MTH314: Discrete Mathematics for Engineers

Graph Theory II

Dr Ewa Infeld

Ryerson University

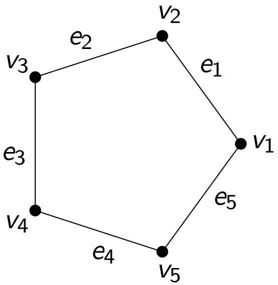
For most of this lecture, we will be talking about simple, undirected graphs.

A graph $G = (V, E)$ is a data structure/mathematical object that consists of a set of vertices/nodes V and a relation E (edges) on this set.

Because the graph is simple and undirected, at most one edge exists between any two vertices, there are no loops from a vertex to itself and the relation E is symmetric.

Walks

A walk in a graph is a sequence of vertices and edges that describe a path in a graph (i.e. a vertex can only follow another vertex if it's adjacent to it via the edge listed between them.)



V is the vertex set

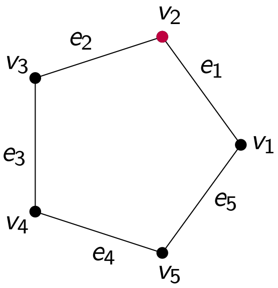
$$V = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E = \{e_1, e_2, e_3, e_4, e_5\}$$

is the set of edges

Walks

A walk in a graph is a sequence of vertices and edges that describe a path in a graph (i.e. a vertex can only follow another vertex if it's adjacent to it via the edge listed between them.)



V is the vertex set
$V = \{v_1, v_2, v_3, v_4, v_5\}$

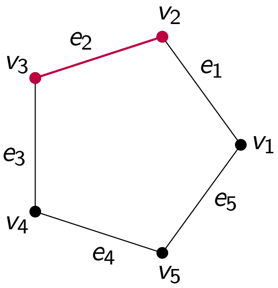
$E = \{e_1, e_2, e_3, e_4, e_5\}$
is the set of edges

Walk:

v_2

Walks

A walk in a graph is a sequence of vertices and edges that describe a path in a graph (i.e. a vertex can only follow another vertex if it's adjacent to it via the edge listed between them.)



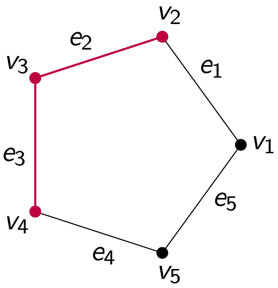
V is the vertex set
$V = \{v_1, v_2, v_3, v_4, v_5\}$

$E = \{e_1, e_2, e_3, e_4, e_5\}$
is the set of edges

Walk:
v_2, e_2, v_3

Walks

A walk in a graph is a sequence of vertices and edges that describe a path in a graph (i.e. a vertex can only follow another vertex if it's adjacent to it via the edge listed between them.)



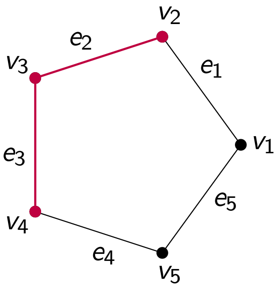
V is the vertex set
$V = \{v_1, v_2, v_3, v_4, v_5\}$

$E = \{e_1, e_2, e_3, e_4, e_5\}$
is the set of edges

Walk:
v_2, e_2, v_3, e_3, v_4

Walks

A walk in a graph is a sequence of vertices and edges that describe a path in a graph (i.e. a vertex can only follow another vertex if it's adjacent to it via the edge listed between them.)



V is the vertex set
$V = \{v_1, v_2, v_3, v_4, v_5\}$

$E = \{e_1, e_2, e_3, e_4, e_5\}$
is the set of edges

Walk:
v_2, e_2, v_3, e_3, v_4

If the graph is simple, it's enough to list the vertices: v_2, v_3, v_4.

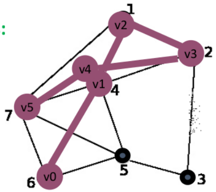
Traveling in a Graph

Definitions: Let $G = (V, E)$ be a graph and let $u, v \in V$.

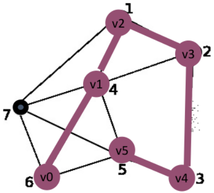
- A **walk** from u to v is an alternating sequence of vertices and edges
$$u = a_1, e_1, a_2, e_2, \dots, e_{n-1}, a_n = v$$
where each edge e_i has endpoints $\{a_i, a_{i+1}\}$.
- A **trail** from u to v is a walk from u to v that does not use an edge twice
- A **path** from u to v is a walk from u to v that does not use a vertex twice.
- A **closed walk** is a walk that has the same start and end vertex.
- A **circuit** (cycle) is a closed walk with at least one edge, but no repeated edges.
- A **simple circuit** is a circuit with no repeated vertices, except the first and the last.

Examples:

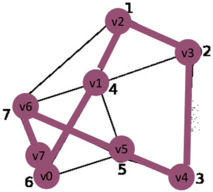
G=(V,E)



A walk in G



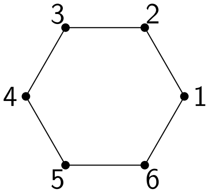
A path in G

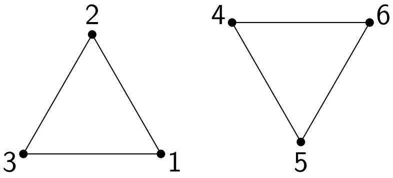


A simple circuit in G

Connected Graphs

A graph is if for any two vertices v, w there exists a walk which begins at v and ends at w.

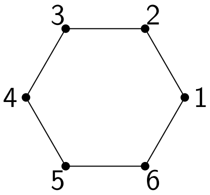


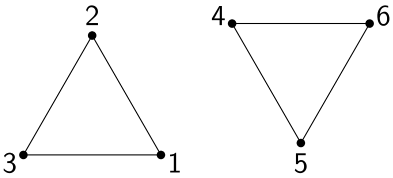


In the six vertex-graph on the left, there exists a walk from any vertex to any other vertex. In the graph on the right, there is no walk from 1 to 4. The graph on the right is not connected.

Connected Graphs

A graph is if for any two vertices v, w there exists a walk which begins at v and ends at w.



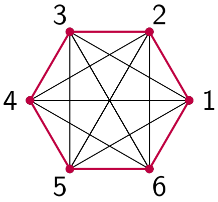


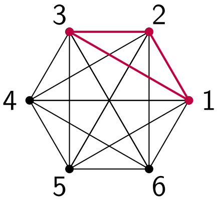
In the six vertex-graph on the left, there exists a walk from any vertex to any other vertex. In the graph on the right, there is no walk from 1 to 4. The graph on the right is not connected.

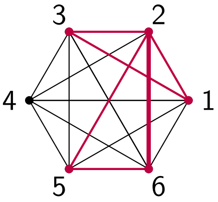
Is a single isolated vertex a connected graph?

Cycles/circuits

A circuit/cycle is a closed walk with at least one edge and no repeated edges.

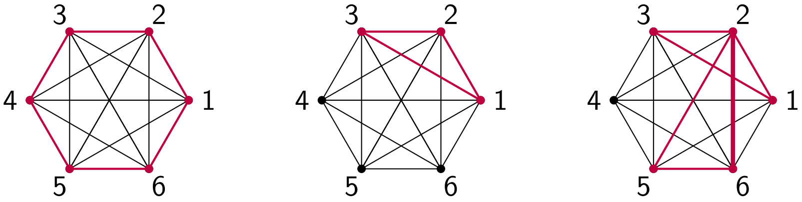




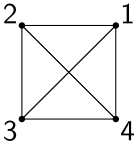


Cycles/circuits

A circuit/cycle is a closed walk with at least one edge and no repeated edges.

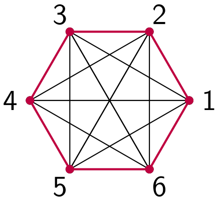


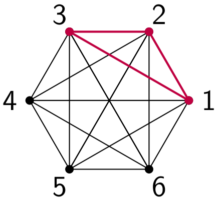
How many possible cycles are there in a clique on 4 vertices?

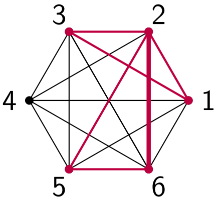


Cycles/circuits

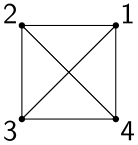
A circuit/cycle is a closed walk with at least one edge and no repeated edges.



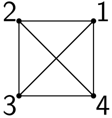




How many possible cycles are there in a clique on 4 vertices?



How many possible cycles are there in a clique on 4 vertices?

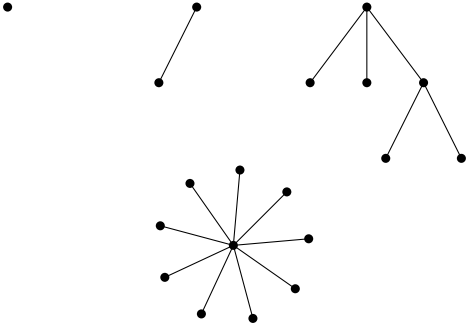


We need either 3 or 4 vertices for a cycle. Any more than that will start reusing edges.Let's count the lenth-3 cycles. Pick a starting point, and pick which vertex we're leaving out. Then we still get to pick one of the two directions around the triangle. So we have $4 \cdot 3 \cdot 2 = 24$ cycles of length 3.

Cycles of lenth 4 either go around the square, or across the diagonal in an hourglass shape. There are 8 ways to go around the square, since we pick the starting point and then the direction. Similarly, there are 8 ways to go around the hourglass-shape made by sequence 2-4-1-3 and 8 ways to go around the same for 2-4-3-1. The total number of cycles is 48.

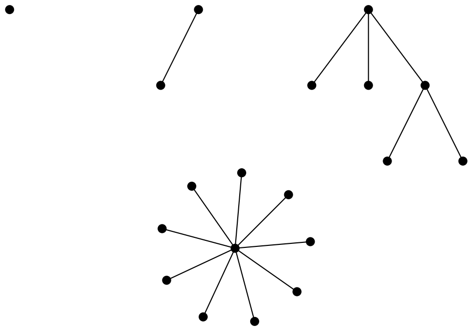
Trees

A tree is a conected graph with no cycles.



Trees

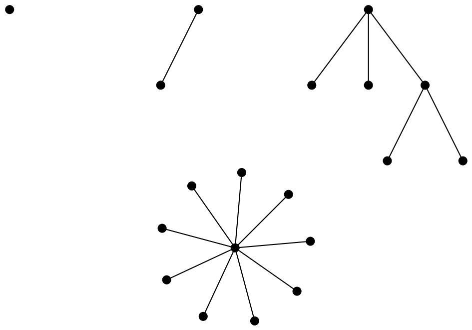
A tree is a conected graph with no cycles.



Each of these is a tree. Together they are a forest.

Trees

A tree is a conected graph with no cycles.



Each of these is a tree. Together they are a forest.

A forest is a graph with no cycles.

Equivalent definitions of a tree.

A simple, undirected graph is a tree if and only if it satisfies these equivalent conditions:

a) T is connected and acyclic.

b) T is connected and has $n-1$ edges.

c) For any two distinct vertices v, w in T, there is a unique path from v to w.

We can use definition $c)$ to show that every tree is bipartite.

Theorem

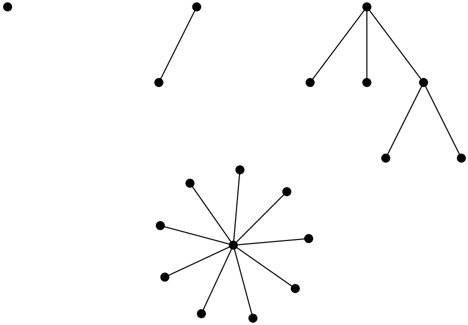
Every tree is a bipartite graph.

Sketch of proof: Pick any vertex in the tree, color it blue. There is a unique path from that vertex to any other one. If it's of odd length, color that vertex red. If it's of even length, color it blue. It is not hard to see that this is a bipartition.

Every tree on n vertices has $n-1$ edges.

Theorem

Every tree on n vertices has $n-1$ edges.



Every tree on n vertices has $n-1$ edges.

Theorem

Every tree on n vertices has $n-1$ edges.

Proof: We will prove this by strong induction.

Every tree on n vertices has $n - 1$ edges.

Theorem

Every tree on n vertices has $n - 1$ edges.

Proof: We will prove this by strong induction.

Base case, $n = 1$, is a single vertex with no edges. $\checkmark$

Every tree on n vertices has $n-1$ edges.

Theorem

Every tree on n vertices has $n-1$ edges.

Proof: We will prove this by strong induction.

Base case, $n = 1$, is a single vertex with no edges. ✓

So suppose that any tree with $n \leq k$ vertices for some $k \geq 1$ has $n-1$ edges. We want to show that for any tree with $k+1$ vertices has k edges.

Every tree on n vertices has $n-1$ edges.

Theorem

Every tree on n vertices has $n-1$ edges.

Proof: We will prove this by strong induction.

Base case, $n = 1$, is a single vertex with no edges. $\checkmark$

So suppose that any tree with $n \leq k$ vertices for some $k \geq 1$ has $n-1$ edges. We want to show that for any tree with $k+1$ vertices has k edges.

So take any tree on $k+1$ vertices. Since $k+1 \geq 2$, there exists at least one edge. What happens if we remove that edge?

Every tree on n vertices has $n-1$ edges.

Theorem

Every tree on n vertices has $n-1$ edges.

Proof: We will prove this by strong induction.

Base case, $n = 1$, is a single vertex with no edges. ✓

So suppose that any tree with $n \leq k$ vertices for some $k \geq 1$ has $n-1$ edges. We want to show that for any tree with $k+1$ vertices has k edges.

So take any tree on $k+1$ vertices. Since $k+1 \geq 2$, there exists at least one edge. If we remove that edge, the number of edges goes down by 1 and the graph gets disconnected into two separate trees.

Theorem

Every tree on n vertices has $n-1$ edges.

Proof: We will prove this by strong induction.

Base case, $n = 1$, is a single vertex with no edges. ✓

So suppose that any tree with $n \leq k$ vertices for some $k \geq 1$ has $n-1$ edges. We want to show that for any tree with $k+1$ vertices has k edges.

So take any tree on $k+1$ vertices. Since $k+1 \geq 2$, there exists at least one edge. If we remove that edge, the number of edges goes down by 1 and the graph gets disconnected into two separate trees, on n_1 and n_2 vertices respectively with $n_1, n_2 < k+1$, $n_1 + n_2 = k+1$.

Every tree on n vertices has $n-1$ edges.

So suppose that any tree with $n \leq k$ vertices for some $k \geq 1$ has $n-1$ edges. We want to show that for any tree with $k+1$ vertices has k edges.

So take any tree on $k+1$ vertices. Since $k+1 \geq 2$, there exists at least one edge. If we remove that edge, the number of edges goes down by 1 and the graph gets disconnected into two separate trees, on n_1 and n_2 vertices respectively with $n_1, n_2 < k+1$, $n_1 + n_2 = k+1$.

By inductive hypothesis, the number of edges in these trees is respectively $n_1 - 1$ and $n_2 - 1$. Together they have $n_1 + n_2 - 2 = k-1$ edges. The original tree had all of those, plus the edge we removed. That's k edges. $\square$

Theorem

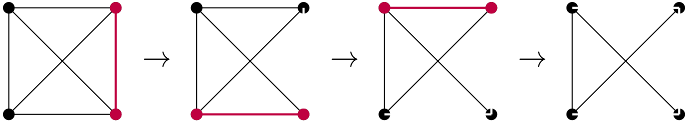
Any connected graph with n vertices and $n-1$ edges is a tree.

Proof: we will prove this by contradiction.

Theorem

Any connected graph with n vertices and $n-1$ edges is a tree.

Proof: we will prove this by contradiction. Suppose that we have a connected graph G with n vertices, $n-1$ edges and there are cycles in the graph. if you remove an edge that is in a cycle, the graph is still connected. So remove an edge from a cycle. If there are cycles left do it again, until there are no cycles left. This graph is still connected.



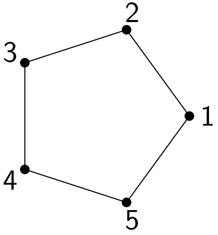
Theorem

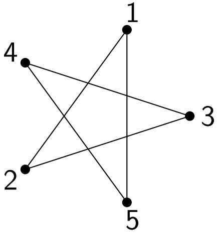
Any connected graph with n vertices and $n-1$ edges is a tree.

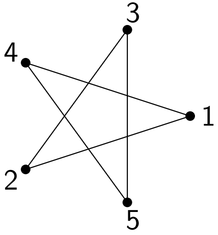
Proof: we will prove this by contradiction. Suppose that we have a connected graph G with n vertices, $n-1$ edges and there are cycles in the graph. if you remove an edge that is in a cycle, the graph is still connected. So remove an edge from a cycle. If there are cycles left do it again, until there are no cycles left. This graph is still connected.

Now the graph is a tree, and so has $n-1$ edges. But we just removed some nonzero number of edges, so we must have started with more than $n-1$. $\square$

Graph Isomorphism

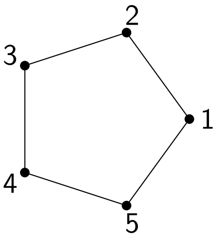


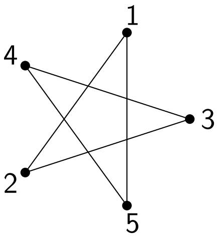


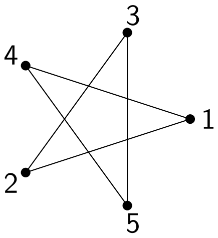


The first two are the same graph. The third on eis a different graph - for example vertex 1 is adjacent to 4, unlike in the other two.

Graph Isomorphism







The first two are the same graph. The third on eis a different graph - for example vertex 1 is adjacent to 4, unlike in the other two.

But this whole lecture we didn't use labels for vertices... and if the vertices were unlabeled, the second and third graph would be identical. We need some way to talk about graphs being the same *up to a relabelling of vertices.*

Graph Isomorphism

Two things are isomorphic if they have the same structure.

Graph Isomorphism

Two things are isomorphic if they have the same structure.

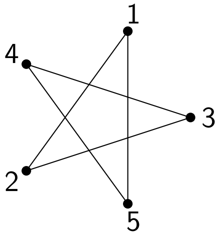
Two graphs $G = (V, E)$ and $H = (W, F)$ are isomorphic if there exists a one to one function $f : V \to W$ that preserves adjacency. That is, $x, y \in V$ are adjacent in G if and only if $f(x), f(y) \in W$ are adjacent in H.

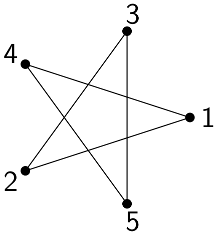
Graph Isomorphism

Two things are isomorphic if they have the same structure.

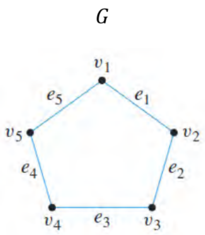
Two graphs $G = (V, E)$ and $G' = (V', E')$ are isomorphic if there exists a one to one function $f : V \to V'$ that preserves adjacency. That is, $x, y \in V$ are adjacent in G if and only if $f(x), f(y) \in V'$ are adjacent in G'.

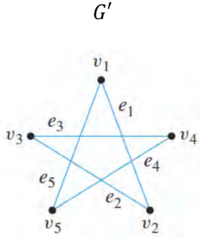
Example:

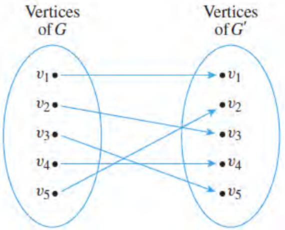


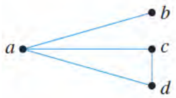


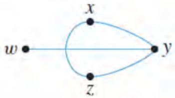
$$\begin{aligned} f(1) &= 3 \\ f(2) &= 2 \\ f(3) &= 1 \\ f(4) &= 4 \\ f(5) &= 5 \end{aligned}$$

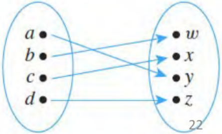

G
v1
e5
e1
v5
v2
e4
e2
v4
e3
v3


G′
v1
e1
e3
v3
v4
e4
e5
e2
v5
v2


Vertices of G
Vertices of G′
v1
v2
v3
v4
v5
v1
v2
v3
v4
v5


a
b
c
d


x
w
y
z


a
b
c
d
w
x
y
z
22

What is Common Between Isomorphic Graphs?

*Properties that are **(necessarily)** preserved if two graphs are isomorphic are called **invariants** (under isomorphism).*

Theorem: The following properties are (some examples of) invariants under isomorphism

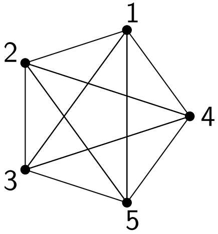
- The number of vertices is n.
- The number of edges is m.
- The number of degree d vertices is k.
- The number of simple circuits of length l is t.
- The number of circuits of length l is t.
- The graph is connected (to be defined shortly).
- The graph has an Eulerian circuit (to be defined shortly).
- The graph has a Hamiltonian circuit (to be defined shortly).

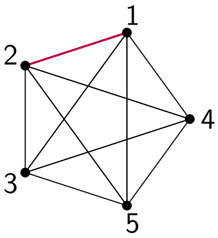
Observation: Preservation of properties above is **NOT sufficient** to prove that graphs are isomorphic.

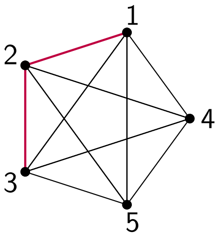
	Repeated Edge?	Repeated Vertex?	Starts and Ends at Same Point?	Must Contain at Least One Edge?
Walk	allowed	allowed	allowed	no
Trail	no	allowed	allowed	no
Path	no	no	no	no
Closed walk	allowed	allowed	yes	no
Circuit	no	allowed	yes	yes
Simple circuit	no	first and last only	yes	yes

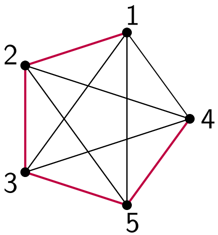
We have two special kinds of graph traversals.

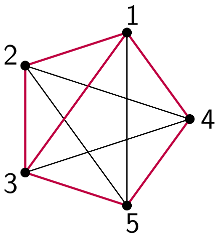
- Eulerian circuit of a graph G is a circuit (no repeated edges) that contains every vertex and every edge of G.
- Hamiltonian circuit of a graph G is a simple circuit that contains every vertex of G exactly once (except that it ends in the same vertex it sarted from.)

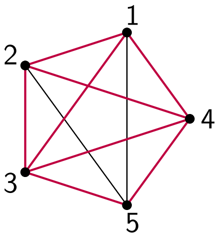

1
2
4
3
5

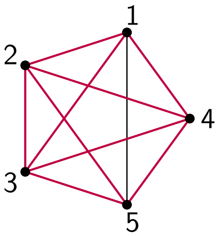

1
2
3
4
5

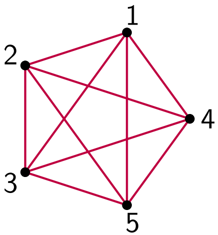

1
2
4
3
5

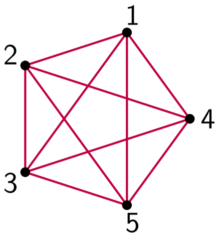

1
2
4
3
5


1
2
4
3
5

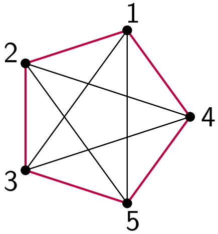

1
2
4
3
5


1
2
4
3
5


1
2
4
3
5

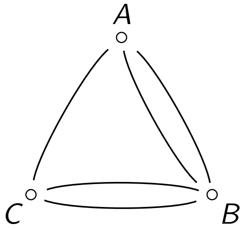


Eulerian circuit covers all edges. Hamiltonian circuit hits every vertex.



Adjacency matrices again

An adjacency matrix is an integer matrix that encodes the graph. Rows correspond to vertices, and columns correspond to vertices. i, j-entry (ith row and jth column) is the integer representing how many edges connect vertices i and j.



$$\begin{array}{c|ccc} & \text{A} & \text{B} & \text{C} \\ \hline \text{A} & 0 & 2 & 1 \\ \text{B} & 2 & 0 & 2 \\ \text{C} & 1 & 2 & 0 \end{array}$$

We're no longer assuming the graph is simple.

Adjacency matrices again

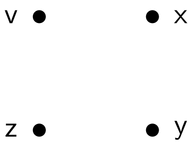
Find an undirected graph that has the following adjacency matrix.

$$\begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \end{bmatrix}$$

Adjacency matrices again

Find an undirected graph that has the following adjacency matrix.

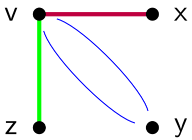
$$\begin{array}{c} v \\ x \\ y \\ z \end{array} \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \end{bmatrix}$$



Adjacency matrices again

Find an undirected graph that has the following adjacency matrix.

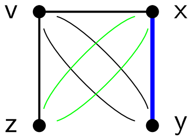
$$\begin{array}{c} v \\ x \\ y \\ z \end{array} \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \end{bmatrix}$$



Adjacency matrices again

Find an undirected graph that has the following adjacency matrix.

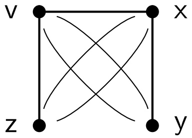
$$\begin{array}{c} v \\ x \\ y \\ z \end{array} \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \end{bmatrix}$$



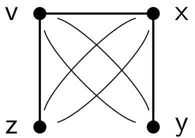
Adjacency matrices again

Find an undirected graph that has the following adjacency matrix.

$$\begin{array}{c} v \\ x \\ y \\ z \end{array} \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \end{bmatrix}$$



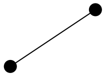
$$\begin{array}{c} v \\ x \\ y \\ z \end{array} \begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \end{bmatrix}$$



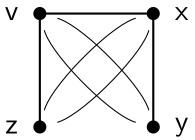
An adjacency matrix lists the number of edges from one vertex to another. Equivalently, the number of length-1 walks from one vertex to another. For example, there are 2 ways we can get from v to y in one step - pick one of the two edges.

Call this matrix A. The entries of A count the 1-step walks, the entries of A^2 count the 2-step walks. Let's look at an easy example:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^2 = \begin{bmatrix} & \\ & \end{bmatrix}$$



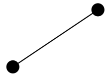
$$\begin{array}{c} v \\ x \\ y \\ z \end{array}\begin{bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \end{bmatrix}$$



An adjacency matrix lists the number of edges from one vertex to another. Equivalently, the number of length-1 walks from one vertex to another. For example, there are 2 ways we can get from v to y in one step - pick one of the two edges.

Call this matrix A. The entries of A count the 1-step walks, the entries of A^2 count the 2-step walks. Let's look at an easy example:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



The entries of the adjacency matrix A count the 1-step walks, the entries of A^2 count the 2-step walks.

The entries of A^2 are governed by this formula:

$$(A^2)_{i,j} = \sum_k A_{i,k} A_{k,j}$$

$A_{i,k}$ is the number of edges from i to k.
$A_{k,j}$ is the number of edges from k to j.
So $A_{i,k} A_{k,j}$ is the number of 2-step walks from i through k to j. If we add those over all possible k, we get the total number of 2-step walks from i to j.

Similarly, A^3 counts the number of 3-step walks from one vertex to another, and so on for any A^n.