Module 12.1: Subgraphs and Counting Paths

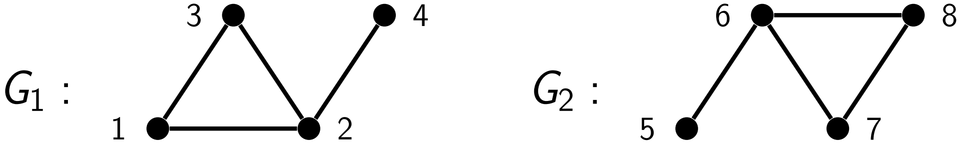
MCIT Online - CIT592 - Professor Val Tannen

Lecture Notes

Graph isomorphism

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **isomorphic**, notation $G_1 \simeq G_2$, when there is a **bijection** $\beta : V_1 \to V_2$ such that for any $u_1, v_1 \in V_1$ we have $u_1{-}v_1 \in E_1$ **iff** $\beta(u_1){-}\beta(v_1) \in E_2$.

Example:



$G_1 \simeq G_2$ by the bijection $4 \mapsto 5,\ 2 \mapsto 6,\ 1 \mapsto 7,\ 3 \mapsto 8$.

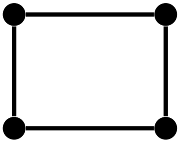
$1 \mapsto 8,\ 3 \mapsto 7$ also works! Note that the bijection must preserve node degree.

More graph isomorphism examples

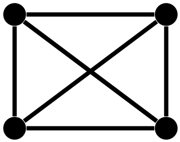
Proposition. Any two complete graphs are isomorphic iff they have the same number of vertices. The same holds for path, cycle, and edgeless graphs. Any two $m \times n$ grids are isomorphic, as well as isomorphic to any $n \times m$ grids!

(Proving isomorphism formally is tedious. Visual intuition is much better! Below is just a reminder of what some of these graphs look like.)

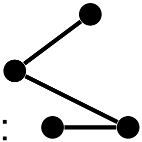
C_4 :



K_4 :



P_4 :



ACTIVITY : Special graphs revisited

We can now give a precise definition: a **path graph on n vertices** is a graph isomorphic to P_n. Hence, a **path graph of length ℓ** is a graph isomorphic to $P_{\ell+1}$.

But who is P_n? Recall that we did not name its vertices when we drew it. We didn't because it does not matter, as you see! You can choose any names you want, for instance: $P_3 = \{\{1, 2, 3\}, \{1–2, 2–3\}\}$. Regardless of the vertex names chosen, the class of path graphs is the same. In many problems the names of the vertices do not matter, and can talk about **the** graph P_n, as we already did.

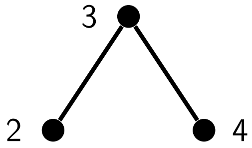
The same definition and terminology will be used for, and the same discussion applies to **cycle**, **complete**, and **edgeless graphs on n vertices**, and **grid graphs on $m \times n$ vertices**..

Subgraphs

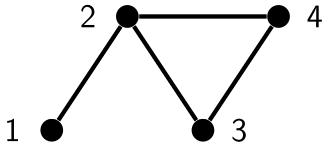
A graph $G_1 = (V_1, E_1)$ is a **subgraph** of the graph $G_2 = (V_2, E_2)$ when $V_1 \subseteq V_2$ and $E_1 \subseteq E_2$. (Beware: not all pairs of such subsets form graphs!)

Example:

G_1 :



G_2 :



Here, G_1 is a subgraph of G_2. Another subgraph of G_2 has vertices $2, 3, 4$ and edges 2–3, 3–4, 4–2.

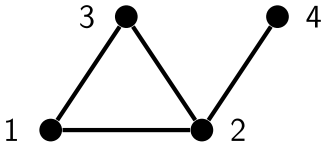
Yet another subgraph of G_2 has vertices $1, 2, 3, 4$ and edges 2–1, 2–3, 2–4.

Induced subgraphs

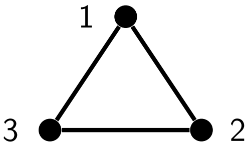
If $G = (V, E)$ is a graph and $V' \subseteq V$ is a set consisting of some of G's nodes, the subgraph of G **induced** by V' is the graph $G' = (V', E')$ where E' consists of all the edges of G whose endpoints are both in V'.

Example:

G :



G' :



G' is the subgraph of G **induced** by the subset of vertices $\{1, 2, 3\}$.

The subgraph of G induced by $\{2, 3, 4\}$ is path graph of length 2.

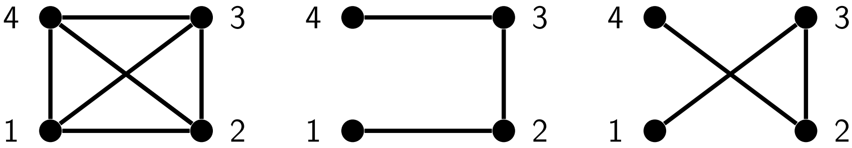
The subgraph of G induced by $\{1, 3, 4\}$ is disconnected.

Counting paths

When we count **paths of length** n in a graph G, we count in fact the subgraphs of G that **are** path graphs on $n+1$ vertices.

Problem. Consider the complete graph on nodes $\{1,2,3,4\}$. Find two **different** path subgraphs that have the **same** set of nodes.

Answer.

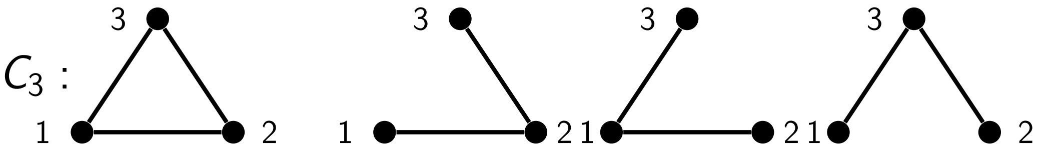


The paths 1–2–3–4 and 1–3–2–4 have the same set of nodes but different sets of edges. They correspond to the two distinct path subgraphs shown.

Counting paths in cycles I

Problem. How many paths of length 2 are there in C_3?

Answer.



Any path of length 2 has two edges. In C_2 any two edges form a path of length 2 since they must have an endpoint in common. There are $\binom{3}{2} = 3$ combinations of two edges therefore 3 path subgraphs (see figure).

Counting paths in cycles II

Problem. How many paths of length 2 are there in C_n $(n \geq 3)$?

Answer. For $n \geq 4$ there are combinations of two edges that do not form a path subgraph. We must count in a different way.

Observe that we can define (in any graph) a function from path subgraphs of length 2 to vertices by associating to each such path subgraph its middle vertex. In general, this function is neither injective nor surjective but in C_n it is both, hence it's a bijection.

By the bijection rule there are as many path subgraphs of length 2 as there are vertices. This gives us the answer: n. This answer is valid for $n = 3$ and it's the same as on the previous slide.

Quiz

How many paths (of any length) are there in P_4?

(A) 6

(B) 15

(C) 10

Answer

(A) 6

Incorrect. Have you counted the paths of length 0?

(B) 15

Incorrect. Recall that P_4 has length 3.

(C) 10

Correct. There are 4 paths of length 0, 3 paths of length 1, 2 paths of length 2, and 1 path of length 3. In total there are $4 + 3 + 2 + 1 = 10$ paths.

Module 12.2: Cycles

MCIT Online - CIT592 - Professor Val Tannen

LECTURE NOTES

Closed walks and cycles

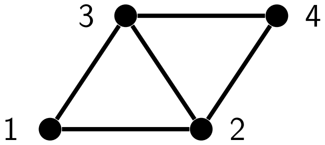
A **closed walk** is a walk in which the first and the last vertex are the same.

A **cycle** is a closed walk **of length at least 3** in which all nodes are pairwise distinct, except for the last and the first.

The **length** of the cycle is the length of the closed walk.

Examples of closed walks (cw's) and cycles:

G :



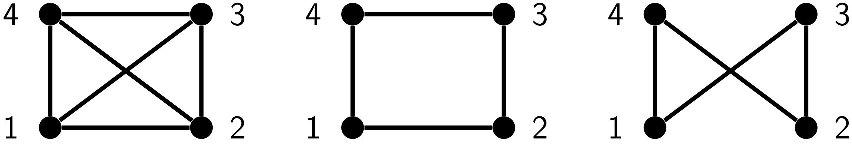
- 1–2–3–4–2–1 cw, **not** cycle
- 2–3–4–2 cycle, length 3
- 2 cw, **not** cycle
- 2–4–2 cw, **not** cycle
- 1–2–4–3–1 cycle, length 4

Counting cycles

When we count **cycles of length** n in a graph G, we count in fact the subgraphs of G that **are** cycle graphs on n vertices.

Problem. Consider the complete graph on nodes $\{1, 2, 3, 4\}$. Find two **different** cycle subgraphs that have the **same** set of nodes.

Answer.



The cycles 1–2–3–4–1 and 1–3–2–4–1 have the same set of nodes but different sets of edges. They correspond to the two distinct cycle subgraphs shown.

Counting cycles in K_4

Problem. How many cycles are there in K_4?

Answer. The subgraph induced by any three vertices is a cycle subgraph.
Therefore, the number of cycles of length 3 is $\binom{4}{3} = 4$.

We count the cycles of length 4 from the perspective of one node, say node 1.
Any cycle going through 1 uses 2 of the 3 edges incident to node 1. Once we choose these 2 edges the rest of the cycle of length 4 is determined. Indeed, there is only one more node to go through and this can be done in only one way.
Therefore, the number of cycles of length 4 is $\binom{3}{2} = 3$.

And the total number of cycles in K_4 is $4 + 3 = 7$.

Quiz I

How many cycles of length 4 are there in K_5?

(A) 15

(B) 12

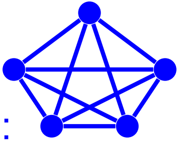
(A) 15

Correct. Construct a cycle subgraph of length 4 in two steps:

Step 1: Choose 4 out of 5 vertices in $\binom{5}{4} = 5$ ways

Step 2: Construct a cycle of length 4 on the chosen 4 vertices. Recall that earlier in this video we showed that this can be done in 3 ways.

In total there are $5 \cdot 3 = 15$ cycles of length 4. You can make sure that this counting is correct by checking K_5 manually

K_5 : 

(B) 12

Incorrect.

Quiz II

What is the total number of cycles in K_5?

(A) 35

(B) 36

(C) 37

Answer

(A) 35

Incorrect.

(B) 36

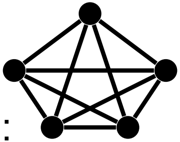
Incorrect.

(C) 37

Correct. We have $\binom{5}{3} = 10$ cycles of length 3 and from the previous quiz we have 15 cycles of length 4. For length 5 begin similarly to length 4 in K_4: from the perspective of node 1. Choose 2 of the 4 edges incident to node 1 in $\binom{4}{2} = 6$ ways. Next, let u and v be the other endpoints of the chosen 2 edges. To complete the cycle we need a path of length 3 from u to v that does not go through 1. There are two ways to order the two intermediate nodes on this path. In total there are $6 \cdot 2 = 12$ cycles of length 5. Finally, $10 + 15 + 12 = 37$.

More Information

You can make sure that this counting is correct by checking K_5 manually

K_5 : 

Module 12.3: Forests, Trees, Leaves

MCIT Online - CIT592 - Professor Val Tannen

Lecture Notes

Connected components as graphs

Let $G = (V, E)$ be a graph. We defined **connected components** (cc's) as subsets $C \subseteq V$ and observed that they form a **partition** of V.

Now, we regard a cc C as a graph, namely the subgraph of G **induced** by the set of vertices in C. The cc's also partitions the set of edges E:

Proposition. Every edge $\{u, v\} \in E$ belongs to **exactly one** of the subgraphs induced by the cc's of G.

Proof. Indeed, both u and v must be in the same cc.

Proposition. To count the number of paths of length k in G we add up the number of paths of length k in each of its connected components. Similar for cycles.

Acyclic graphs and trees

A graph in which there are no cycles is called **acyclic**. The cc's of an acyclic graph are also acyclic.

A graph that is both connected and acyclic is called a **tree**.

Consequently, an acyclic graph is also called a **forest** since all its cc's are trees!

Proposition. Let G_1 and G_2 be two **isomorphic** graphs, $G_1 \simeq G_2$. Then:

- G_1 is acyclic iff G_2 is acyclic.
- G_1 is connected iff G_2 is connected.
- G_1 is a tree iff G_2 is a tree.

(A formal proof would be tedious and these facts are intuitively obvious as we think of isomorphism as "copy".)

How many edges in a tree/forest?

Proposition. A tree has one more vertex than edges. That is, if $G = (V, E)$ is a tree then $|E| = |V| - 1$.

Corollary. If $G = (V, E)$ is a forest then $|E| = |V| - |CC|$ where CC is the set of cc's of G.

Proof. Let's say that G has two cc's, (V_1, E_1) and (V_2, E_2). The cc's partition both vertices and edges: $V = V_1 \cup V_2$ and $E = E_1 \cup E_2$.

. $|V| = |V_1| + |V_2|$ $\qquad$ $|E| = |E_1| + |E_2|$

The two cc's must be trees so by the proposition:

. $|E_1| = |V_1| - 1$ $\qquad$ $|E_2| = |V_2| - 1$

Put together the equalities: $|E| = |V| - 2 = |V| - |CC|$ since here we have just two cc's. The general case is similar.

Vertices and edges in a tree

Proposition. Any tree with n vertices has $n-1$ edges.

Proof. By induction on the number n of vertices.

(BC) $n = 1$: An edgeless graph with one vertex is a tree. (It is the only edgeless tree!) It has 0 edges. Check.

(IS) Let k be an arbitrary natural number ≥ 1. Assume (IH) that every tree with k vertices has $k-1$ edges.

Now we want to prove that any tree, G, with $k+1$ vertices has k edges.

To apply the IH we would like to delete a vertex from G. But which one? And how do we insure that after the deletion we still have a tree?

Trees have leaves

In a graph a **leaf** is a node of degree 1.

Lemma. Every tree with edges has at least one leaf (actually, at least two!).

Proof. Consider the set $L \subseteq \mathbb{N}$ of lengths of paths (not walks!) in G. Since paths cannot have length more than the total number of edges, L is finite. The **Well-Ordering Principle** implies that L has a greatest element.

Therefore, there exists a path W of maximum length.

Claim. The endpoints u and v of W are leaves.

Indeed, suppose, toward a contradiction, that u is not a leaf. Then it has degree ≥ 2. Thus W can be extended by one more edge. The other end of that edge is not in W since there are no cycles. Thus W does not have maximum length. Contradiction.

Remove a leaf

We considered an arbitrary $k \geq 1$. We assumed (IH) that every tree with k vertices has $k - 1$ edges. Then we took an arbitrary tree, G, with $k + 1$ vertices and wanted to show it has k edges.

Since $k + 1 \geq 2$, G has edges. By the lemma it has at least one leaf. Let G' be the graph obtained by removing this leaf. Nicely:

Lemma. Removing a leaf from a tree with edges leaves again a tree.

Proof. The resulting graph is still acyclic. It is also still connected because the only paths affected have the removed leaf as an endpoint.

Vertices and edges in a tree (finale)

We considered an arbitrary $k \geq 1$. We assumed (IH) that every tree with k vertices has $k-1$ edges. Then we took an arbitrary tree, G, with $k+1$ vertices and wanted to show it has k edges. We removed a leaf from G obtaining G' and showed that G' is still a tree.

G' has $k+1-1=k$ nodes, so we can apply the IH.
By IH, G' has $k-1$ edges.

When we removed the leaf from G we removed exactly one edge, because the degree of the leaf is 1.

Hence G has one more edge that G', i.e., it has $k-1+1=k$ edges. Done.

Quiz

I have a tree and in my tree all nodes have the same degree! Then my tree has:

(A) Two edges.

(B) Strictly more than two edges.

(C) Strictly less than two edges.

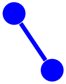
Answer

(A) Two edges.
Incorrect. Any tree with two edges will have a node with degree 1 (i.e. a leaf) and a node with degree equal to 2.

(B) Strictly more than two edges.
Incorrect. Any tree with more than two edges will have a node with degree equal to 1 (i.e. a leaf) and a node of degree at least 2.

(C) Strictly less than two edges.
Correct. The tree consists of two nodes connected by one edge, as shown below:

T :



Module 12.4: Properties of Trees

MCIT Online - CIT592 - Professor Val Tannen

Lecture Notes

Every tree is minimally connected

Proposition. Removing **any** edge in a tree disconnects it.

Proof. (Warm-up: This is true even if we remove the only edge incident to a leaf. It leaves a graph with 2 cc's one of which is edgeless with one vertex.)

Erase an edge, now "one more vertex than edges" fails so the graph is not a tree anymore.

But erasing an edge does not create cycles so the resulting graph is still acyclic.

The only way it can fail to be a tree is if it is disconnected.

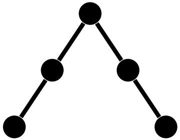
ACTIVITY: Removing edges from a tree

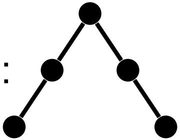
Consider a tree with 5 nodes. Remove 2 edges.

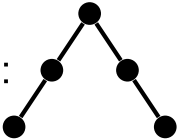
Question. How many cc's will the resulting graph have?

Question. Why must at least one of the resulting cc's have at least 2 nodes?

Answer. TIKZ TREE; TIKZ TWO DIFFERENT WAYS OF REMOVING 2 EDGES

T :


T_1 :


T_2 :


Every tree is maximally acyclic

Proposition. Adding an edge between **any** two non-adjacent vertices in a tree creates a cycle.

Proof. (Warm-up: Draw a cycle graph. Remove an edge. The result is a path graph. But every path graph is a tree! Now put back the edge you removed. You have "created" a cycle!)

Now let u and v be non-adjacent vertices in $G = (V, E)$.

We add u–v and obtain $G_{uv} = (V, E \cup \{u\text{–}v\})$.

Since G is connected there is a path from u to v in G. Adding u–v to this path produces a cycle in G_{uv}.

Quiz

We add 2 edges to a tree. Clearly we can create 2 cycles. But can we create 3 cycles?

(A) Yes.

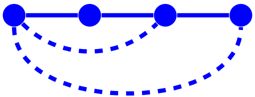
(B) No.

Answer

(A) Yes.

Correct. Consider the following example: we start with P_4 and add one edge to make C_4. We then add a diagonal to make two more cycles, as shown below:

G :



(B) No.

Incorrect. Before you see the answer, try and think the construction on your own. You may want to use pencil and paper and experiment.

Every tree is unique-path connected

Proposition. Any two distinct vertices of a tree are connected by a **unique** path.

Proof. A tree is connected so any two vertices are connected by **at least** one path. We need to prove there is only one such path.

Case 1: two adjacent vertices. u–v is a path. Suppose, toward a contradiction, that there is **another** path from u to v. This path together with u–v forms a cycle, which contradicts the acyclicity of the tree.

Case 2: two non-adjacent vertices. If $G = (V, E)$ is the tree, consider $G_{uv} = (V, E \cup \{u\text{–}v\})$ which has a cycle that includes u–v. Suppose, toward a contradiction, there were **two** distinct paths in G from u to v. Each path together with u–v creates a **distinct** cycle in G_{uv}. See next lemma that contradicts this!

At most one cycle I

Lemma. Adding an edge to an acyclic graph creates **at most** one cycle.

Proof. (Warm-up: why did we say "at most" and not "exactly one"? It is certainly "exactly one" if the graph was a tree, we proved that. But if it is a forest with at least two trees, the added edges can go between nodes in these trees!)

Let u, v be two distinct non-adjacent vertices in an acyclic graph $G = (V, E)$. We add u–v thus producing $G_{uv} = (V, E \cup \{u\text{–}v\})$.

Claim. G_{uv} has at most one cycle.

Suppose, toward a contradiction, that G_{uv} has at least two distinct cycles C_1 and C_2. Since G_{uv} was acyclic u–v must belong to both C_1 and C_2. Since C_1 and C_2 are distinct one of them must contain an edge that is not in the other one. On the next slide we will derive a contradiction from this.

At most one cycle II

Proof (continued). We had u, v distinct non-adjacent vertices in an acyclic graph $G = (V, E)$. We added u–v thus producing $G_{uv} = (V, E \cup \{u\text{–}v\})$. We claimed that G_{uv} has at most one cycle.

We assumed, toward a contradiction, that G_{uv} has two distinct cycles C_1 and C_2. Both go through u–v and one of them must contain an edge that is not in the other one. Let e be that edge.

Deleting u–v from C_1 gives us a path from u to v in G. Deleting u–v from C_2 gives us a path from v to u in G. Concatenating these two gives us a closed walk from u to u that traverses e exactly once. Next, we will see another lemma that says that such a walk must contain a cycle, which contradicts the acyclicity of G. And that will end the proof.

Cycle in a closed walk

Lemma. Any closed walk of non-zero length that traverses at least one of its edges **exactly once** contains a **cycle**.

We omit the (longish) proof but provide some visual intuition through examples.

The closed walk of length 4 $u – v – w – v – u$ does not contain any cycles because it traverses each of its edges twice.

The closed walk of length 5 $u – v – w – z – v – u$ traverses three of its edges, namely $v – w$, $w – z$ and $z – v$ exactly once and indeed it contains a cycle: $v – w – z – v$.

Unique path connectivity

Proposition. A graph such that any two distinct vertices are connected by a unique path must be a tree.

Proof. The graph is clearly connected. It remains to show that it is also acyclic.

Suppose, toward a contradiction, that the graph has a cycle. Let u and v be two distinct nodes in this cycle. The edges and the rest of the vertices of the cycle yield two **distinct** paths from u to v, contradiction.