Engineering Informatics 1
Data Structures

Dr. Jonas Pfoh

WS 2013-2014
09.01.2014
What is a Data Structure

A data structure is simply a mechanism for organizing data.
What is a Data Structure

A data structure is simply a mechanism for organizing data.
A singly linked list is a basic data structure that makes use of pointers to store a dynamic number of elements and is traversable in a single direction.
A doubly linked list is a basic data structure that makes use of pointers to store a dynamic number of elements and is traversable in two directions.
A stack is a First In, Last Out (FILO) data structure.
Stack

Operations

- **push**: add an element to the top of the stack
- **pop**: remove an element from the top of the stack
- **peek**: inspect the element on the top of the stack without removing it
A stack is a First In, First Out (FIFO) data structure.
Stack

operations

enqueue add an element to the end of the queue

dequeue remove an element from the front of the queue
Both stacks and queues may be represented with (doubly) linked lists.
A *graph* is a data structure that is comprised of a set of vertices and edges connecting those vertices. Graphs can be used to store data and the relationship between that data.
A *directed graph* is a graph in which the vertices are directed.
A directed acyclic graph is a graph in which the vertices are directed and there are no cycles. A cycle is defined as a path in which the same vertex occurs twice.
A *tree* is a graph in which there is exactly one path between every pair of vertices.
Tree

Terminology I

**root** A vertex without any incoming edges.

**leaf** A vertex without any outgoing edges.

**parent** The vertex with which a vertex shares its incoming edge.

**child/children** The vertex/vertices with which a vertex shares its outgoing edge(s).
**ancestors** The ancestors of a vertex are all the vertices on the path from that vertex to the root.

**descendants** The descendants of a vertex are all the vertices that are traversed when considering all paths from that vertex to a leaf (including the leaves).
A *binary tree* is a tree in which every node has a maximum of two children.
Graphs can be represented in various ways. However for the purpose of this course, we will consider two fashions, an *adjacency matrix* (i.e., a two-dimensional array) and dynamically allocated structures with pointers.
Representation

Adjacency Matrix

\[ \begin{array}{ccc}
A & B & C \\
A & 0 & 1 & 1 \\
B & 1 & 0 & 0 \\
C & 1 & 0 & 0 \\
\end{array} \]
Representation

Pointers

A
B
C

A
B
C

B
C

A
B
C

A
B
C
The height of a tree is the length of the longest path from the root to a leaf. A tree with only one vertex (the root) has a height of 0.

Height = 2
More Tree Terminology

Vertex Depth

The depth of a vertex is the length of the path from the root to the vertex. The root has a depth of 0.

Depth = 1
A subtree is the tree formed by a vertex and all its descendants.
A tree is balanced if the height of all children’s subtrees differ by no more than one, *for each vertex*. 