Data Structures

Lecture 16.2: Tree Traversals

Nopadon Juneam
Department of Computer Science
Kasetsart university

Outlines

- Tree traversals:
- Preorder traversal
- Postorder traversal

Tree Traversals

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

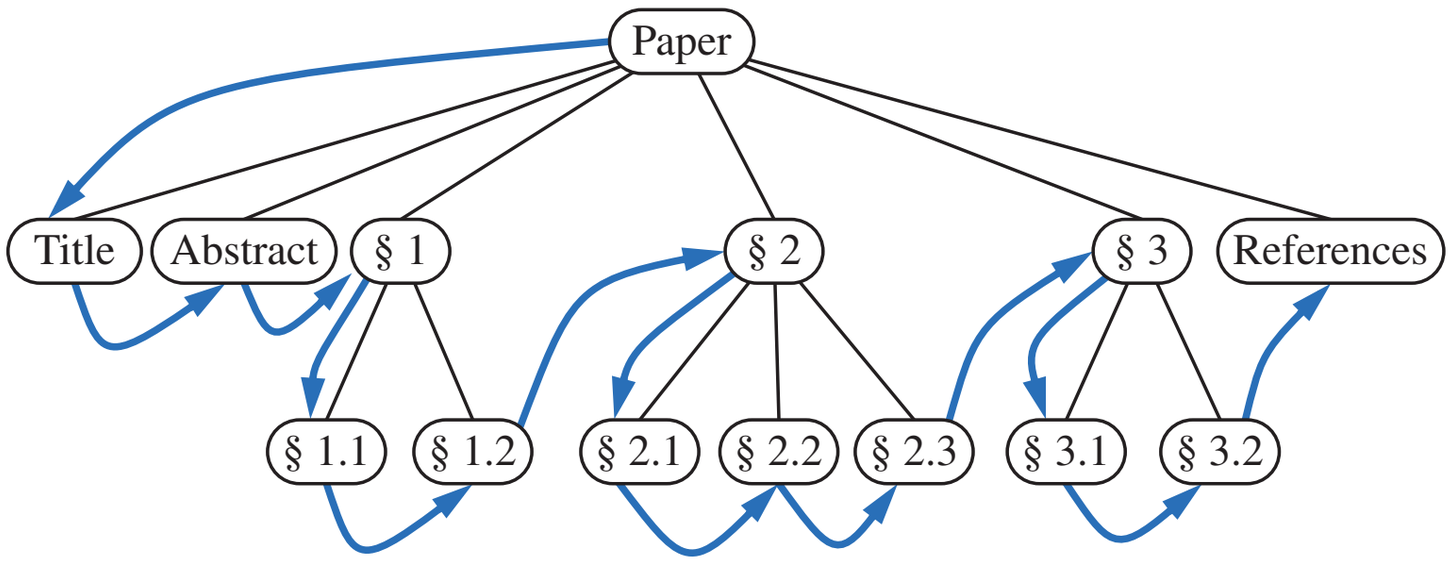
- **Graph traversal** := a systematic way of accessing (visiting, traversing) all the nodes and edges of a graph.
    - BFS/DFS as we saw in the previous lectures.

- **Tree traversal** := a systematic way of accessing all the nodes of a tree.
    - Of course, we can apply BFS/DFS directly.
    - Preorder & postorder traversals (kinds of DFS).

Preorder Traversal

- **Preorder traversal** : In a 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.



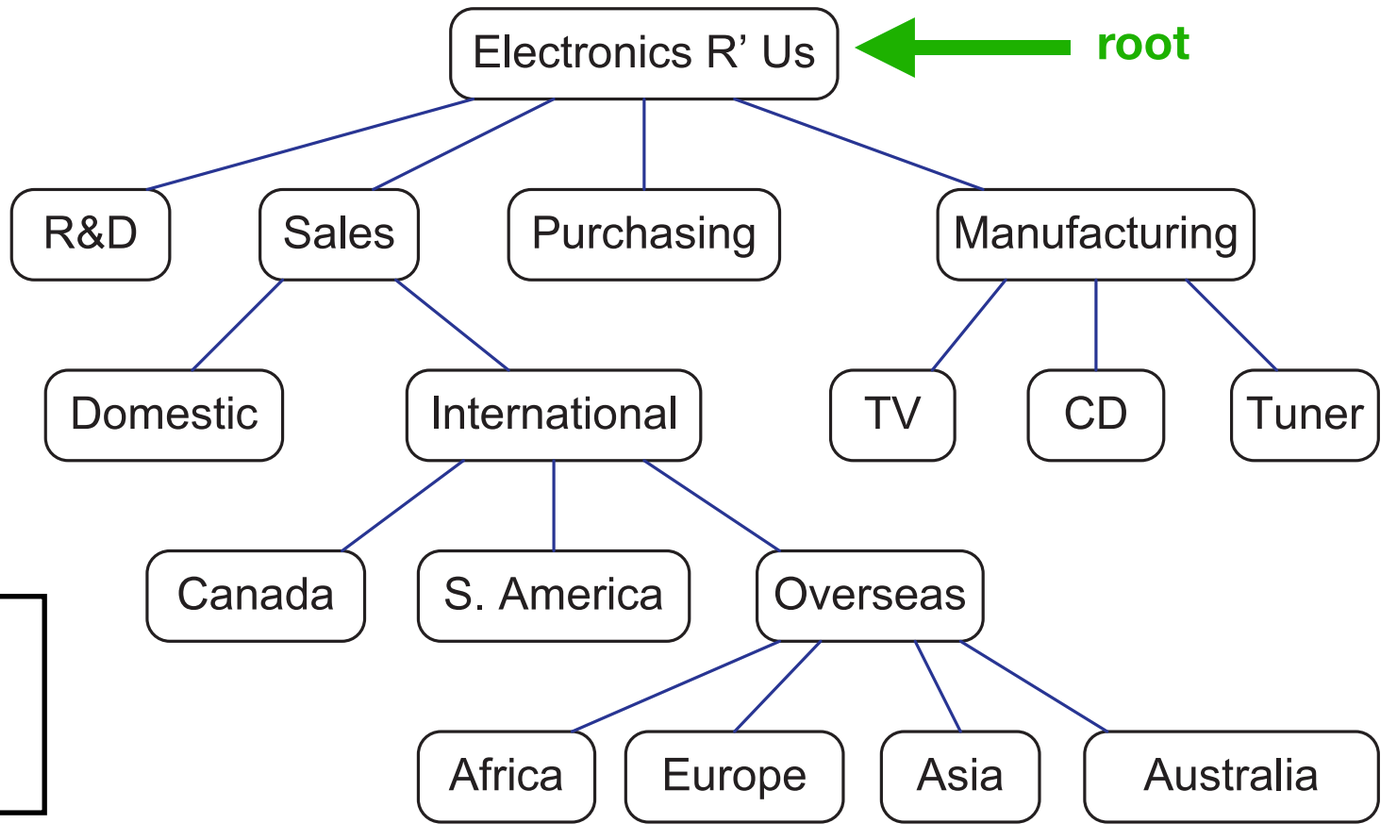
Preorder Traversal: Pseudocode (Root, Left, Right)

- In a 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
    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



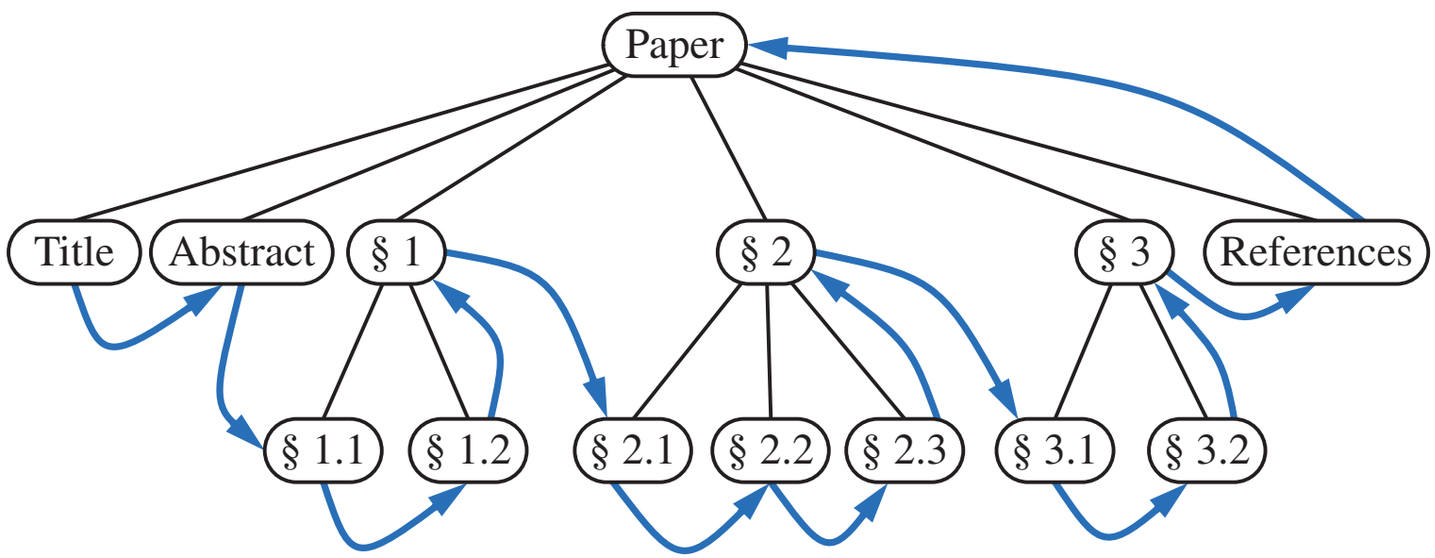

```
preorder(r,T):
      visit node r
      for each child q of r:
         preorder(q,T)
```

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

Postorder Traversal

- **Postorder traversal** : As opposed to preorder traversal, in a 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.



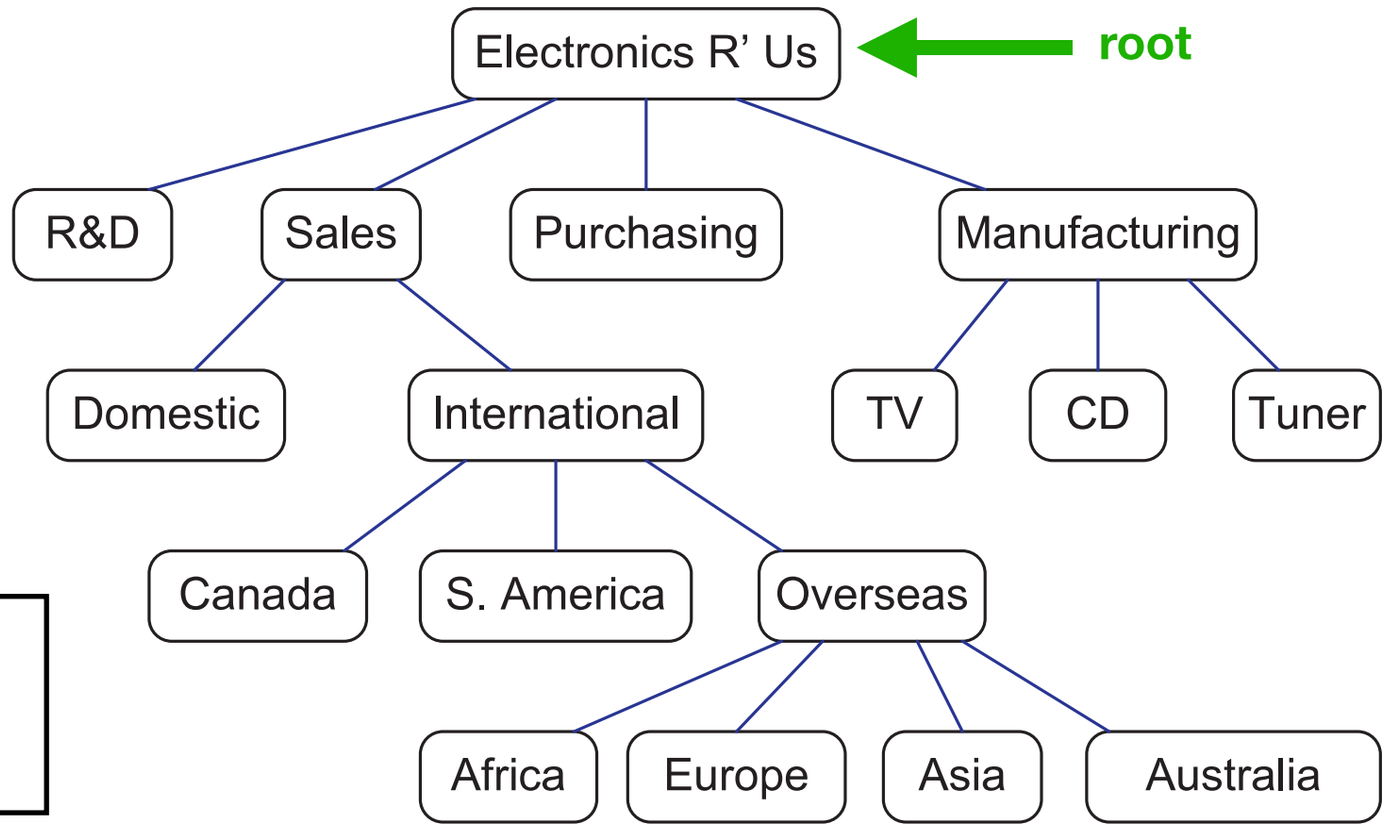
Postorder Traversal: Pseudocode (Left, Right, Root)

- In a 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




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

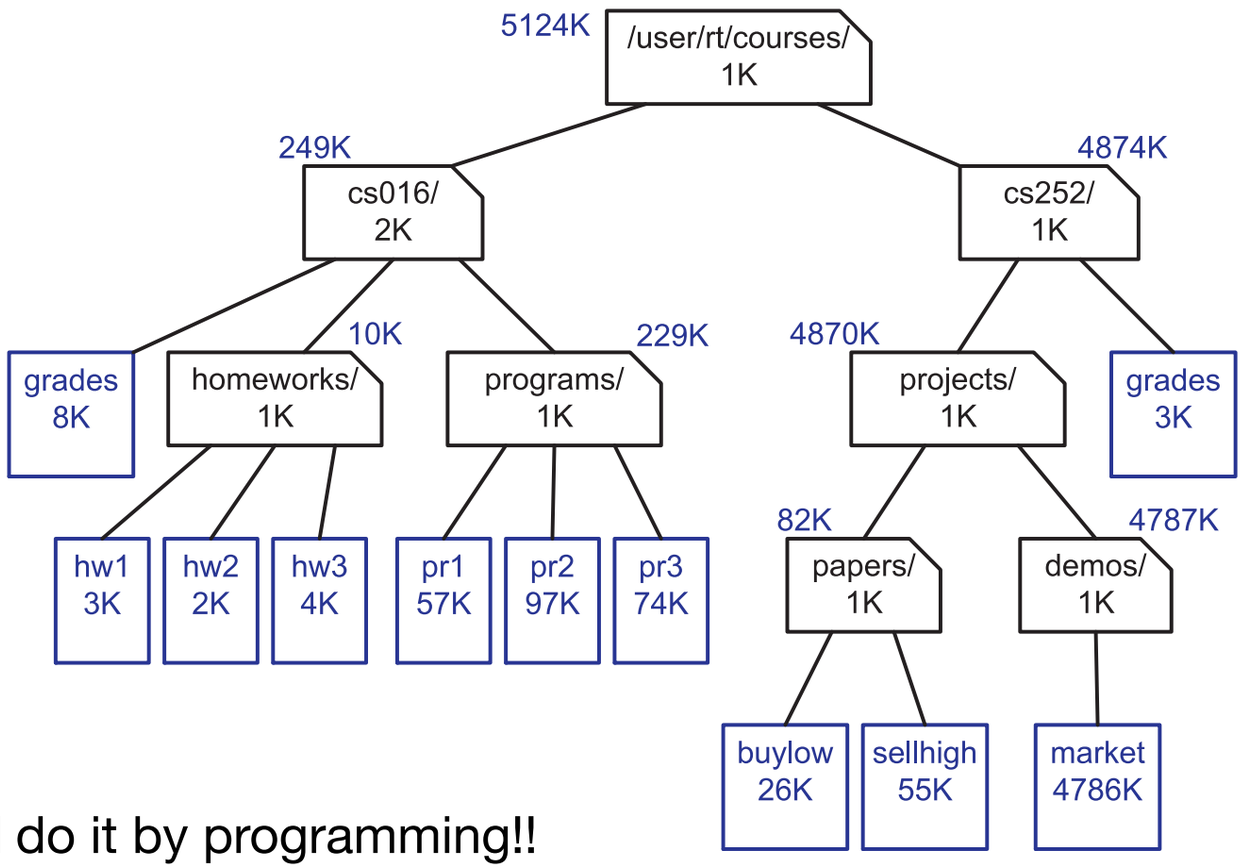
- Let’s try to perform a postorder traversal of the rooted tree above.

Complexity of Operations on Ordered Trees

Tree traversal algorithms	Complexity
`preorder`	$O(n)$
`postorder`	$O(n)$
	Remarks: a tree of n nodes can only has n edges. Each traversal need to visit every node of the tree

Assignment 2

- Now that you know everything about preorder/postorder traversal. You next assignments will be to construct the graph 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.



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