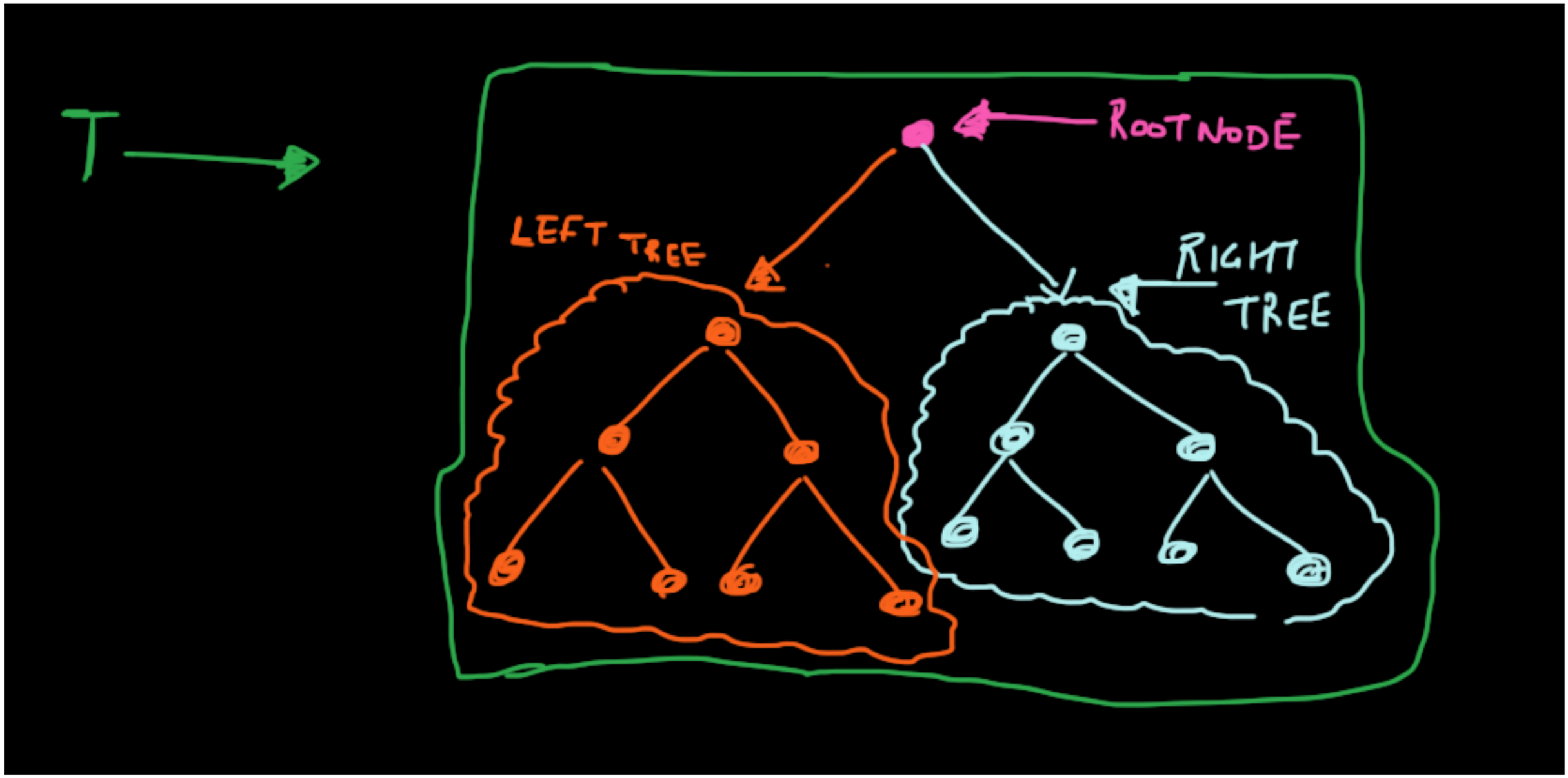
Lets get an intuition first…



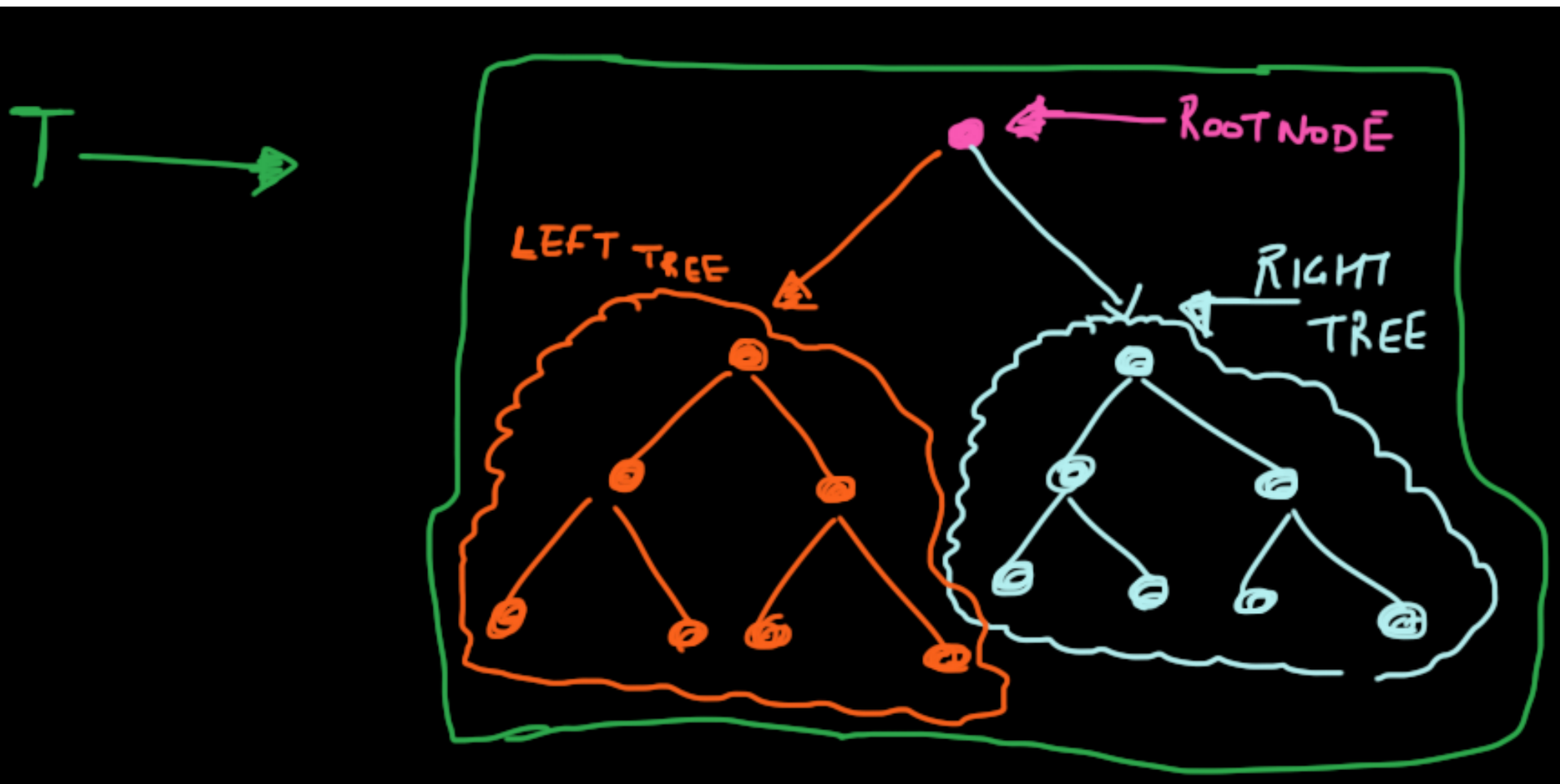
Lets get an intuition first…



Lets get an intuition first…



Lets get an intuition first…



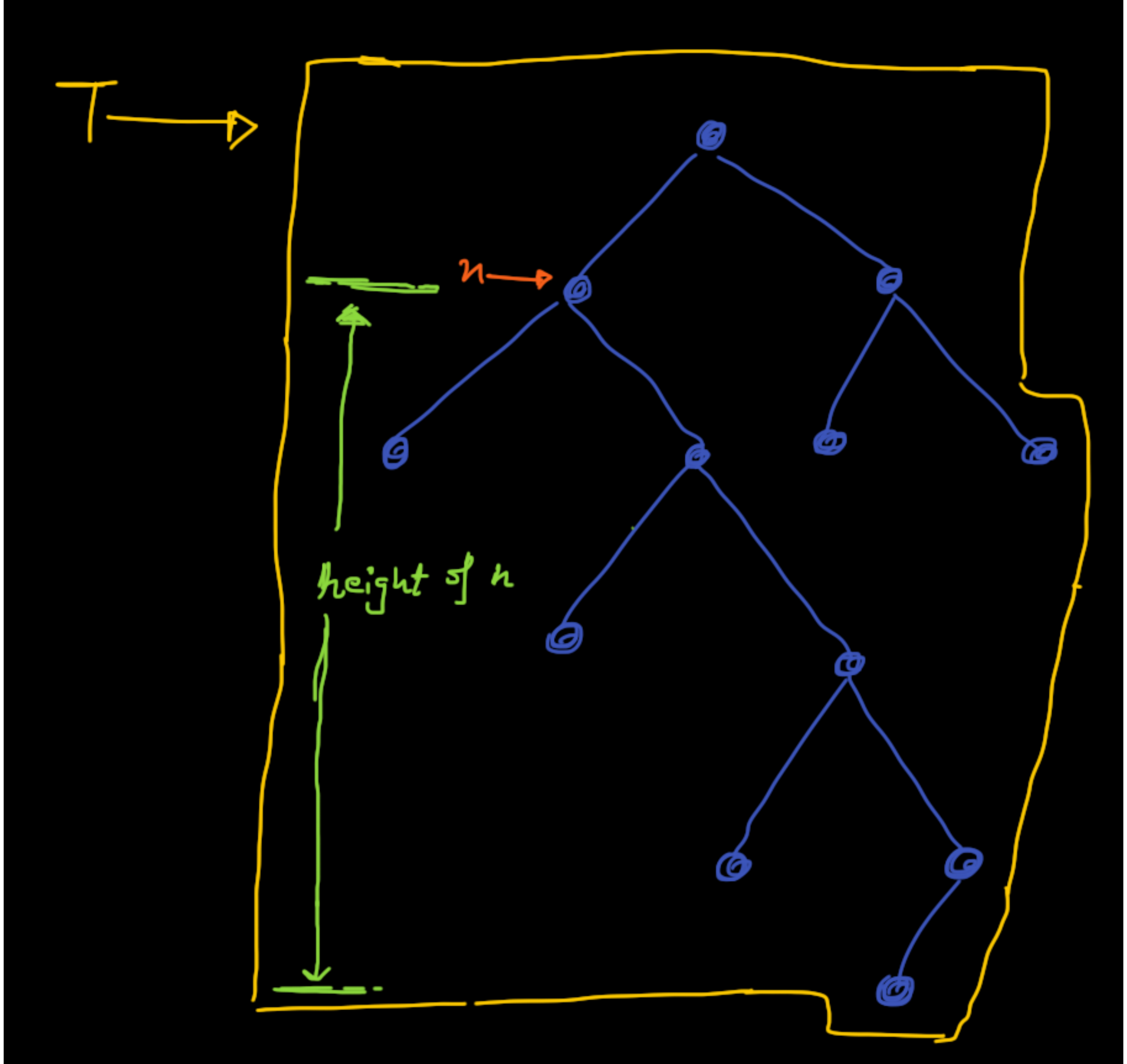
T. RootNode ⟵ refers to the rootnode of T

T. LEFT TREE ⟵ refers to the leftTREE of T

T. RIGHT TREE ⟵ refers to the rightTREE of T

Height of a node ’n'

- The height of a node ’n’ in a tree T is defined as:
  - If ’n’ is a leaf node, then the height of ’n’ is 0.
  - Otherwise the height of ’n’ is one more than the maximum of the heights of ’n’s’ children.

- 
T
n
height of n