Data Structures and Algorithms

Lecture 19: Tree Traversals

Nopadon Juneam
Department of Computer Science
Kasetsart university

Outlines

- Tree traversals: for rooted tree
  - Preorder traversal
  - Postorder traversal

Tree Traversals

การเข้าไปทุกสมาชิกใน tree

- **Traversal** := A systematic way of accessing (visiting, traversing) all the elements of the data structure

- **Graph traversal** := A systematic way of accessing (visiting, traversing) all the nodes and edges of a graph

  - BFS/DFS as we saw in the previous lectures

- **Tree traversal** := A systematic way of accessing all the nodes of a tree.

  - Of course, we can apply BFS/DFS to traverse a tree

  - Preorder & postorder traversals (DFS variants)

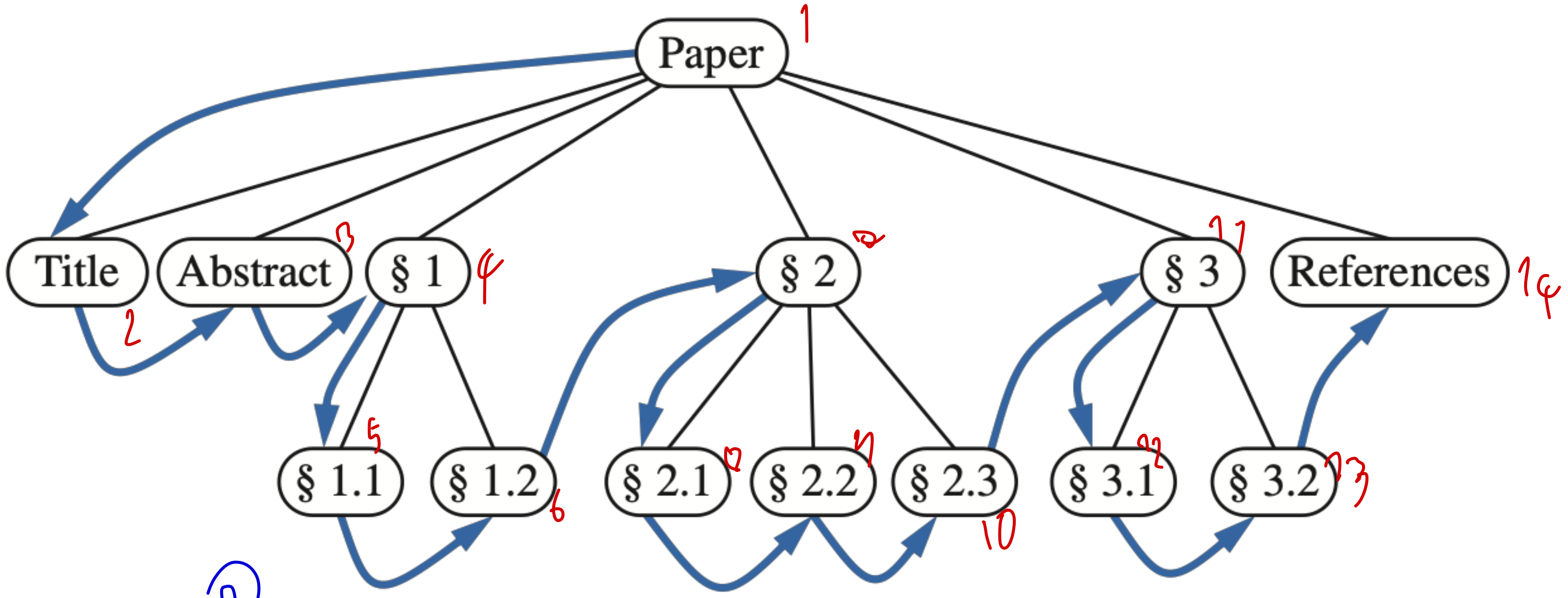
ก่อน

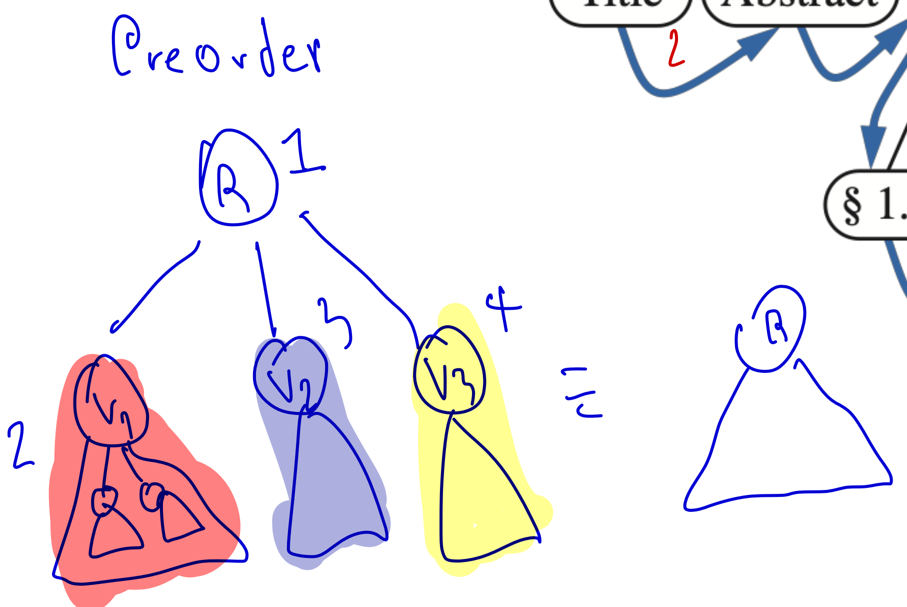
Preorder Traversal

visit Root ก่อนเสมอ

- **Preorder traversal**: In preorder traversal of a rooted tree *T*, we visit the root of *T* first and then the subtrees rooted at its children are traversed recursively

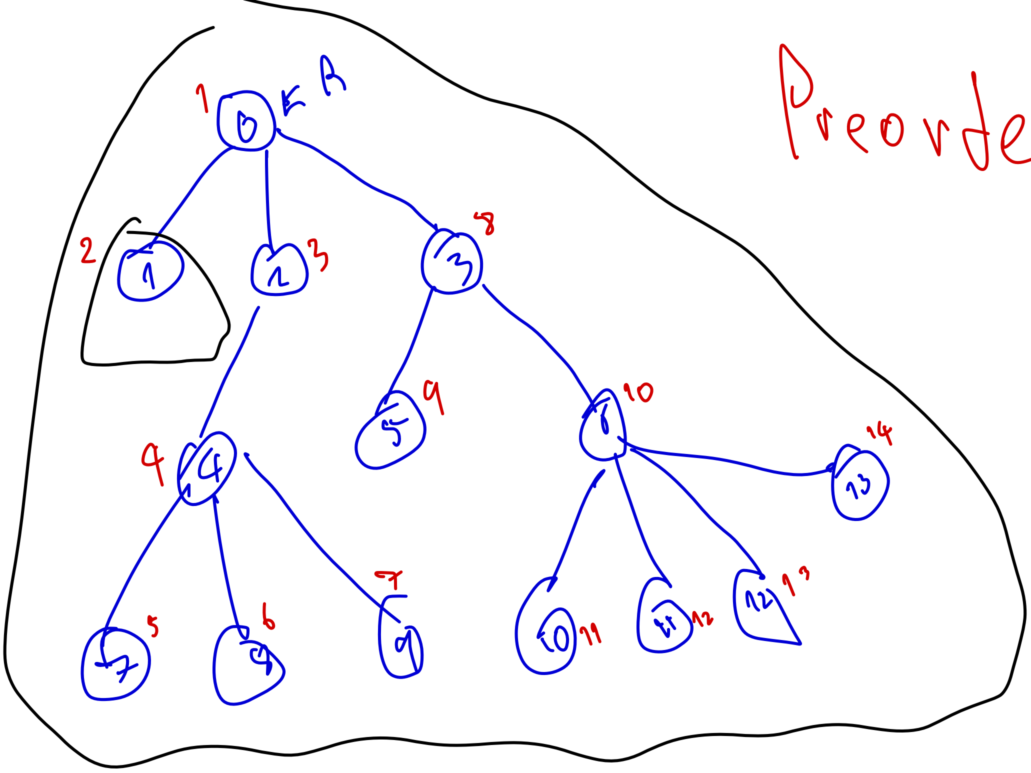
    - If the tree is ordered, then the subtrees are traversed according to the order of the children



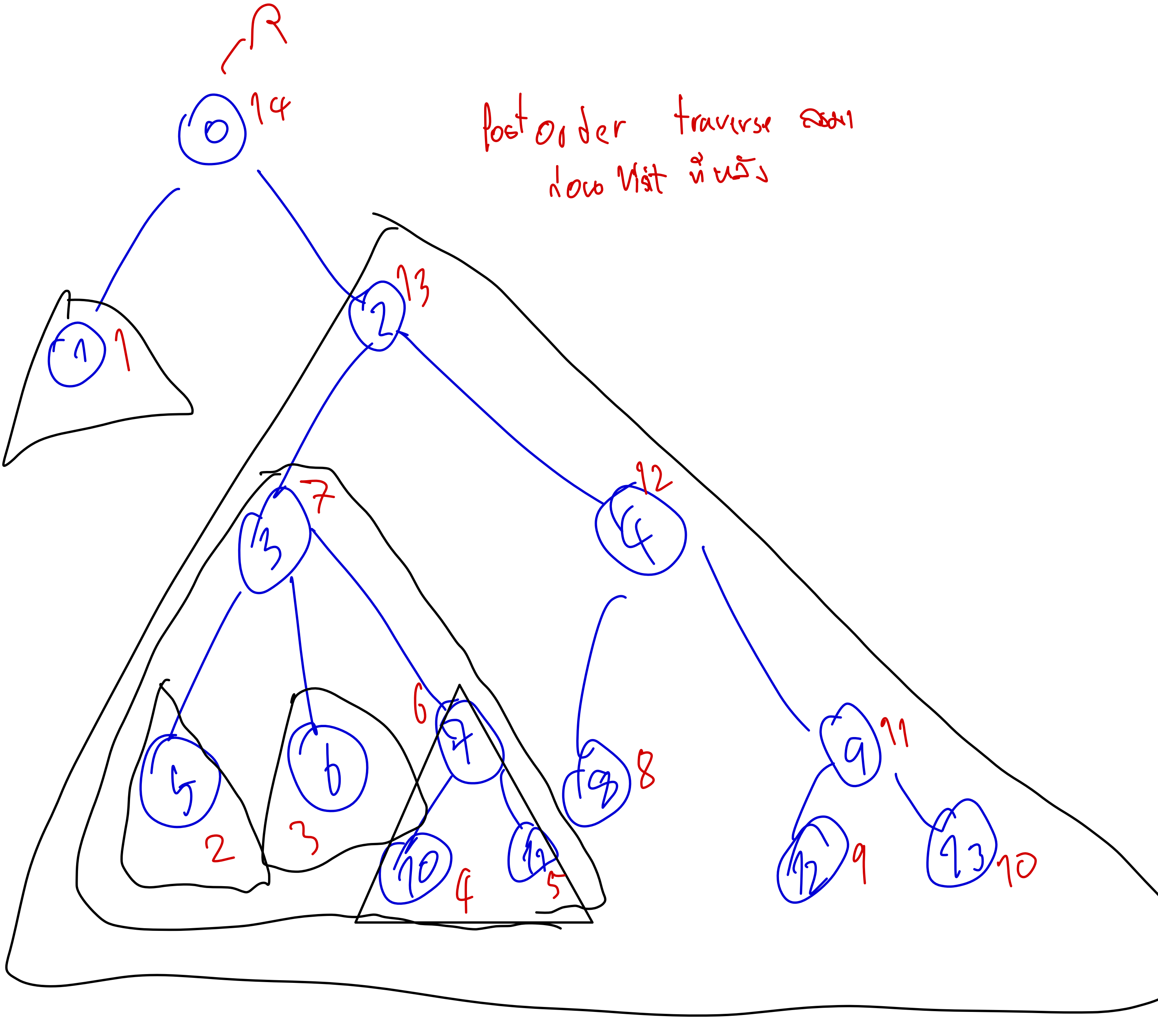


sequence of actions !

1. visit root
2. traverse subtree rooted at v1, a child of r
3. ______ v2



Preorder



postorder traverse ลงไป
ก่อน visit ทีหลัง

Preorder Traversal: Pseudocode (Root, Left, Right)

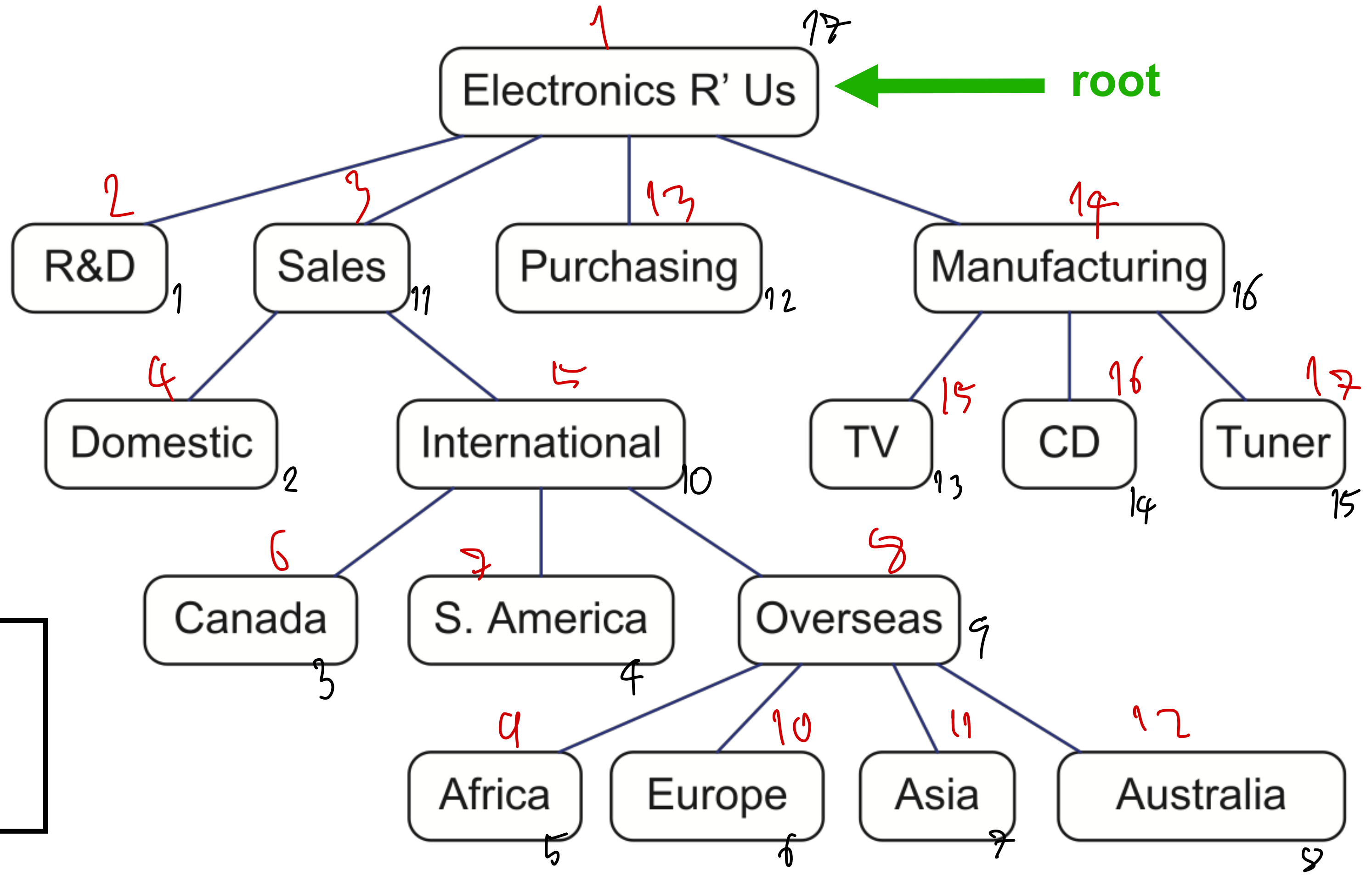
- In preorder traversal of a rooted tree *T*, we visit the root of *T* first and then the subtrees rooted at its children are traversed recursively

```
preorder(r,T):
      visit node r               ลูกของ r
      for each child q of r:
          preorder(q,T)
```

- In other words, we visit the *root* fist, then recursively visit the *left* child, and its *right* siblings

Let's Do Preorder Traversal

● Pre
● Post

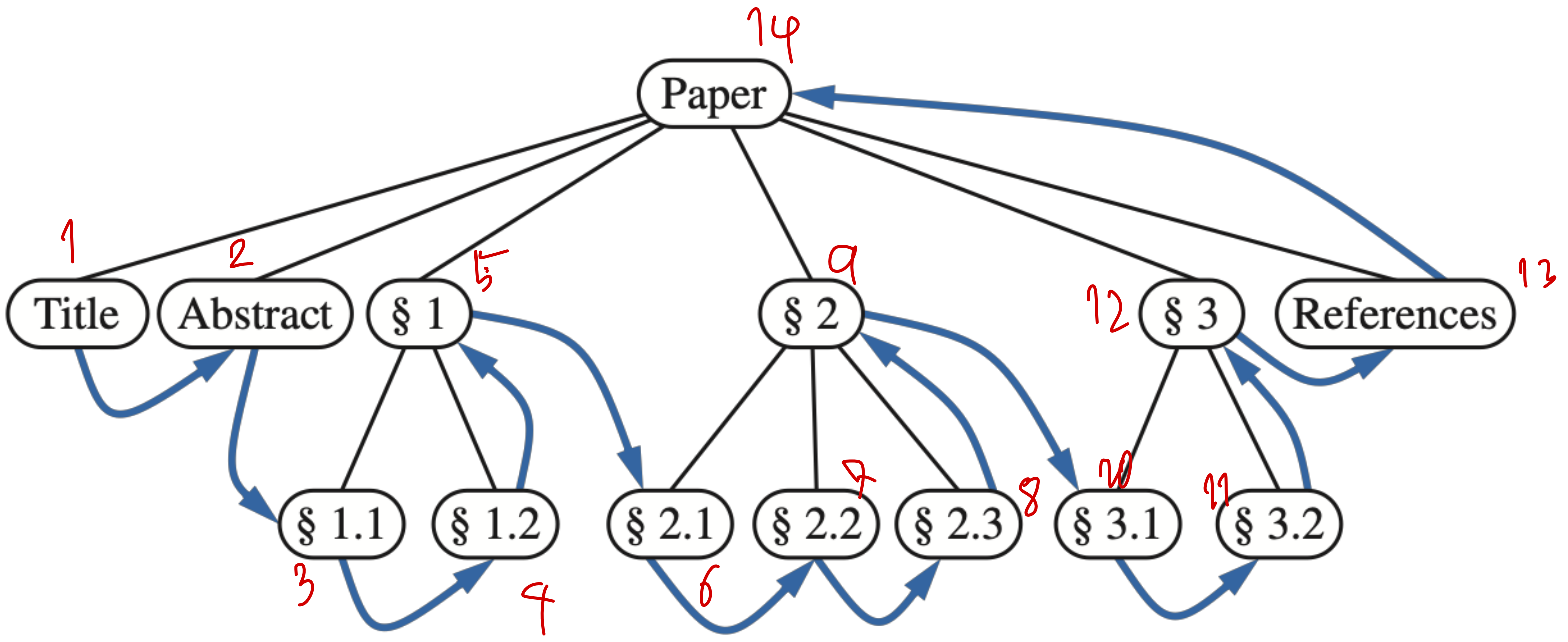


```
preorder(r,T):
    visit node r
    for each child q of r:
      preorder(q,T)
```

- Let's try to perform preorder traversal of the above rooted tree

Postorder Traversal

- **Postorder traversal** : As opposed to preorder traversal, in postorder traversal of a rooted tree *T*, we recursively traverse the subtrees rooted at the children of the root first and then visits the root
  - If the tree is ordered, then the subtrees are traversed according to the order of the children



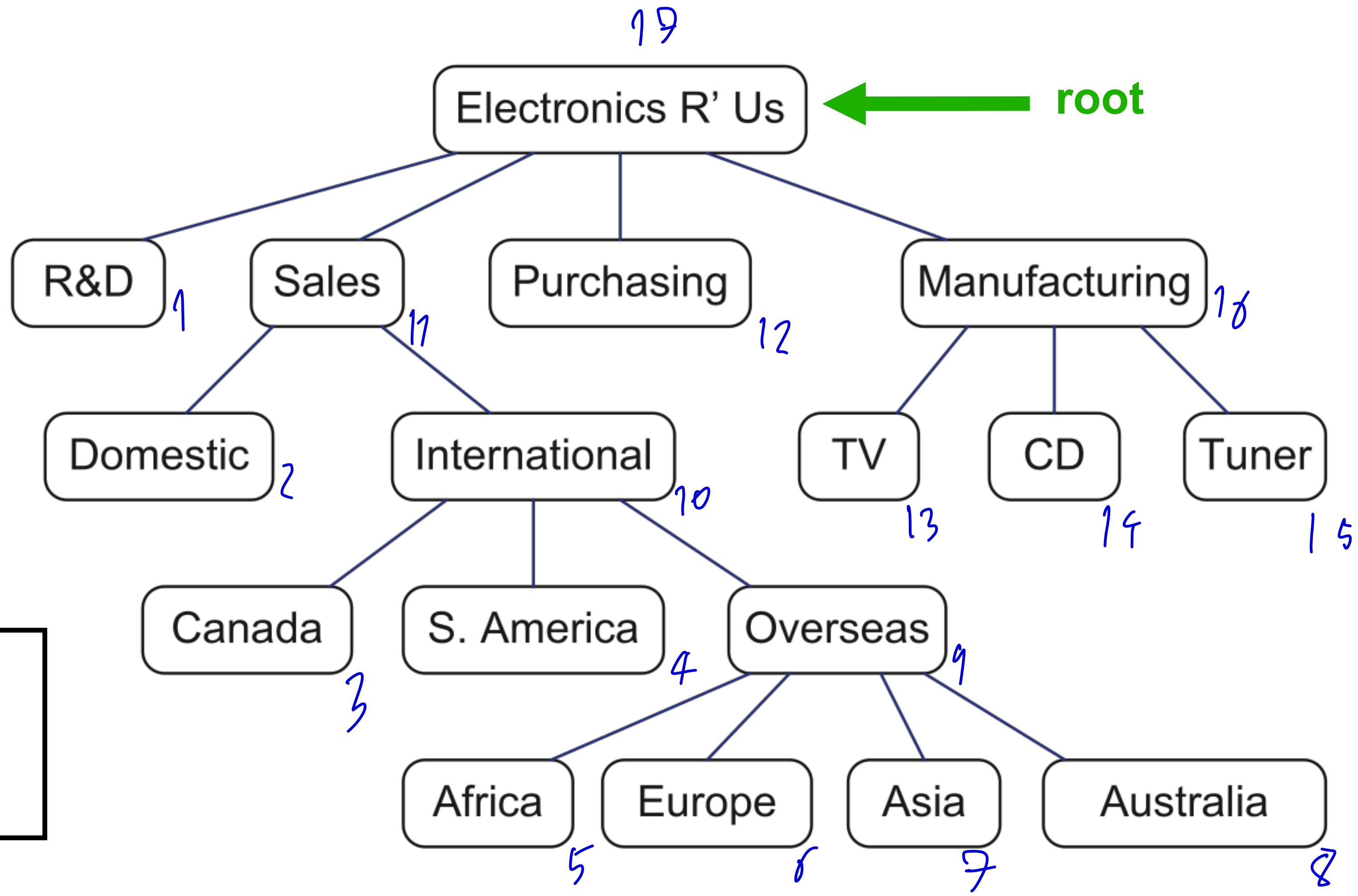
Postorder Traversal: Pseudocode (Left, Right, Root)

- In post traversal of a rooted tree *T*, we recursively traverse the subtrees rooted at the children of the root first and then visits the root

```
postorder(r,T):
    for each child q of r:
        postorder(q,T)
    visit node r
```

- In other words, we visit recursively visit the *left* child, and its *right* siblings, then visit the *root*

Let’s Do Postorder Traversal



ทุกๆ Node ที่เป็น leaf
ต้องมี 1 parent

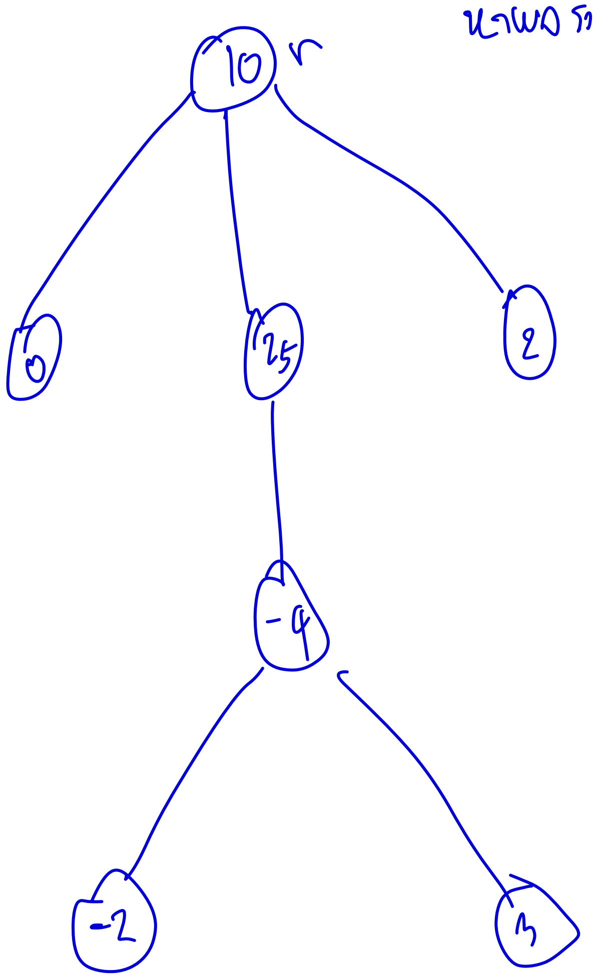
```
postorder(r,T):
    for each child q of r:
      postorder(q,T)
    visit node r
```

- Let’s try to perform a postorder traversal of the rooted tree above

Complexity of Traversals on Ordered Trees

Tree traversal algorithms	Complexity
preorder	$O(n)$
postorder	$O(n)$
	Remarks: a tree of n nodes can only have $n-1$ edges. A traversal need to visit every node of the tree

หาผลรวม



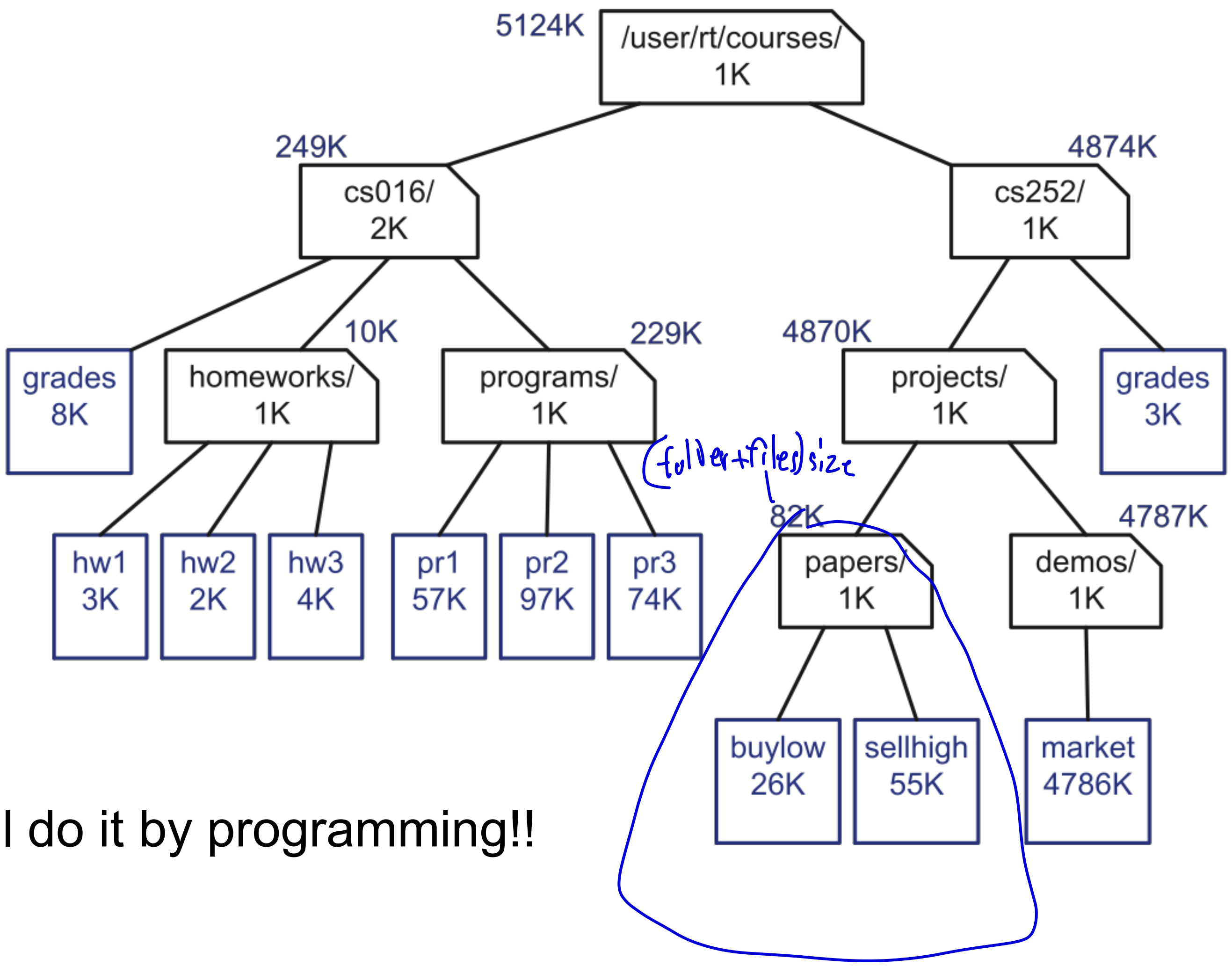
```
s = 0
Preorder (r, T)
    s += r.key
    for each child q of r:
        Preorder(q, T)
```

```
struct node {
    int key
    struct node * firstChild
    struct node * next sibling
}
```

```
Postorder ( Node *root) {
 // visit   root
     for (Node* child = root -> firstchild; child != Null; child = child -> Nextsiblin)
            Postorder (child)
```

Programming Exercise

- Now that you know everything about preorder/postorder traversal. You next assignments will be to construct the tree below which represents the directory structure of the root folder /user/rt/courses/ as well as to compute the file sizes of each directory that resides in the root folder.



5124K /user/rt/courses/ 1K

249K cs016/ 2K

4874K cs252/ 1K

grades 8K

10K homeworks/ 1K

229K programs/ 1K

4870K projects/ 1K

grades 3K

hw1 3K

hw2 2K

hw3 4K

pr1 57K

pr2 97K

pr3 74K

82K papers/ 1K

4787K demos/ 1K

buylow 26K

sellhigh 55K

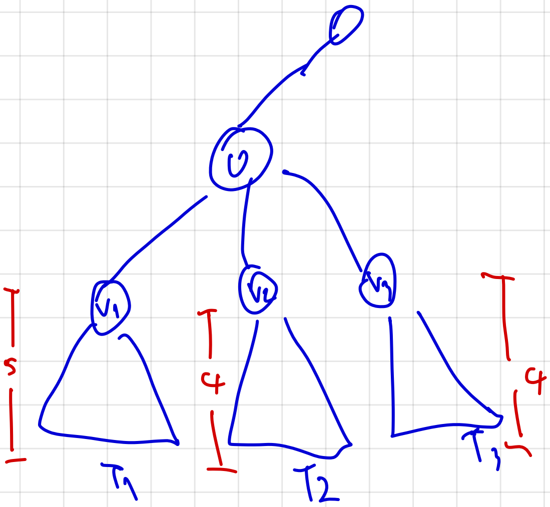
market 4786K

(folder+files)size

- Of course, you will do it by programming!!

- ไม่มีลูก Height = 0

- u มีลูก $\text{height}(u) = \max_{v \in \text{child}(u)} \{ \text{height}(v) \} + 1$



ใช้วิธี
Postorder*