Lecture 3: Paths, Cycles, Trees

Dr. George Mertzios

george.mertzios@durham.ac.uk

Reminder from last lecture

- A graph G is a pair $(V(G), E(G))$, where
  - $V(G)$ is a nonempty set of vertices (or nodes),
  - $E(G)$ is a set of unordered pairs uv with $u, v \in V(G)$ and $u \neq v$, called the edges of G.
- When G is clear from the context, we write simply V and E.

Reminder from last lecture

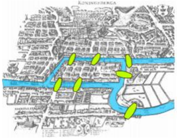
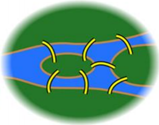
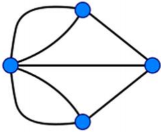
- A graph G is a pair $(V(G), E(G))$, where
  - $V(G)$ is a nonempty set of vertices (or nodes),
  - $E(G)$ is a set of unordered pairs uv with $u, v \in V(G)$ and $u \neq v$, called the edges of G.
- When G is clear from the context, we write simply V and E.
- A path in G is a sequence $v_0, v_1, \ldots, v_n$ of distinct vertices such that $v_i v_{i+1} \in E$ for $i = 0, \ldots, n-1$.
- A cycle in G is a path $v_0, v_1, \ldots, v_n$ such that $v_n v_0 \in E$.
- The length of a path or a cycle is the number of edges in it.
- A graph is connected if any two vertices in it are connected by a path.

Contents for today's lecture

- Eulerian and Hamiltonian cycles;
- The traveling salesman problem;
- Trees and their properties;
- Applications of trees;
- Examples and exercises.

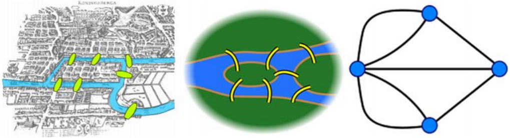
Special circuits/cycles in graphs

- Can we travel along the edges of a given graph G so that we start and finish at the same vertex and traverse each edge exactly once?
  - Such a circuit in G is called a Eulerian circuit, after Leonhard Euler (1707-83).

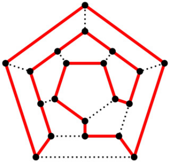




Special circuits/cycles in graphs

- Can we travel along the edges of a given graph G so that we start and finish at the same vertex and traverse each edge exactly once?
  - Such a circuit in G is called a Eulerian circuit, after Leonhard Euler (1707-83).



- Can we travel along the edges of a given graph so that we start and finish at the same vertex and visit each vertex exactly once?
  - Such a cycle is called a Hamiltonian cycle, after William Hamilton (1805-65).



Eulerian circuits

Eulerian circuits

Theorem

A connected graph with at least two vertices has an Eulerian curcuit iff each of its vertices has even degree.

Eulerian circuits

Theorem

A connected graph with at least two vertices has an Eulerian curcuit iff each of its vertices has even degree.

Idea of the proof:

Necessity ($\Rightarrow$): each time this circuit passes through a vertex v, it contributes 2 to $deg(v)$. Since each edge is used exactly once, $deg(v)$ must be even.

Eulerian circuits

Theorem

A connected graph with at least two vertices has an Eulerian curcuit iff each of its vertices has even degree.

Idea of the proof:

Necessity ($\Rightarrow$): each time this circuit passes through a vertex v, it contributes 2 to $deg(v)$. Since each edge is used exactly once, $deg(v)$ must be even.

Sufficiency ($\Leftarrow$): Induction on the number of vertices in G.
Induction base: $G = K_3$, the claim is obvious.

Eulerian circuits

Theorem

A connected graph with at least two vertices has an Eulerian curcuit iff each of its vertices has even degree.

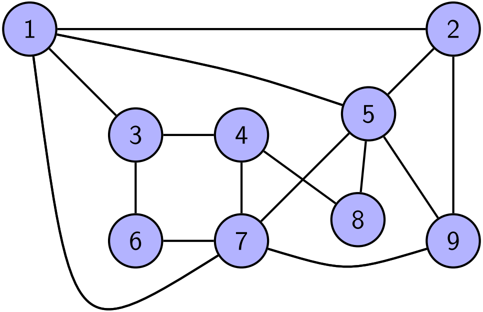
Idea of the proof:

Necessity ($\Rightarrow$): each time this circuit passes through a vertex v, it contributes 2 to $deg(v)$. Since each edge is used exactly once, $deg(v)$ must be even.

Sufficiency ($\Leftarrow$): Induction on the number of vertices in G.
Induction base: $G = K_3$, the claim is obvious. Induction step:

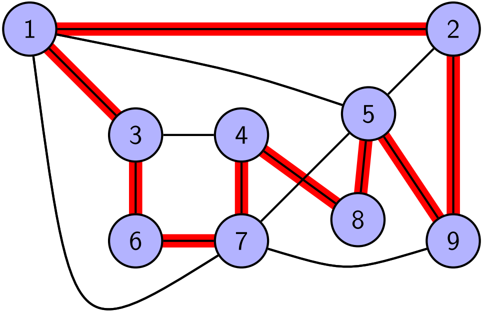
- Start walking from any vertex u along the untraversed edges, and continue by "marking" every edge when you traverse it.
- Stop when you arrive at a vertex where you can't continue (all edges of it are are already traversed). This vertex must be u again (only even-degrees!).
- Hence we have a circuit C. Delete all edges in C from G to obtain a smaller graph H in which all degrees are also even.
- By induction hypothesis, each conn. component of H has an Eulerian circuit.
- Combine C and these circuits to obtain the required circuit for G. (why?)

Hamiltonian cycles



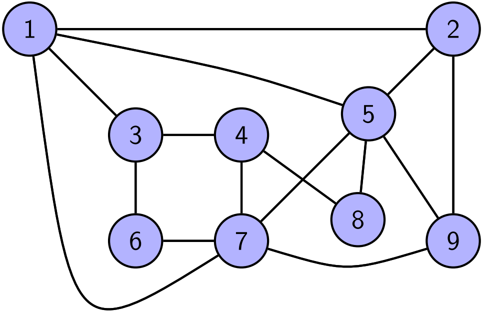
- Does this graph have a Eulerian circuit?
- Does it have a Hamiltonian cycle?

Hamiltonian cycles



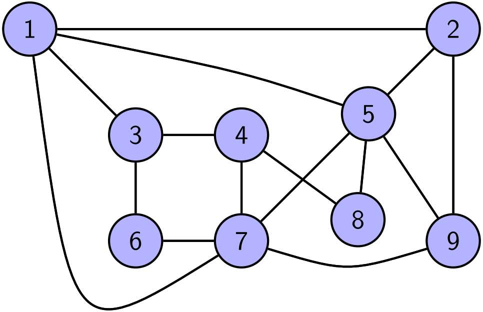
- Does this graph have a Eulerian circuit?
- Does it have a Hamiltonian cycle?

Hamiltonian cycles



- Does this graph have a Eulerian circuit?
- Does it have a Hamiltonian cycle?
- Detecting Eulerian circuits algorithmically is easy. (How?)

Hamiltonian cycles



- Does this graph have a Eulerian circuit?
- Does it have a Hamiltonian cycle?
- Detecting Eulerian circuits algorithmically is easy. (How?)
- Detecting Hamiltonian cycles is hard (NP-complete).

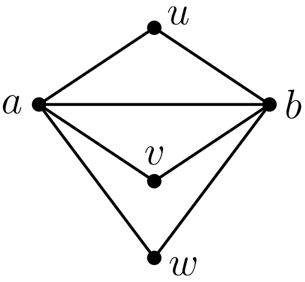
Exercise

Exercise 1: Which of the graphs Q_3, Q_4, K_3, K_4 are Eulerian? Hamiltonian? (that is, contains the corresponding circuit/cycle).

Exercise

Exercise 1: Which of the graphs Q_3, Q_4, K_3, K_4 are Eulerian? Hamiltonian? (that is, contains the corresponding circuit/cycle).

What about this graph?



Travelling Salesman Problem (TSP)

Travelling Salesman Problem (TSP)

The (famous) TSP is the following problem:

- A salesman should visit cities $c_1, c_2, \ldots, c_n$ in some order, visiting each city exactly once and returning to the starting point
- A (positive integer) cost $d(i, j)$ of travel between each pair (c_i, c_j) is known.
- Goal: find an optimal (i.e. cheapest) route for the salesman.

Travelling Salesman Problem (TSP)

The (famous) TSP is the following problem:

- A salesman should visit cities $c_1, c_2, \ldots, c_n$ in some order, visiting each city exactly once and returning to the starting point
- A (positive integer) cost $d(i, j)$ of travel between each pair (c_i, c_j) is known.
- Goal: find an optimal (i.e. cheapest) route for the salesman.

Given a graph G with set V of vertices ($|V| = n$) and set E of edges,

- for each vertex v, create a city c_v;
- for each pair of distinct $u, v \in V$, set $d(c_u, c_v) = 1$ if $uv \in E$ and $d(c_u, c_v) = 2$ otherwise.

Travelling Salesman Problem (TSP)

The (famous) TSP is the following problem:

- A salesman should visit cities $c_1, c_2, \ldots, c_n$ in some order, visiting each city exactly once and returning to the starting point
- A (positive integer) cost $d(i, j)$ of travel between each pair (c_i, c_j) is known.
- Goal: find an optimal (i.e. cheapest) route for the salesman.

Given a graph G with set V of vertices ($|V| = n$) and set E of edges,

- for each vertex v, create a city c_v;
- for each pair of distinct $u, v \in V$, set $d(c_u, c_v) = 1$ if $uv \in E$ and $d(c_u, c_v) = 2$ otherwise.

Then detecting a Hamiltonian cycle in G can be viewed as TSP:

- if G has a Hamiltonian cycle then the cycle is a route of cost exactly n.
- if there is a route of cost n then it can't use pairs with cost 2 (why?) and so goes through edges of G and hence is a Hamiltonian cycle.

Trees

We now turn to a special graph class that has many applications in many areas.

Definitions

A forest is an acyclic graph, i.e. graph without cycles.

Trees

We now turn to a special graph class that has many applications in many areas.

Definitions

A forest is an acyclic graph, i.e. graph without cycles.

A tree is a connected forest, i.e. a connected acyclic graph.

Trees

We now turn to a special graph class that has many applications in many areas.

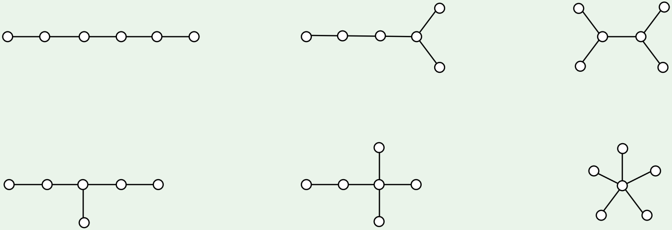
Definitions

A forest is an acyclic graph, i.e. graph without cycles.

A tree is a connected forest, i.e. a connected acyclic graph.

Examples

The different trees on 6 vertices are shown below.



We can also consider this as a forest on 36 vertices.

Spanning trees

A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is spanning if $V' = V$.

Spanning trees

A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is spanning if $V' = V$.

Theorem

Every connected graph contains a spanning tree (a spanning subgraph that is a tree).

Spanning trees

A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is spanning if $V' = V$.

Theorem

Every connected graph contains a spanning tree (a spanning subgraph that is a tree).

An algorithmic proof.

Let G be a connected graph.

- If G contains no cycles, it is a tree, and hence a spanning tree of itself.

Spanning trees

A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is spanning if $V' = V$.

Theorem

Every connected graph contains a spanning tree (a spanning subgraph that is a tree).

An algorithmic proof.

Let G be a connected graph.

- If G contains no cycles, it is a tree, and hence a spanning tree of itself.
- If G contains a cycle, we can remove one edge from the cycle.
- The new graph is still connected. (Why?)

Spanning trees

A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is spanning if $V' = V$.

Theorem

Every connected graph contains a spanning tree (a spanning subgraph that is a tree).

An algorithmic proof.

Let G be a connected graph.

- If G contains no cycles, it is a tree, and hence a spanning tree of itself.
- If G contains a cycle, we can remove one edge from the cycle.
- The new graph is still connected. (Why?)
- Repeating this, we can destroy all cycles and end up with a spanning tree. □

Spanning trees

A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is spanning if $V' = V$.

Theorem

Every connected graph contains a spanning tree (a spanning subgraph that is a tree).

An algorithmic proof.

Let G be a connected graph.

- If G contains no cycles, it is a tree, and hence a spanning tree of itself.
- If G contains a cycle, we can remove one edge from the cycle.
- The new graph is still connected. (Why?)
- Repeating this, we can destroy all cycles and end up with a spanning tree. $\square$

How many repetitions do we need for the above algorithm?

Spanning trees

A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is spanning if $V' = V$.

Theorem

Every connected graph contains a spanning tree (a spanning subgraph that is a tree).

An algorithmic proof.

Let G be a connected graph.

- If G contains no cycles, it is a tree, and hence a spanning tree of itself.
- If G contains a cycle, we can remove one edge from the cycle.
- The new graph is still connected. (Why?)
- Repeating this, we can destroy all cycles and end up with a spanning tree. □

How many repetitions do we need for the above algorithm?

It follows that trees are the smallest connected structures.

Spanning trees

A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is spanning if $V' = V$.

Theorem

Every connected graph contains a spanning tree (a spanning subgraph that is a tree).

An algorithmic proof.

Let G be a connected graph.

- If G contains no cycles, it is a tree, and hence a spanning tree of itself.
- If G contains a cycle, we can remove one edge from the cycle.
- The new graph is still connected. (Why?)
- Repeating this, we can destroy all cycles and end up with a spanning tree. □

How many repetitions do we need for the above algorithm?

It follows that trees are the smallest connected structures.

Finding minimum-weight spanning trees in edge-weighted graphs is an important task in practice: we will learn fast algorithms for it in a few lectures.

Leaves in trees

A leaf in a tree is a vertex of degree 1.

Lemma

Every tree on at least two vertices contains a leaf.

Leaves in trees

A leaf in a tree is a vertex of degree 1.

Lemma

Every tree on at least two vertices contains a leaf.

Proof.

By contradiction:

- Assuming that every vertex has degree 0 or at least 2, we will show that the graph is not a tree.

Leaves in trees

A leaf in a tree is a vertex of degree 1.

Lemma

Every tree on at least two vertices contains a leaf.

Proof.

By contradiction:

- Assuming that every vertex has degree 0 or at least 2, we will show that the graph is not a tree.
- If a vertex has degree 0, then:

Leaves in trees

A leaf in a tree is a vertex of degree 1.

Lemma

Every tree on at least two vertices contains a leaf.

Proof.

By contradiction:

- Assuming that every vertex has degree 0 or at least 2, we will show that the graph is not a tree.
- If a vertex has degree 0, then: the graph (which contains at least two vertices) is not connected, hence not a tree.
- If every vertex has degree at least 2:

Leaves in trees

A leaf in a tree is a vertex of degree 1.

Lemma

Every tree on at least two vertices contains a leaf.

Proof.

By contradiction:

- Assuming that every vertex has degree 0 or at least 2, we will show that the graph is not a tree.
- If a vertex has degree 0, then: the graph (which contains at least two vertices) is not connected, hence not a tree.
- If every vertex has degree at least 2: just start at a vertex, go to one of its neighbours, from there go to another neighbour, etc.
- Since the vertex set is finite, at some stage we encounter a vertex we have already visited.

Leaves in trees

A leaf in a tree is a vertex of degree 1.

Lemma

Every tree on at least two vertices contains a leaf.

Proof.

By contradiction:

- Assuming that every vertex has degree 0 or at least 2, we will show that the graph is not a tree.
- If a vertex has degree 0, then: the graph (which contains at least two vertices) is not connected, hence not a tree.
- If every vertex has degree at least 2: just start at a vertex, go to one of its neighbours, from there go to another neighbour, etc.
- Since the vertex set is finite, at some stage we encounter a vertex we have already visited.
- This implies that the graph contains a cycle, so is not a tree, contradiction. □

Leaves in trees

A leaf in a tree is a vertex of degree 1.

Lemma

Every tree on at least two vertices contains a leaf.

Proof.

By contradiction:

- Assuming that every vertex has degree 0 or at least 2, we will show that the graph is not a tree.
- If a vertex has degree 0, then: the graph (which contains at least two vertices) is not connected, hence not a tree.
- If every vertex has degree at least 2: just start at a vertex, go to one of its neighbours, from there go to another neighbour, etc.
- Since the vertex set is finite, at some stage we encounter a vertex we have already visited.
- This implies that the graph contains a cycle, so is not a tree, contradiction. □

Can a tree have exactly one leaf?

Leaves in trees

A leaf in a tree is a vertex of degree 1.

Lemma

Every tree on at least two vertices contains a leaf.

Alternative proof (when every vertex has degree at least 2):

Proof.

- Consider a longest path P and an end vertex v of P.

Leaves in trees

A leaf in a tree is a vertex of degree 1.

Lemma

Every tree on at least two vertices contains a leaf.

Alternative proof (when every vertex has degree at least 2):

Proof.

- Consider a longest path P and an end vertex v of P.
- All neighbours of v are on P. (why?)

Leaves in trees

A leaf in a tree is a vertex of degree 1.

Lemma

Every tree on at least two vertices contains a leaf.

Alternative proof (when every vertex has degree at least 2):

Proof.

- Consider a longest path P and an end vertex v of P.
- All neighbours of v are on P. (why?)
- If $\deg(v) \geq 2$, then there is a cycle.
- The same also holds for the second end vertex u of P

$\Rightarrow$ $\deg(u) = \deg(v) = 1$

□

Edges of trees

How many edges does a tree on n vertices have?

Edges of trees

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree iff it has $n - 1$ edges.

Edges of trees

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree iff it has $n-1$ edges.

Proof.

($\Rightarrow$). Show, by induction on n, that a tree on n vertices has $n-1$ edges.

Edges of trees

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree iff it has $n-1$ edges.

Proof.

($\Rightarrow$). Show, by induction on n, that a tree on n vertices has $n-1$ edges.

- For small n the lemma holds: a tree on one vertex has no edges; a tree on two vertices has one edge.

Edges of trees

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree iff it has $n-1$ edges.

Proof.

($\Rightarrow$). Show, by induction on n, that a tree on n vertices has $n-1$ edges.

- For small n the lemma holds: a tree on one vertex has no edges; a tree on two vertices has one edge.
- Suppose each tree on $n-1$ vertices has $n-2$ edges (induction hypothesis).
- Take a tree T on n vertices, for some $n \geq 3$.

Edges of trees

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree iff it has $n-1$ edges.

Proof.

($\Rightarrow$). Show, by induction on n, that a tree on n vertices has $n-1$ edges.

- For small n the lemma holds: a tree on one vertex has no edges; a tree on two vertices has one edge.
- Suppose each tree on $n-1$ vertices has $n-2$ edges (induction hypothesis).
- Take a tree T on n vertices, for some $n \geq 3$.
- T contains a leaf v. Consider the graph $T - v$, it has one vertex less and one edge less than T.

Edges of trees

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree iff it has $n-1$ edges.

Proof.

($\Rightarrow$). Show, by induction on n, that a tree on n vertices has $n-1$ edges.

- For small n the lemma holds: a tree on one vertex has no edges; a tree on two vertices has one edge.
- Suppose each tree on $n-1$ vertices has $n-2$ edges (induction hypothesis).
- Take a tree T on n vertices, for some $n \geq 3$.
- T contains a leaf v. Consider the graph $T - v$, it has one vertex less and one edge less than T.
- $T - v$ is still connected and (still) acyclic.

Edges of trees

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree iff it has $n-1$ edges.

Proof.

($\Rightarrow$). Show, by induction on n, that a tree on n vertices has $n-1$ edges.

- For small n the lemma holds: a tree on one vertex has no edges; a tree on two vertices has one edge.
- Suppose each tree on $n-1$ vertices has $n-2$ edges (induction hypothesis).
- Take a tree T on n vertices, for some $n \geq 3$.
- T contains a leaf v. Consider the graph $T-v$, it has one vertex less and one edge less than T.
- $T-v$ is still connected and (still) acyclic.
- $T-v$ is a tree with $n-1$ vertices, by induction hypothesis it has $n-2$ edges.
- T has one edge more, so $n-1$ edges.

Edges of trees, cont'd

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree if and only if it has $n - 1$ edges.

Proof.

$(\Leftarrow)$.

Edges of trees, cont'd

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree if and only if it has $n-1$ edges.

Proof.

$(\Leftarrow)$.

- Assume that G is a connected graph with n vertices and $n-1$ edges.
- Then, as we proved before, G contains a spanning tree T.

Edges of trees, cont'd

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree if and only if it has $n-1$ edges.

Proof.

$(\Leftarrow)$.

- Assume that G is a connected graph with n vertices and $n-1$ edges.
- Then, as we proved before, G contains a spanning tree T.
- By the first part of the proof, T contains exactly $n-1$ edges.

Edges of trees, cont'd

How many edges does a tree on n vertices have?

Theorem

A connected graph on n vertices is a tree if and only if it has $n-1$ edges.

Proof.

$(\Leftarrow)$.

- Assume that G is a connected graph with n vertices and $n-1$ edges.
- Then, as we proved before, G contains a spanning tree T.
- By the first part of the proof, T contains exactly $n-1$ edges.
- T is a subgraph of G, and it has the same number of edges as G.
- Hence, T and G are the same.
- In particular, G is a tree.

$\square$

Paths in trees

Paths in trees

Since a tree is a connected graph, between any two vertices in a tree there is a path. Can there be more than one path between two vertices in a tree?

Paths in trees

Since a tree is a connected graph, between any two vertices in a tree there is a path. Can there be more than one path between two vertices in a tree?

Lemma

Let T be a tree and $u, v \in V(T)$ with $u \neq v$.
Then there is a unique path in T between u and v.

Paths in trees

Since a tree is a connected graph, between any two vertices in a tree there is a path. Can there be more than one path between two vertices in a tree?

Lemma

Let T be a tree and $u, v \in V(T)$ with $u \neq v$.
Then there is a unique path in T between u and v.

Proof.

By contradiction.

- There is a path between u and v in T, since T is connected.
- Suppose there are two paths P and Q in T between u and v, and derive a contradiction.

Paths in trees

Since a tree is a connected graph, between any two vertices in a tree there is a path. Can there be more than one path between two vertices in a tree?

Lemma

Let T be a tree and $u, v \in V(T)$ with $u \neq v$.
Then there is a unique path in T between u and v.

Proof.

By contradiction.

- There is a path between u and v in T, since T is connected.
- Suppose there are two paths P and Q in T between u and v, and derive a contradiction.
- Let x and y in $V(T)$ be distinct and chosen in such a way that x and y are on both P and Q, but between x and y the vertices on P and Q are disjoint. (It is possible that $x = u$ and $y = v$, but this is not necessarily the case.)
- Then the segments of P and Q between x and y together form a cycle.
- This contradicts that T is a tree. Hence there is a unique (u, v)-path in T.

$\square$

Exercises

We have shown that, for a graph G on n vertices, the following conditions are equivalent:

1. G is tree;
2. G is connected and has $n-1$ edges.

Exercise 2: Show that these conditions are also equivalent to each of the following:

3. G is acyclic and has $n-1$ edges;
4. any two distinct vertices of G are connected by a unique path;
5. for any distinct $u, v \in V$, if $uv \notin E(G)$ then the graph $G + uv$ contains a unique cycle.

Rooted trees

Rooted trees

Definition

A (directed) rooted tree is a tree in which one vertex is fixed as the root (vertex) (and every edge is directed away from this root).

Rooted trees

Definition

A (directed) rooted tree is a tree in which one vertex is fixed as the root (vertex) (and every edge is directed away from this root).

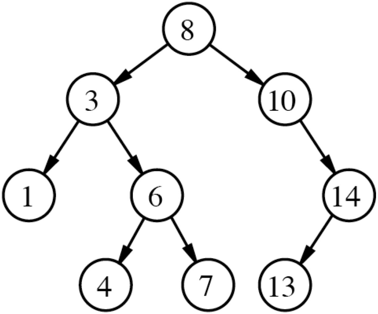
We usually draw a rooted tree in (horizontal) levels, starting with the root (level 0), then the neighbours of the root (level 1), etc.

Rooted trees

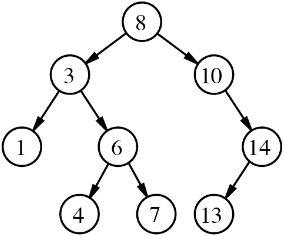
Definition

A (directed) rooted tree is a tree in which one vertex is fixed as the root (vertex) (and every edge is directed away from this root).

We usually draw a rooted tree in (horizontal) levels, starting with the root (level 0), then the neighbours of the root (level 1), etc.



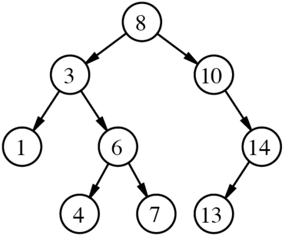
Rooted trees, children and parents



Definitions

Let v be a vertex in a rooted tree T.

Rooted trees, children and parents

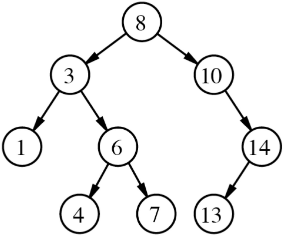


Definitions

Let v be a vertex in a rooted tree T.

- The neighbours of v in the next level are called the children of v.

Rooted trees, children and parents

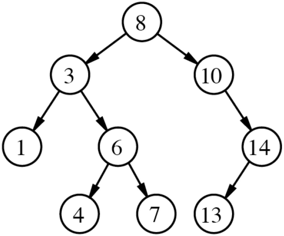


Definitions

Let v be a vertex in a rooted tree T.

- The neighbours of v in the next level are called the children of v.
- the (unique) neighbour of v in the previous level (if v is not the root) is called the parent of v.

Rooted trees, children and parents

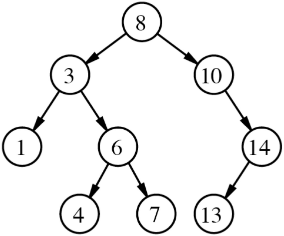


Definitions

Let v be a vertex in a rooted tree T.

- The neighbours of v in the next level are called the children of v.
- the (unique) neighbour of v in the previous level (if v is not the root) is called the parent of v.
- If v has no children then it is called a leaf of T;

Rooted trees, children and parents



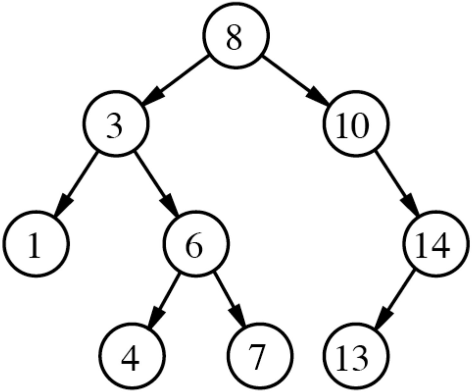
Definitions

Let v be a vertex in a rooted tree T.

- The neighbours of v in the next level are called the children of v.
- the (unique) neighbour of v in the previous level (if v is not the root) is called the parent of v.
- If v has no children then it is called a leaf of T;
- If v has children, then it is an internal vertex.

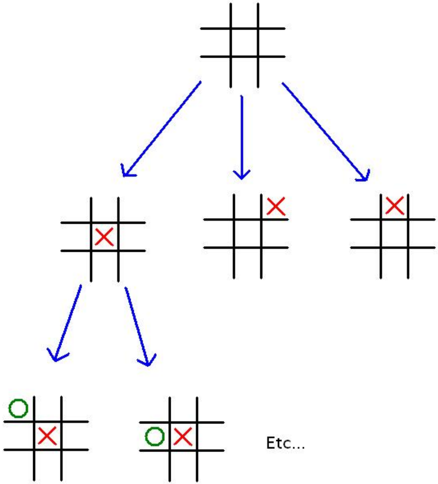
Some applications of trees

- Binary search trees (we have seen these earlier in ADS)



Some applications of trees

- Search trees (more on this in AI Search)



Some applications of trees

- Phylogenetic trees (Bioinformatics)