Previously ...

Linked Lists
- singly linked lists
- doubly linked lists
- circular

Stacks

Queues

Queues: Application

Round-robin scheduler using a queue Q
```
event = Q.dequeue()
process event
Q.enqueue(event)
```

Today ...

Trees

List, Stacks, Queues are linear data structures

Trees allow for hierarchical relationships
- nodes have parent-child relation

Trees (jargon)

A tree is either empty or a root node connected to 0 or more trees (called subtrees)

Each node is either a leaf or an internal node
- an internal node has one or more children
- a leaf node (external node) has no children

Nodes with the same parent are siblings

Paths

A path from node $v_i$ to $v_k$ is a sequence of nodes $v_i, v_{i+1}, v_{i+2}, ..., v_k$, where there is an edge from one node to the next

The descendants of a node $v$ are all nodes reached by a path from node $v$ to the leaf nodes

The ancestors of a node $v$ are all nodes found on the path from the root to node $v$
Depth and Height

The length of a path is the number of edges in the path.
The depth (level) of a node \( v \) is the length of the path from \( v \) to root.
The height of a node \( v \) is the length of the path from \( v \) to its deepest descendant.

Properties

Depth of tree is the depth of deepest node.
Height of tree is the height of the root.

Linked Structure for Trees

Every node has:
- data
- parent
- children array

Traversals

Preorder Traversal

```
algorithmpreorder(p) {
    visit(p)
    for each child c of p {
        preorder(c)
    }
}
```

Example: What type of traversal?

Compute space used by files in folders and subfolders:

```
$ du -h -d 2
```