Mathematics for Computer Science

Prof.dr.hab. Viorel Bostan

Technical University of Moldova

viorel.bostan@adm.utm.md

Lectures 12–13

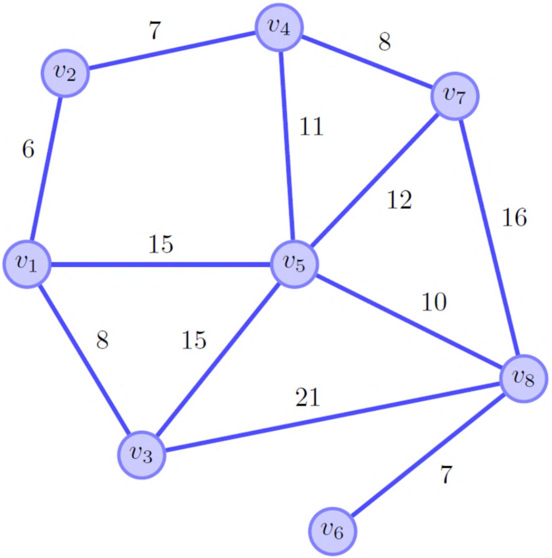
Previous Lecture Summary

- Graph theory history, 6 degrees of separation, graph applications;
- Graph definitions, Graph representations;
- Complete graph, Isomorphic graphs;
- Sub-graphs, induced and spanning sub-graphs;
- Operations on graphs, adjacency and incident matrices;
- Vertex degree, average degree, regular graph, Handshaking lemma.

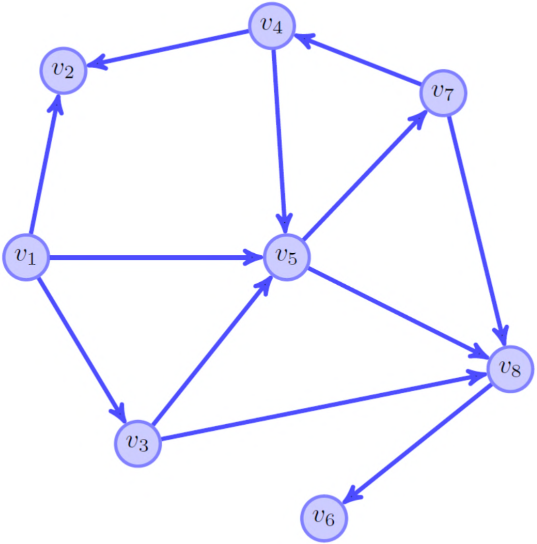
In previous lecture we considered the graphs, called **simple graphs**, in which the edges are without direction, in other words, edges, for example $(2, 5)$ and $(5, 2)$ are the same edge. Also, we might consider, directional graphs (digraphs), where these two edges are considered different. In addition, we might assign to each edge a weight, generally a positive number.

Also, there are other special types of graphs, such as graphs with loops and/or multiple edges. In what follows, unless specified otherwise we will consider only simple graphs or digraphs with or without weights.

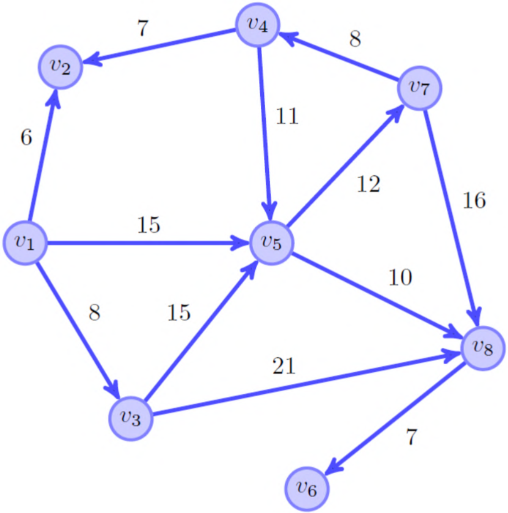
Weighted graph

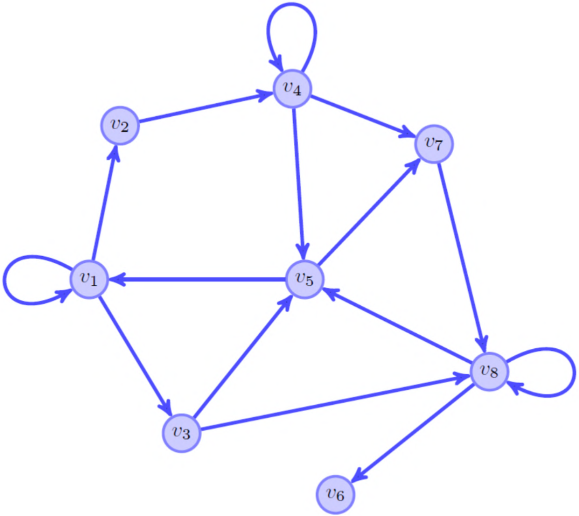


Directed graph digraph)

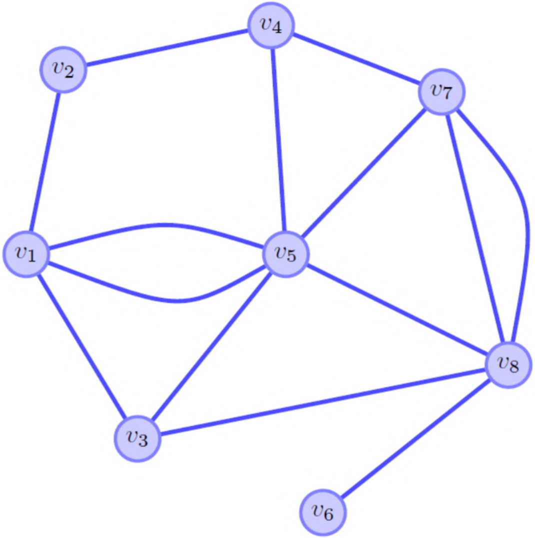


Weighted digraph

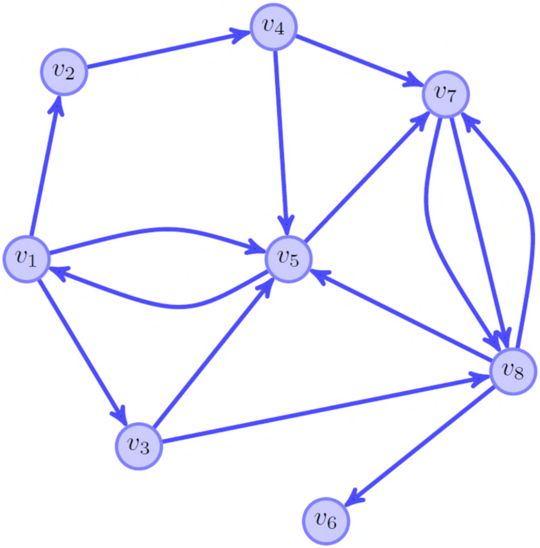



v_4
v_2
v_7
v_1
v_5
v_8
v_3
v_6

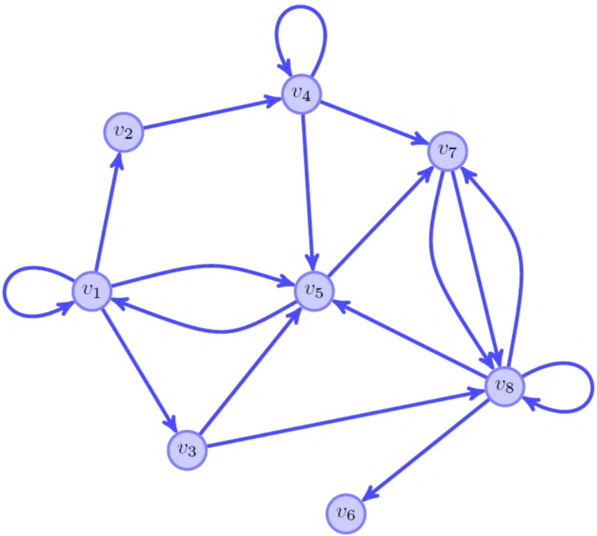
Multi-graph (with multiple edges)



Multi-digraph



Multi-digraph with loops



Graph walks

Definition

A **walk** is a non-empty graph $W = (V, E)$ of the form

$$V = \{x_0, x_1, \ldots, x_k\}, \quad E = \{x_0x_1, x_1x_2, \ldots, x_{k-1}x_k\}.$$

We call x_0 and x_k the **endpoints** of the walk.

In other words, a walk is a sequence of consecutive vertices and edges.
In the case of multiple edges we must keep track of which edges are being used.

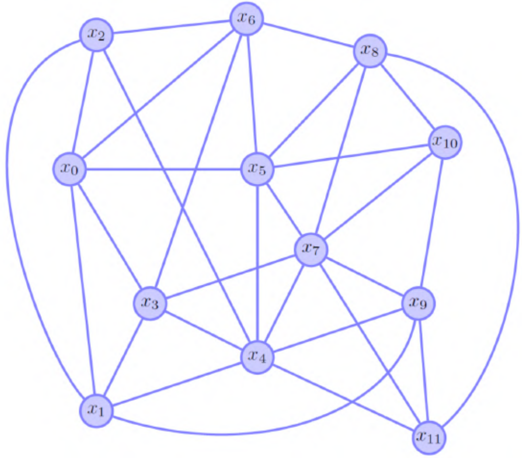
We will use a short hand notation for a walk: $x_0x_1 \ldots x_k$.

Definition

The word **length** refers to the number of edges in the walk.
A walk is called **closed** if $x_0 = x_k$.
A walk is called **spanning** if it contains every vertex of the graph.

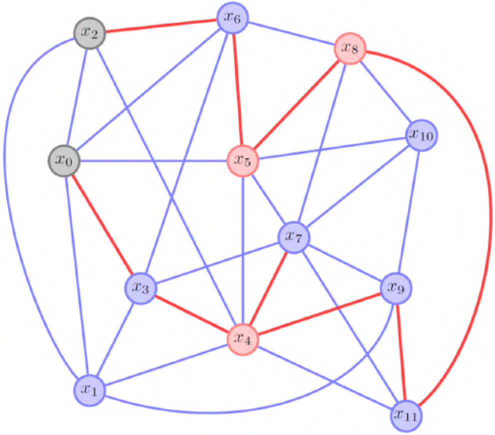
Graph walks

Consider walk $x_0x_3x_4x_7x_4x_9x_{11}x_8x_5x_8x_5x_6x_2$ in graph G:



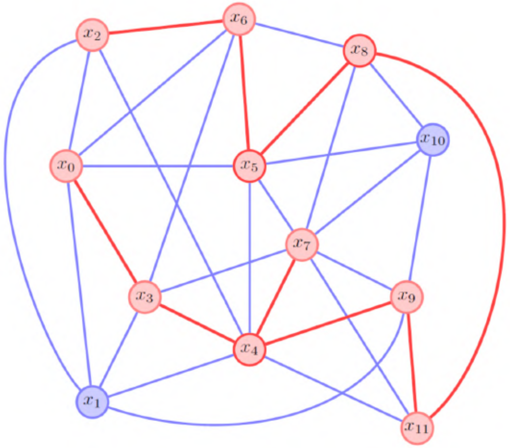
Walk on graph G.

Consider walk $x_0x_3x_4x_7x_4x_9x_{11}x_8x_5x_8x_5x_6x_2$ in graph G

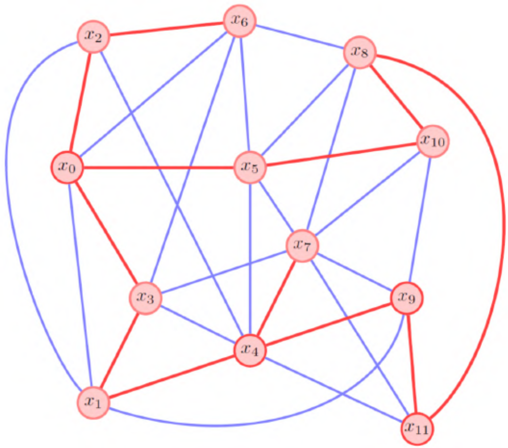


Walk on graph G: endpoints–grey.

Consider walk $x_0x_3x_4x_7x_4x_9x_{11}x_8x_5x_8x_5x_6x_2$ in graph G



Walk on graph G: vertices visited twice–red bordered.

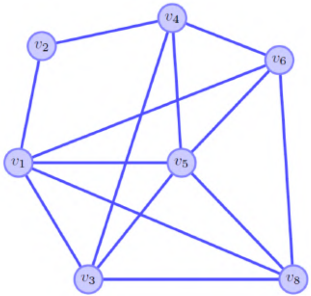


Spanning walk $x_0 x_3 x_1 x_4 x_7 x_4 x_9 x_{11} x_8 x_{10} x_5 x_0 x_2 x_6$ of length 13 on graph G.

Definition

A **path** is a walk with no repeated vertices. A **trail** is a path with no repeated edges.

Any path is a trail, but not vice-versa, i.e. we can have a trail, that is not a path.

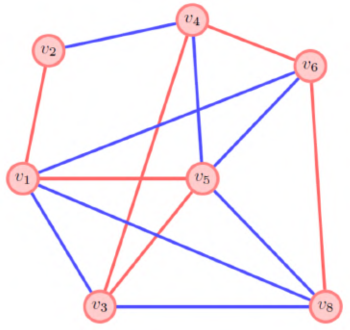


Path (spanning) $v_2 v_1 v_5 v_3 v_4 v_6 v_8$ of length 6.

Definition

A **path** is a walk with no repeated vertices. A **trail** is a path with no repeated edges.

Any path is a trail, but not vice-versa, i.e. we can have a trail, that is not a path.

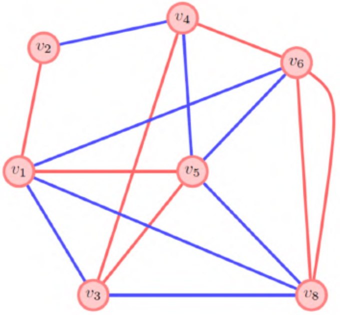


Path (spanning) $v_2 v_1 v_5 v_3 v_4 v_6 v_8$ of length 6.

Definition

A **path** is a walk with no repeated vertices. A **trail** is a path with no repeated edges.

Any path is a trail, but not vice-versa, i.e. we can have a trail, that is not a path.

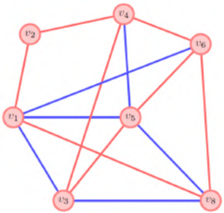


Trail (spanning) $v_2 v_1 v_5 v_3 v_4 v_6 v_8 v_6$ of length 7.

Definition

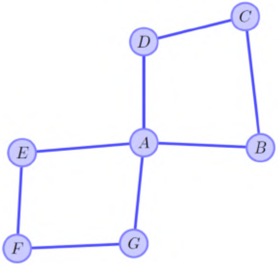
A **circuit (tour)** is a closed trail. A **cycle** is a circuit with at least one edge and in which the only repeated vertex is the first one. A cycle of length k is called a $k-$**cycle**, denoted C^k.

A 1−cycle is a loop, a 2−cycle is made up of multiple edges, a 3−cycle is a triangle, a 4−cycle is a quadrilateral. In simple graphs/digraphs, the shortest length for a cycle is 3, since no loops or multiple edges are allowed.



A 7−cycle $v_1 v_2 v_4 v_3 v_5 v_6 v_8 v_1$. A triangle $v_1 v_5 v_3 v_1$, a quadrilateral $v_1 v_5 v_3 v_8 v_1$, etc.

Walks, paths, tours, circuits, cycles,. . .



- $ABCD$ is a walk of length 3;
- $ABCB$ is a walk that is not neither a path nor a trail;
- $ABCDAE$ is a trail but not a path;
- AB is a path;
- ABA is a closed walk that is not a circuit (tour);
- $ABCDEFGA$ is a circuit (tour) but not a cycle;
- $ABCDA$ is a 4−cycle.

Definition

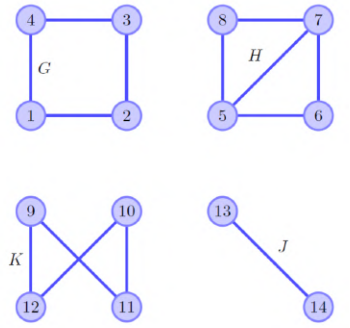
Given a graph G, an **Eulerian circuit** (tour) is a circuit that includes each edge of G exactly once, and a graph containing an eulerian circuit is called **Eulerian**.

Definition

A **Hamiltonian cycle** is a spanning cycle, and a graph containing a Hamiltonian cycle is called **Hamiltonian**.

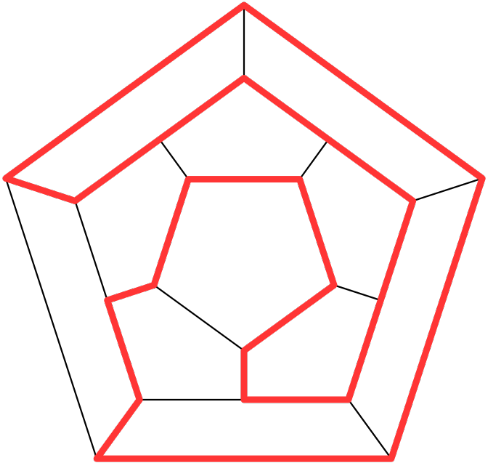
Note that other than the repeat of the first and last vertex, a Hamiltonian cycle includes each vertex of G exactly once.

Eulerian and Hamiltonian graphs

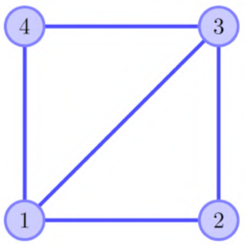


Graph G is both Eulerian and Hamiltonian,
graph H is Hamiltonian, but not Eulerian,
graph K is both Eulerian and Hamiltonian,
graph J is neither Eulerian nor Hamiltonian.

Hamiltonian graph



How many walks of length 3 there are from v_1 to v_4?



There are 5 walks of length 3 from v_1 to v_4:

$v_1 v_2 v_1 v_4$,

$v_1 v_3 v_1 v_4$,

$v_1 v_2 v_3 v_4$,

$v_1 v_4 v_1 v_4$,

$v_1 v_4 v_3 v_4$

Number of walks in a graph

Theorem

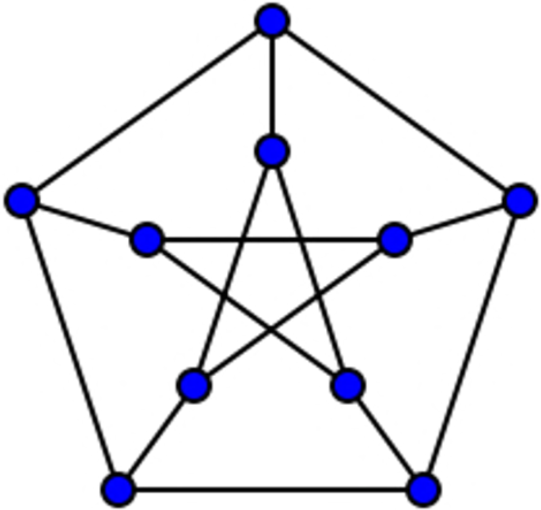
Let $G = (V, E)$ be an $n-$node graph with $V = \{v_1, v_2, \ldots, v_n\}$ and let $A_G = (a_{ij})$ denote the adjacency matrix for G. Let $a_{ij}^{(k)}$ denote the ij entry of the $k-$th power of A_G.
Then, the number of walks of length k between v_i and v_j is $a_{ij}^{(k)}$.

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad \Rightarrow A^3 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}^3 \quad = \begin{bmatrix} 4 & 5 & 5 & \color{red}{5} \\ 5 & 2 & 5 & 2 \\ 5 & 5 & 4 & 5 \\ 5 & 2 & 5 & 2 \end{bmatrix}.$$

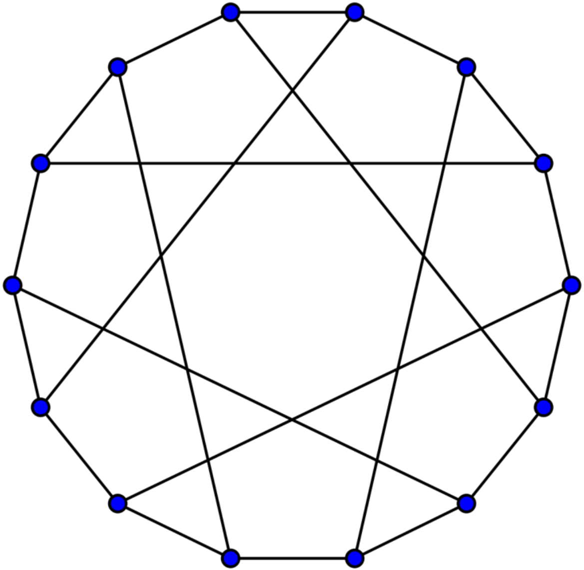
Definition

Girth of a graph G is the length of the shortest cycle contained in the graph.

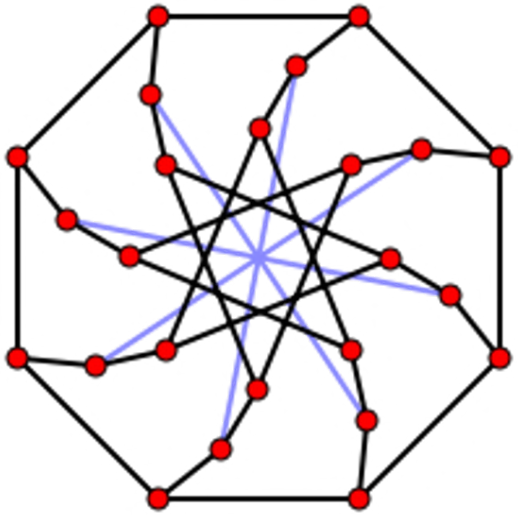
Petersen graph (girth 5)



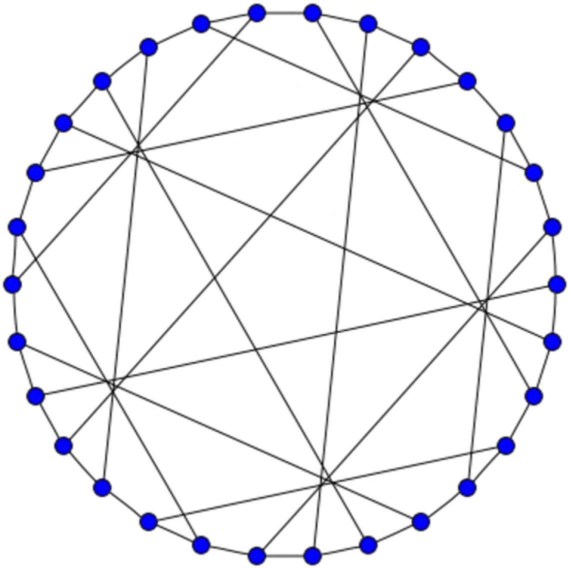
Heawood graph (girth 6)



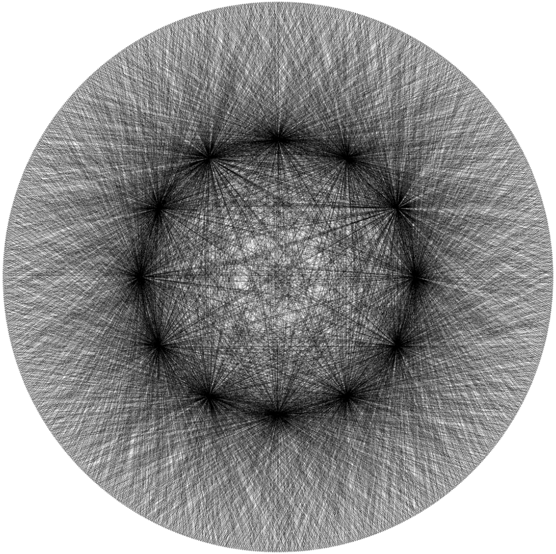
McGee graph (girth 7)



Tutte–Coxeter graph (girth 8)



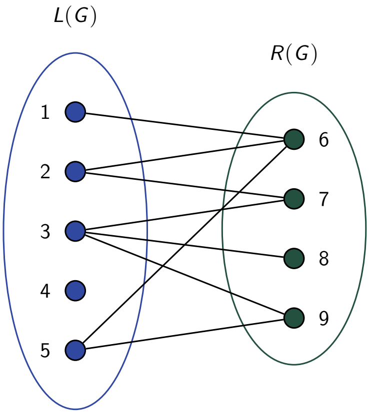
Graph with girth 6



Definition

A **bipartite graph** is a graph whose vertices can be partitioned into 2 sets, $L(G)$ and $R(G)$, such that every edge has one endpoint in $L(G)$ and the other endpoint in $R(G)$.

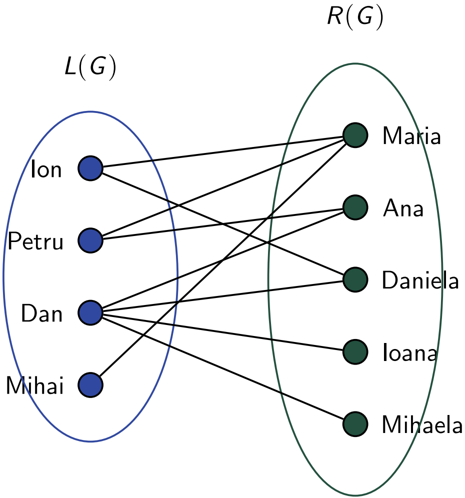
Bipartite graphs are used in the so-called **Matching Problem**.



Consider a set of men and an equal-sized or larger set of women, and there is a graph with an edge between a man and a woman if the man likes the woman.
Observe that the "likes" relationship need not be symmetric.

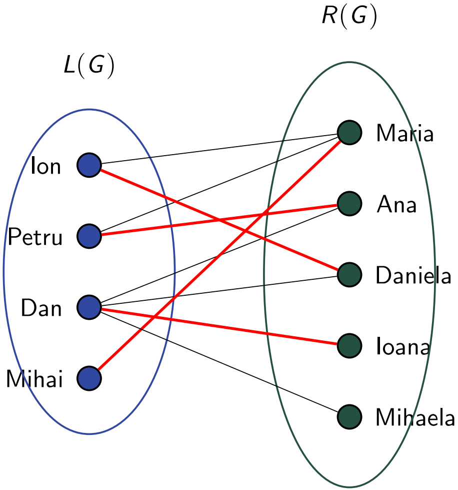
Matching Problem

A **matching** is defined to be an assignment of a woman to each man so that different men are assigned to different women, and a man is always assigned a woman that he likes.



Matching Problem

A **matching** is defined to be an assignment of a woman to each man so that different men are assigned to different women, and a man is always assigned a woman that he likes.



Matching Condition

Every subset of men likes at least as large a set of women.

Example

It is impossible to find a matching if some set of 4 men like only 3 women!

Theorem (Hall's Matching Theorem)

A matching for a set M of men with a set W of women can be found if and only if the matching condition holds.

Graph Coloring

Graph Coloring Problem

Given a graph G, assign colors to each node such that adjacent nodes have different colors.

Definition

A color assignment with this property is called a valid **coloring** of the graph.

Definition

A graph G is k-**colorable** if it has a coloring that uses at most k colors.

Definition

The minimum value of k for which a graph G has a valid $k-$coloring is called its **chromatic number**, denoted by $\chi(G)$.

Graph Coloring

Trying to figure out if you can color a graph with a fixed number of colors can take time.

It's a classic example of a problem for which no fast (i.e. polynomial time) algorithms are known.

It is easy to check if a coloring works, but it is really hard to find it.

In order to find a valid coloring, we can use the **greedy algorithm** approach.

But we are not assured that chromatic number (minimal number of colors) will be found.

On the other hand some graph properties can provide useful upper bounds on the chromatic number.

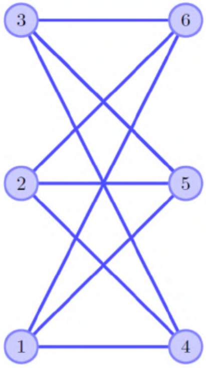
Example

Bipartite graphs are 2-colorable: one color for the nodes in the "left" set and a second color for the nodes in the "right" set.
Actually, this is a characteristics of a bipartite graphs.

Graph Coloring

Lemma

If the graph has any edges at all, then being bipartite is equivalent to being 2-colorable.



Vertices $1, 2$ and 3 colored in green, and vertices $4, 5, 6$ in red.

Theorem (4−Color Theorem)

If the graph is planar, *then the graph is 4-colorable.*

Note: definition of being planar will be given after few slides.

The chromatic number of a graph can also be shown to be small if the vertex degrees of the graph are small.

In particular, if we have an upper bound on the degrees of all the vertices in a graph, then we can easily find a coloring with only one more color than the degree bound.

Theorem

A graph with maximum degree at most k is $(k+1)$-colorable.

Proof.

Use induction on the number of vertices in the graph n.

Let $P(n)$ be the proposition that an n-vertex graph with maximum degree at most k is $(k+1)$-colorable.

Base case $(n = 1)$:
An 1-vertex graph has maximum degree 0 and it is 1-colorable, so $P(1)$ is true.

Inductive step: Assume that $P(n)$ is true.

Let G be an $(n+1)$-vertex graph with maximum degree at most k.

Proof contd.

Remove a vertex v and all edges incident to it. That leaves an n-vertex sub-graph H.

The maximum degree of H is at most k, and so H is $(k+1)$-colorable by assumption $P(n)$.

Add back vertex v.

We can assign v a color from the set of $k+1$ colors that is different from all its adjacent vertices, since there are at most k vertices adjacent to v and so at least one of the $k+1$ colors is still available.

Therefore, G is $(k+)$1-colorable.

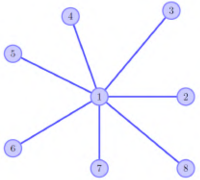
This completes the inductive step, and the theorem follows by induction.

Sometimes $k+1$ colors is the best you can do. Consider complete graph K_n with all vertices having degree $n-1$. Every one of its n vertices is adjacent to all the others, so all n vertices must be assigned different colors. Of course, n colors is also enough, so $\chi(K_n) = n$.

This is an example where previous theorem gives the best possible bound.

By a similar argument, we can show that previous theorem gives the best possible bound for any graph with degree bounded by k that has K_{k+1} as a sub-graph.

Sometimes $k+1$ colors is far from the best that you can do: consider a $n-$node star graph with maximum degree $n+1$.



Obviously it can be colored using just 2 colors.

Updating software

Coloring problems are used frequently in scheduling problems in order to avoid conflicts.

Consider a software, deployed over each of 75 000 servers every few days.

The updates cannot be done at the same time since the servers need to be taken down in order to deploy the software.

Also, the servers cannot be handled one at a time, since it would take forever to update them all (each one takes about an hour).

Moreover, certain pairs of servers cannot be taken down at the same time since they have common critical functions.

This problem was eventually solved by making a 75 000-node conflict graph and coloring it with 8 colors – so only 8 waves of install are needed!

Applications of coloring

Example (Frequency allocation)

Need to assign frequencies to radio stations.

If two stations have an overlap in their broadcast area, they can't be given the same frequency.

Frequencies are precious and expensive, so you want to minimize the number handed out.

This amounts to finding the minimum coloring for a graph whose vertices are the stations and whose edges connect stations with overlapping areas.

Example (Allocating registers for program variables)

While a variable is in use, its value needs to be saved in a register.

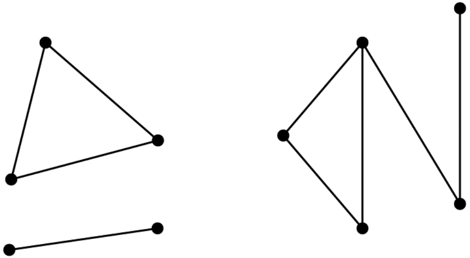
Registers can be reused for different variables but two variables need different registers if they are referenced during overlapping intervals of program execution.

So register allocation is the coloring problem for a graph whose vertices are the variables.

Vertices are adjacent if their intervals overlap, and the colors are registers

Definition

Two vertices are **connected** in a graph when there is a path that begins at one and ends at the other. A graph is called **connected** when every pair of vertices are connected.
A **connected component** of a graph is a sub-graph consisting of some vertex and every node and edge that is connected to that vertex.



A graph with 3 connected components.

Graph Connectivity

Theorem

Every graph with v vertices and e edges has at least $v - e$ connected components.

Theorem says that a graph with few edges must have many connected components.

Proof.

Use induction on the number of edges e.

Let $P(e)$ be the proposition that for every v, every graph with v vertices and e edges has at least $v - e$ connected components.

Base case $(e = 0)$: In a graph with 0 edges and v vertices, each vertex is itself a connected component, and so there are exactly $v = v - 0$ connected components. So $P(0)$ holds.

Inductive step: Assume that induction hypothesis holds for every e-edge graph in order to prove that it holds for every $(e + 1)$-edge graph, where $e \geqslant 0$.

Consider a graph G with $e + 1$ edges and v vertices.

Want to prove that G has at least $v - (e + 1)$ connected components.

Proof contd.

To do this, remove an arbitrary edge $\{a, b\}$ and call the resulting graph G'.

Clearly, G' has e edges and v vertices. By induction assumption, G' has at least $v - e$ connected components.

Now add back the edge$\{a, b\}$ to obtain the original graph G.

If a and b were in the same connected component of G', then G has the same connected components as G', so G has at least $v - e > v - (e + 1)$ components.

Otherwise, if a and b were in different connected components of G', then these two components are merged into one component in G, but all other components remain unchanged, reducing the number of components by 1.

Therefore, G has at least $(v - e) - 1 = v - (e + 1)$ connected components.

So in either case, $P(e + 1)$ holds. This completes the Inductive step. $\square$

We used induction on the number of edges in the graph.

This is very common in proofs involving graphs, as is induction on the number of vertices.

When you're presented with a graph problem, these two approaches should be among the first you consider.

Corollary

Every connected graph with v vertices has at least $v-1$ edges.

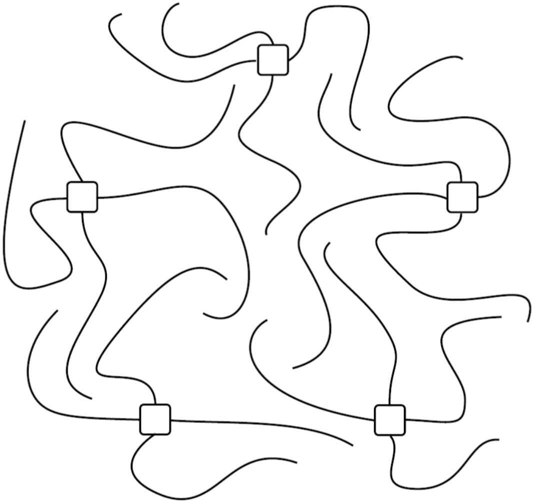
Theorem

The following properties of a graph are equivalent:

1. *The graph is bipartite.*
2. *The graph is $2-$colorable.*
3. *The graph does not contain any cycles with odd length.*
4. *The graph does not contain any closed walks with odd length.*

Planar Graphs

Planar graphs



Planar graphs

Definition

A **drawing** of a graph in the plane consists of an assignment of vertices to distinct points in the plane and an assignment of edges to smooth, nonself-intersecting curves in the plane (whose endpoints are the nodes incident to the edge).

Definition

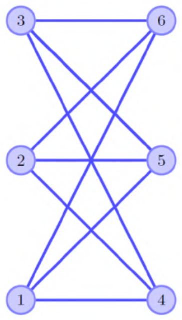
The drawing is **planar** if none of the curves "cross"—that is, if the only points that appear on more than one curve are the vertex points.

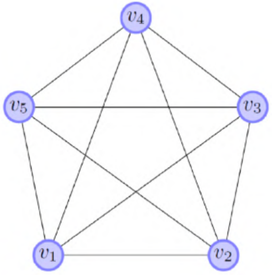
Definition

A **planar graph** is a graph that has a planar drawing.

In other words a graph is planar if it can be redrawn such that its edges will not cross.

The previous 2 questions are asking whether the graphs below are planar:



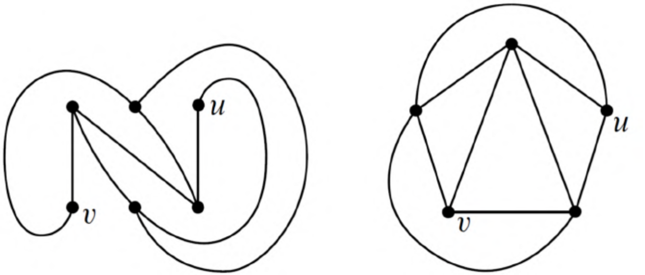


that is, whether they can be redrawn so that no edges cross.

And the answer is: **NO**. Almost NO!

Almost No, means that if we remove one edge, then the graphs become planar.

Planar graphs



Graphs $K_{3,3} - (u, v)$ and $K^5 - (u, v)$ are planar, but $K_{3,3}$ and K_5 are not!

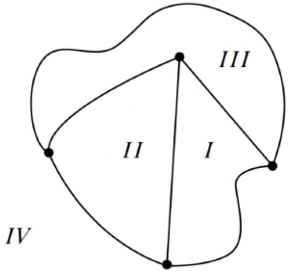
There is a famous theorem characterizing planar graphs:

Theorem (Kuratowsky Theorem)

Graph G is planar if and only if G "contains" *neither K^5 nor $K_{3,3}$ as a topological minor.*

Definition

In a planar drawing of a graph, the curves corresponding to the edges divide up the plane into connected regions. These regions are called the continuous **faces** of the drawing.



Face IV extends to infinity in all directions and it is called the outside face.

Euler's formula for planar graphs

Theorem (Euler's Formula)

If a connected graph is planar, then

$$v - e + f = 2,$$

where v is the number of vertices, e is the number of edges, and f is the number of faces.

Indeed, on the previous graph, we have $v = 4, e = 6, f = 4$, that gives $4 - 6 + 4 = 2$.

Euler's formula for planar graphs

From Euler's formula we can derive useful results for planar graphs.

Lemma (1)

In a planar drawing of a connected graph, each edge is traversed once by each of two different faces, or is traversed exactly twice by one face.

Lemma (2)

In a planar drawing of a connected graph with at least three vertices, each face is of length at least three.

Theorem

Suppose a connected planar graph has $v \geqslant 3$ vertices and e edges. Then

$$e \leqslant 3v - 6.$$

Euler's formula for planar graphs

Proof.

Suppose we have a connected planar graph with v vertices, e edges and f faces. By Lemma (1), every edge is traversed exactly twice by the face boundaries. So the sum of lengths of face boundaries is $2e$. Also by Lemma (2), when $v \geqslant 3$, each face boundary is of length at least 3, so this sum is at least $3f$. This implies that:

$$3f \leqslant 2e$$

But, by Euler's formula: $f = 2 - v + e$ and substituting it in the above inequality gives

$$\begin{aligned} 3(2 - v + e) &\leqslant 2e \\ 6 - 3v + 3e &\leqslant 2e \\ e - 3v + 6 &\leqslant 0 \\ e &\leqslant 3v - 6. \end{aligned}$$

□

$K_{3,3}$ and K_5 not planar!

The last result implies almost immediately that $K_{3,3}$ and K_5 are not planar:

Corollary

K^5 is not planar.

Proof.

K^5 is connected, has 5 vertices and 10 edges, and $10 > 3 \cdot 5 - 6$.
This violates the condition of the last Theorem required for K^5 to be planar. $\square$

Corollary

$K_{3,3}$ is not planar.

Proof.

Proof by contradiction. Assume $K_{3,3}$ is planar and consider any planar embedding of $K_{3,3}$ with f faces. Since $K_{3,3}$ is bipartite, we know that $K_{3,3}$ does not contain any closed walks of odd length.

$K_{3,3}$ and K_5 not planar!

Proof contd.

By Lemma (2), every face has length at least 3.

This means that every face in any drawing of $K_{3,3}$ must have length at least 4.

Therefore, the sum of the lengths of the face boundaries is exactly $2e$ and at least $4f$.

Hence, $4f \leqslant 2e$.

Plugging in $e = 9$ and $v = 6$ for $K_{3,3}$ in Euler's Formula, we find that $f = 2 - v + e = 5$.

But

$$4 \cdot 5 \not\leqslant 2 \cdot 9$$

and so we have a contradiction. Hence $K_{3,3}$ must not be planar. $\square$

Lemma

Every planar graph has a vertex of degree at most 5.

Proof.

Proof by contradiction. Suppose we have a planar connected graph with every vertex of degree at least 6. Then the sum of the vertex degrees is at least $6v$. Since the sum of vertex degrees equals $2e$, by Handshake lemma, we have

$$
\begin{aligned}
2e &= \sum_{v \in V} d(v) \geqslant 6v, \\
e &\geqslant 3v.
\end{aligned}
$$

On the other hand, for planar connected graphs $e \leqslant 3v - 6 < 3v$. Contradiction! $\square$

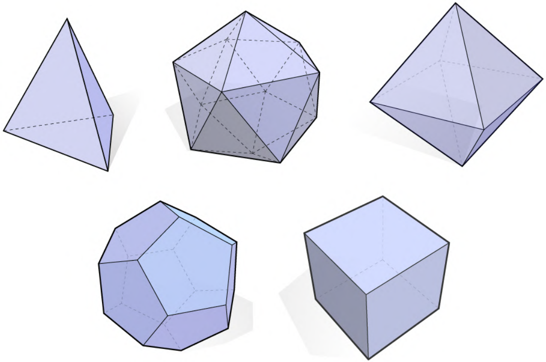
Theorem

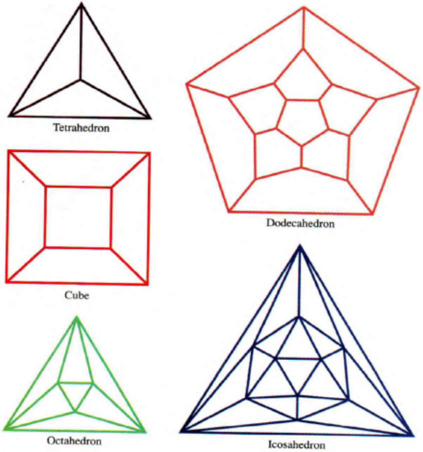
Every planar graph is 5-colorable.

Polyhedrons

Definition

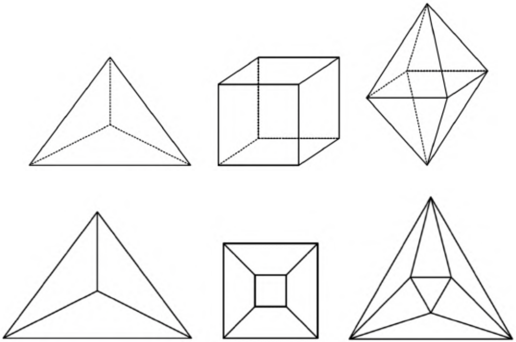
A **polyhedron** is a convex, 3-dimensional region bounded by a finite number of polygonal faces. If the faces are identical regular polygons and an equal number of polygons meet at each corner, then the polyhedron is called **regular**.




Tetrahedron
Dodecahedron
Cube
Octahedron
Icosahedron

Regular Polyhedrons

Question: how many more regular polyhedrons do exist?



Let m be the number of faces that meet at each corner of a polyhedron, and let n be the number of edges on each face.

In the corresponding planar graph, there are m edges incident to each of the v vertices.

Regular Polyhedrons

By the Handshake Lemma, we know:

$$mv = 2e.$$

Also, each face is bounded by n edges.

Since each edge is on the boundary of two faces, we have:

$$nf = 2e.$$

Solving for v and f in these equations and then substituting into Euler's formula gives:

$$\frac{2e}{m} - e + \frac{2e}{n} = 2,$$

which simplifies to

$$\frac{1}{m} + \frac{1}{n} = \frac{1}{e} + \frac{1}{2}.$$

$$\frac{1}{m} + \frac{1}{n} = \frac{1}{e} + \frac{1}{2}.$$

Every non-degenerate polygon has at least 3 sides, so $n \geqslant 3$.

And at least 3 polygons must meet to form a corner, so $m \geqslant 3$.

On the other hand, if either n or m were 6 or more, then the left side of the equation could be at most

$$\frac{1}{3} + \frac{1}{6} = \frac{1}{2},$$

which is less than the right side. Therefore,

$$3 \leqslant n < 6, \quad n \in \mathbb{N},$$
$$3 \leqslant m < 6, \quad m \in \mathbb{N}.$$

Regular Polyhedrons

$$\frac{1}{m} + \frac{1}{n} = \frac{1}{e} + \frac{1}{2} \quad n, m \in \mathbb{N},$$
$$3 \leqslant n < 6, \quad 3 \leqslant m < 6.$$

Checking all cases that remain turns up only 5 solutions:

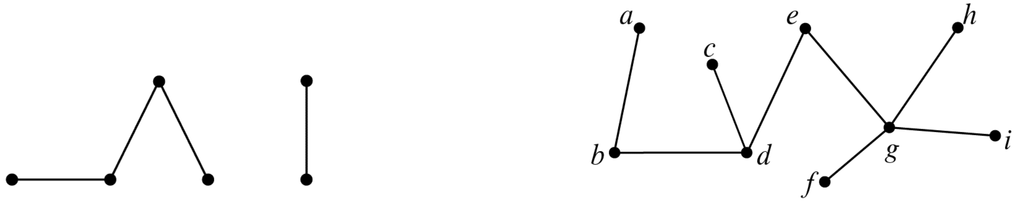
m	n	v	e	f	polyhedrons
3	3	4	6	4	tetrahedron
4	3	8	12	6	cube
3	4	6	12	8	octahedron
3	5	12	30	20	icosohedron
5	3	20	30	12	dodecahedron

Definition

A graph is called **acyclic** if it does not contain any cycles.

Definition

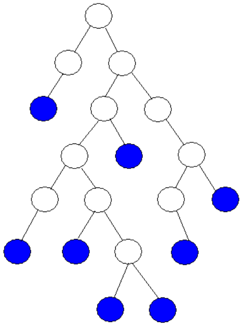
An acyclic graph is also called **forest**. A connected acyclic graph is called a **tree**.
A vertex in a forest with degree 1 is called **a leaf**.



A forest with two trees (left) and a tree with 5 leaves $\{a, c, f, h, i\}$ (right).

Definition

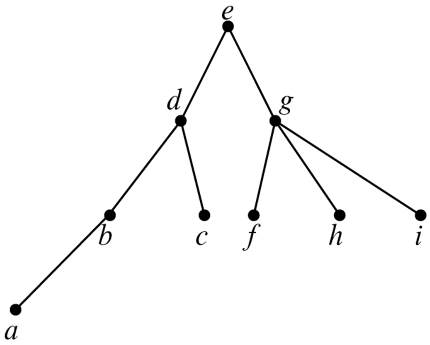
A tree is called **binary** if its maximal degree is three.

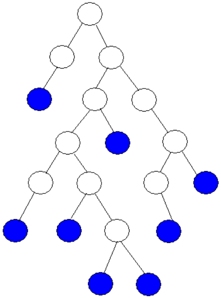


A binary tree with 8 leaves.

Remark

It is often useful to arrange the vertices in levels, where the vertex at the top level is designated as a **root** and where every edge joins a **parent** to a **child** one level below.





A tree with 5 leaves (left). A binary tree with 8 leaves (right).

Tree Properties

Theorem

Every tree has the following properties:

- *Every connected su-bgraph is a tree.*
- *There is a unique path between every pair of vertices.*
- *Adding an edge between nonadjacent vertices in a tree creates a graph with a cycle.*
- *Removing any edge disconnects the graph.*
- *If a tree has at least 2 vertices, then it has at least 2 leaves.*
- *The number of vertices in a tree is one larger than the number of edges. In other words,*

$$|T| = \|T\| + 1.$$

Spanning Tree

Definition

A tree is called a **spanning tree** of graph G if it is a spanning sub-graph of G. In other words, a spanning tree contains all vertices of graph G.

Theorem

Every connected graph contains a spanning tree.

Definition

Given a weighted connected graph, the **minimal spanning tree** is the spanning tree such that the sum all weights of its edges is minimal.

Theorem

Every connected weighted graph has a minimal spanning tree.

Next lecture: algorithms used to build a spanning tree (a minimal spanning tree).

Lecture Summary

- Walk, path, trail, circuit, cycle;
- Eulerian and Hamiltonian graphs;
- Bipartite graphs, matching problem, Hall's marriage theorem;
- Graph coloring, properties, applications of coloring;
- Graph connectivity;
- Planar graphs, Euler's formula for planar graphs;
- Regular polyhedrons and Pythagoras secret;
- Trees, properties, spanning tree.