School of Computer Science

COMP SCI 1103/2103 Algorithm Design & Data Structure

Graphs and Trees

Linked list with complex structure

- A linked list may consist of nodes that include more than one pointer.

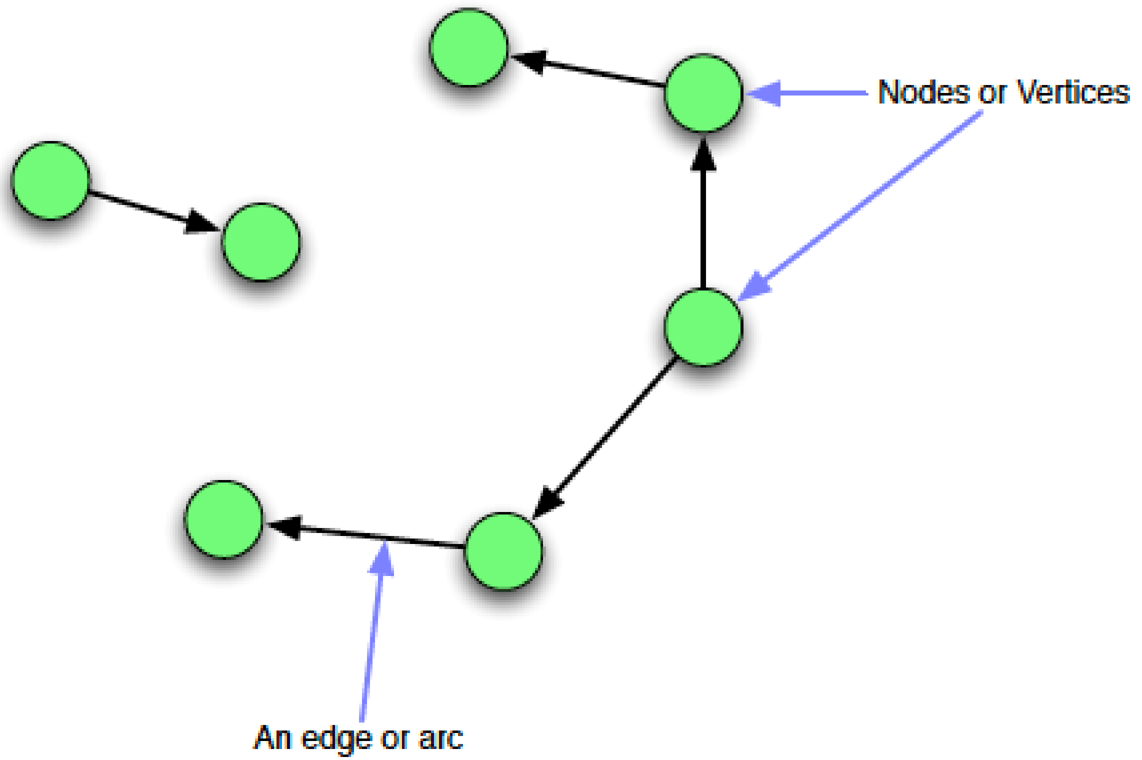
```
struct Node {
int data;
Node *link;
Node *otherLink;
}
```

- This can form a doubly linked list, but can also allow us to create very different structures.

Graph

- A graph is a collection of points (vertices or nodes) where some of the points are connected by line segments (edges or arcs).
- Graphs are a useful mathematical structure and are heavily used in networking, algorithmic studies and advanced computation.
- G= (V,E),
  - V=$\{v_1, v_2, \dots, v_n\}$,
  - E=$\{e_1, e_2, \dots, e_m\}$,
  - $e_i = (v_j, v_k)$ for directed and undirected
  - $e_i = \{v_j, v_k\}$ for undirected graphs
- Maximum number of edges in an undirected graph?

Graph example



Can this structure be represented by a singly linked list?

Can you represent any arbitrary graph by a doubly linked list?

How can we represent graphs in general?

Adjacency lists

Idea: Use for each node v a linked list that stores its outgoing neighbors (alternatively we can also use the incoming neighbors or lists for both).

Advantage:

- Insertion of edges goes in constant time.
- Well suited for sparse graphs (occur often in practice)

Adjacency matrices

Idea: Represent a graph consisting of n nodes by an n×n matrix A. Set

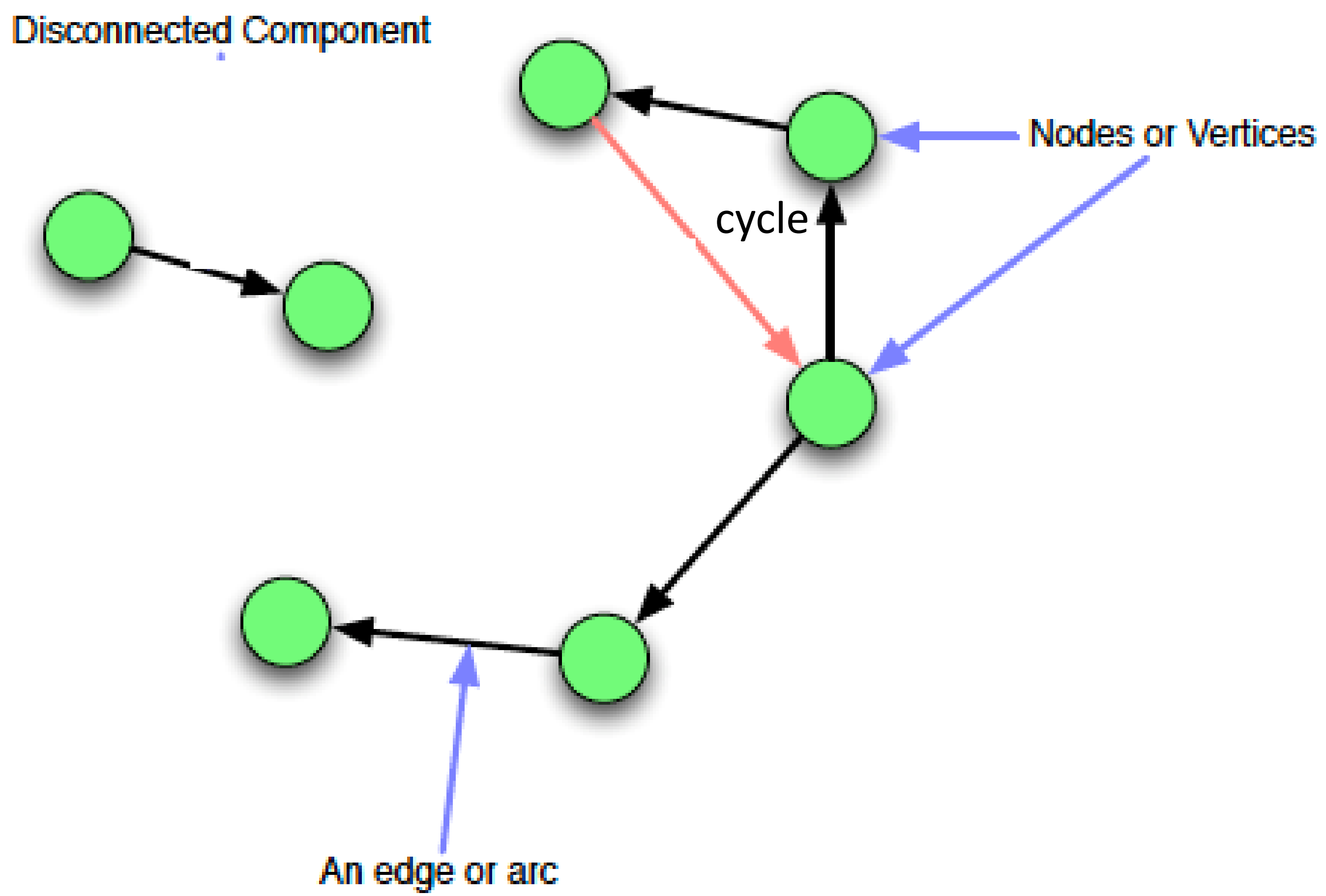
$$A_{ij} = 1 \text{ if } (i,j) \in E$$
$$A_{ij} = 0 \text{ otherwise}$$

- Insertion, removal, edge queries work in constant time.
- O(n) to obtain an edge entering or leaving a node.
- Disadvantage: Storage requirement n^2 even for sparse graphs.

Trees

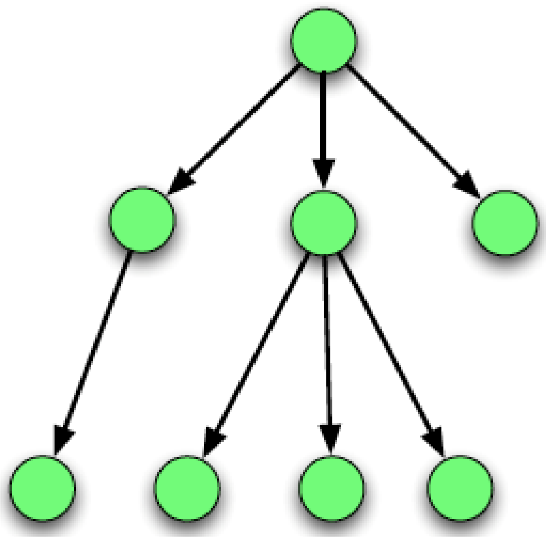
- Graphs with certain properties are called trees.
- Trees are a subset of Graphs.
  - Trees must have all of their nodes connected.
  - Trees cannot contain cycles.
  - In other words, trees are connected, acyclic graphs.
- A tree can be defined in several ways. One natural way to define a tree is using recursion.
- A graph that has no cycles but is not connected is a forest. (Because there's more than one tree…)
- A tree with n nodes, minimum number of edges?

Graph but not tree



Directed rooted tree

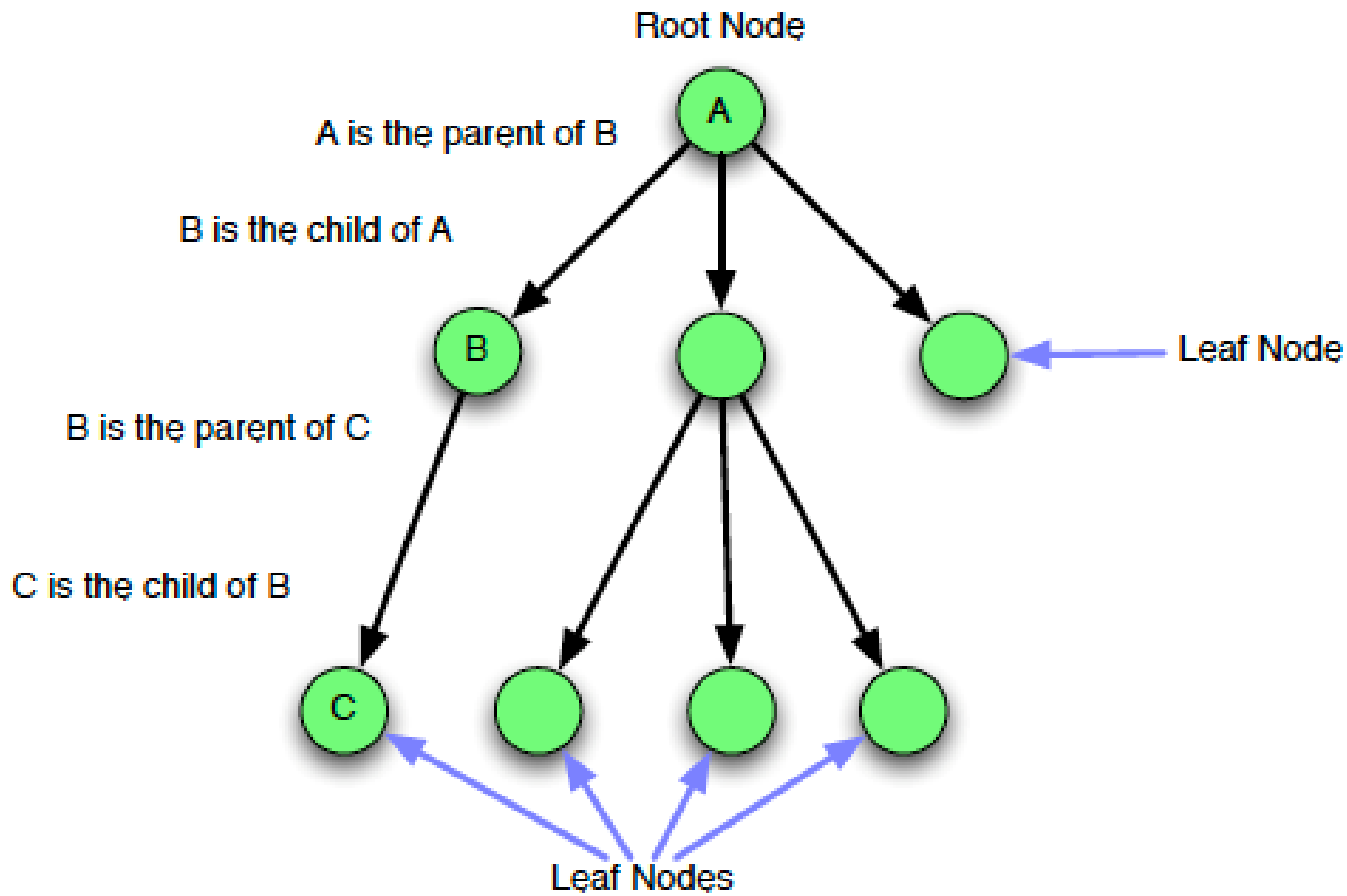
- A tree that consists of
  - a distinguished node r called the root; and
  - zero or more nonempty (sub)trees, each of whose roots are connected by a directed edge from r.



Tree terminology

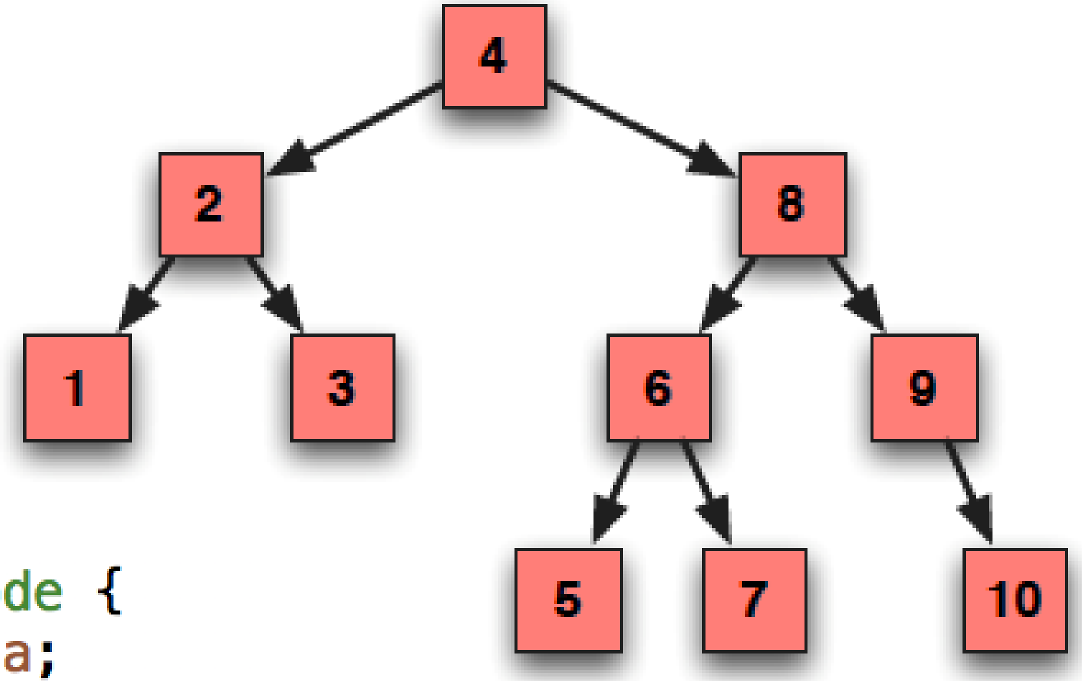
- Trees have a **root**, normally drawn at the top.
- If a node n_1 has a link to another node n_2, then n_1 is the **parent** of n_2 and n_2 is the **child** of n_1.
- If a node has no children, it is called a **leaf**.
- The **depth of a node** n is the length of the path from the root to n. The **depth of a tree** n is the maximum depth of its nodes.
- The **height of a node** the number of edges on the longest path between that node and a leaf. **The height of a tree** is height of its root.
  - Other definition that you can find in references: The number of nodes on the longest path from root to a leaf
  - Height of empty subtree –1 or 0

Tree example



Binary trees

- Trees that have 0, 1 or 2 children.



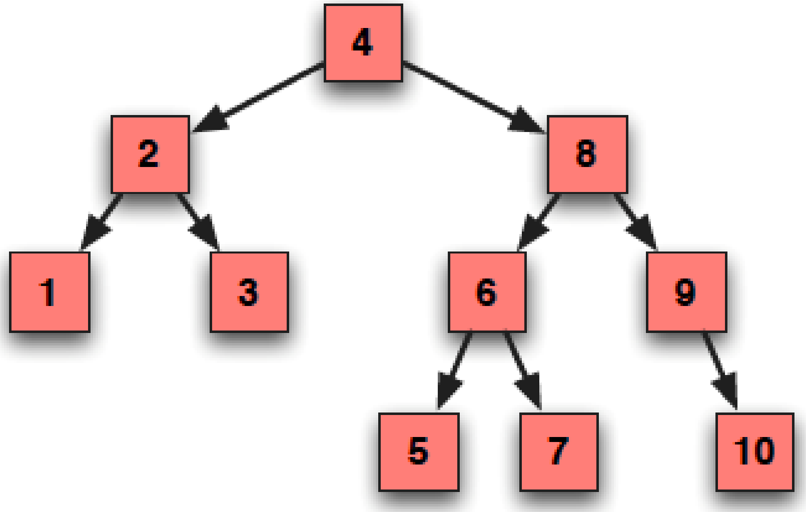

```
struct Node {
  int data;
  Node *link1;
  Node *link2;
}
```

(Un) Balanced trees

- Good to construct the tree such that the height of the tree is minimized
- A tree with all links being NULL remind you of anything?

Ordered, balanced binary tree

- Each node has all elements less than its value in its left-hand child.



- Search for an item ?

Advanced trees

- Some trees are set up to limit their worst case depth in relation to their size. Restricting which items go where in the tree also helps.
- A (fully or partially) ordered and balanced tree can have much better search performance (O(log n)).

THE UNIVERSITY of ADELAIDE