CS 1010 Discrete Structures Lecture 14: Graphs

Maria Francis

February 18, 2021

Graphs

- Informally, a graph is a bunch of dots and lines where the lines connect some pairs of dots.
- The dots are called nodes (or vertices) and the lines are called edges.
- Problems in almost every conceivable discipline can be solved using graph models.
  - The objects represent items of interest such as programs, people, cities, or web pages.
  - we place an edge between a pair of nodes if they are related in a certain way.
  - For example, an edge between a pair of people might indicate that they like (or, in alternate scenarios, that they don't like) each other.
  - An edge between a pair of courses might indicate that one needs to be taken before the other.
- Check out Rosen textbook for numerous applications.

Graphs

- A graph is represented as $G = (V, E)$ where V - a nonempty set of vertices or nodes and E a set of edges.
- Each edge has either one or two vertices associated with it, called its endpoints. An edge is said to connect its endpoints.
- The set of vertices V of a graph G may be infinite.
- A graph with an infinite vertex set or an infinite number of edges is called an infinite graph.
- A graph with a finite vertex set and a finite edge set is called a finite graph. Here, we will usually consider only finite graphs.

Types of Graphs

- A graph in which each edge connects two different vertices and where no two edges connect the same pair of vertices is called a simple graph. So no loops also!
- Graphs that may have multiple edges connecting the same vertices are called multigraphs.
- When there are m different edges associated to the same unordered pair of vertices $\{u, v\}$, $\{u, v\}$ is an edge of multiplicity m.
- Edges that connect a vertex to itself - loops.
- Graphs that may include loops, and possibly multiple edges connecting the same pair of vertices or a vertex to itself, are sometimes called pseudographs.
- So far the graphs have been undirected graphs.

Directed Graphs

- A directed graph (or digraph) (V, E) consists of a nonempty set of vertices V and a set of directed edges (or arcs) E.
- Each directed edge is associated with an ordered pair of vertices.
  - The directed edge associated with the ordered pair (u, v) is said to start at u and end at v.
- When a directed graph has no loops and has no multiple directed edges, it is called a simple directed graph.
- Directed graphs with multiple directed edges from a vertex to a second (possibly same) vertex are called directed multigraphs.
- When there are m directed edges, each associated to an ordered pair of vertices (u, v), we say that (u, v) is an edge of multiplicity m.

Graphs

Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

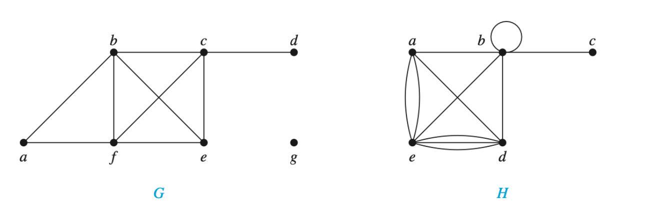
Graph terminology

- Two vertices u and v in an undirected graph G are called adjacent (or neighbors) in G if u and v are endpoints of an edge e of G.
- Such an edge e is called incident with the vertices u and v and e is said to connect u and v.
- Set of all neighbors of a vertex v of $G = (V, E)$, denoted by $N(v)$, is called the neighborhood of v.
- If A is a subset of V, we denote by $N(A)$ the set of all vertices in G that are adjacent to at least one vertex in A. So, $N(A) = \cup_{v \in A} N(v)$.

Graph terminology

- The degree of a vertex v, $\deg(v)$ in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex.
- A special case is a loop, which adds two to the degree, i.e. a vertex with a loop sees itself as an adjacent vertex from both ends of the edge thus adding two, not one, to the degree.
- A vertex of degree zero is called isolated.
- A vertex is pendant if and only if it has degree one. Consequently, a pendant vertex is adjacent to exactly one other vertex.

Graph terminology



Degree of b in H is 6.

Some Results

- What do we get when we add the degrees of all the vertices of a graph?
  - Each edge contributes two to the sum of the degrees of the vertices because an edge is incident with exactly two (possibly equal) vertices.
  - This means that the sum of the degrees of the vertices is twice the number of edges.
  - This gives us the handshaking lemma or handshaking theorem - the handshake involves two hands just like an edge having two endpoints.

Theorem

Let $G = (V, E)$ be an undirected graph with m edges. Then,

$$2m = \sum_{v \in V} deg(v).$$

Note that this applies even if multiple edges and loops are present.

Some Results

- An undirected graph has an even numbers of vertices of odd degree.
- Let V_1 and V_2 be the set of vertices of even degree and odd degree and let there be m edges.

$$2m = \sum_{v \in V} deg(v) = \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v).$$

- Because $deg(v)$ is even for $v \in V_1$, the first term in the RHS of the last equality is even.
- The sum of the two terms on the RHS of the last equality is even, because this sum is $2m$. Hence, the second term in the sum is also even.
- Because all the terms in this sum are odd, there must be an even number of such terms.

Some Results

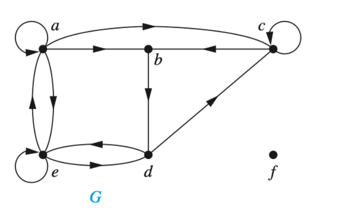
- When (u, v) is an edge of the directed graph G, u is said to be adjacent to v and v is said to be adjacent from u.
- u is called the initial vertex of (u, v), and v is called the terminal or end vertex of (u, v). The initial vertex and terminal vertex of a loop are the same.
- In a directed graph, the in-degree of a vertex v, $deg^-(v)$, is the number of edges with v as their terminal vertex.
- The out-degree of v, $deg^+(v)$, is the number of edges with v as their initial vertex. Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex.

Theorem

Let $G = (V, E)$ be a graph with directed edges. Then

$$\sum_{v \in V} deg^-(v) = \sum_{v \in V} deg^+(v) = |E|.$$

Graph terminology

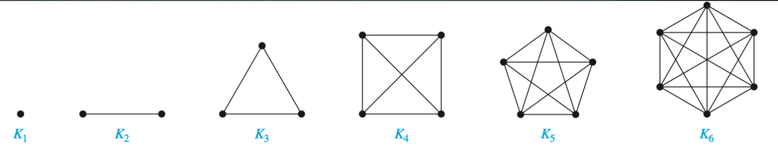


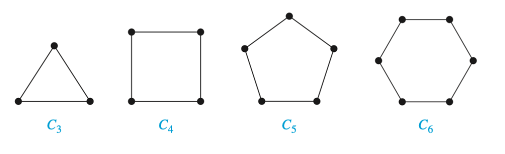
- In-degrees in G are at $a = 2$, at $b = 2$, at $c = 3$, $d = 2$, $e = 3$ and $f = 0$.
- Out-degrees are $a = 4$, at $b = 1$, at $c = 2$, $d = 2$, $e = 3$ and $f = 0$.

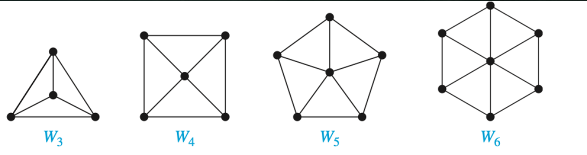
Some special simple graphs

- Complete Graph : A complete graph on n vertices, denoted by K_n, is a simple graph that contains exactly one edge between each pair of distinct vertices.
- The complete graph of n vertices K_n has $n(n-1)/2$ edges. Exercise!
- Cycle $C_n, n \geq 3$ consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}$, $\{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}$ and $\{v_n, v_1\}$.
- Wheels : Add an additional vertex to a cycle C_n, $n \geq 3$ and connect this new vertex to each of the n vertices in C_n by new edges.

Some special simple graphs







Bipartite Graphs

- Let a graph have the property that its vertex set can be divided into two disjoint subsets such that each edge connects a vertex in one of these subsets to a vertex in the other subset.
- For eg: the graph representing marriages between men and women in a group, where each person is represented by a vertex and a marriage is represented by an edge.
- A simple graph G is called bipartite if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2).
- When this condition holds, we call the pair (V_1, V_2) a bipartition of the vertex set V of G.

Bipartite Graphs

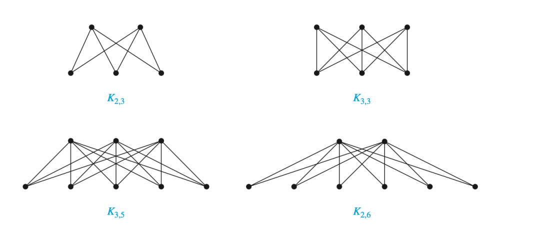
- C_6 is bipartite and K_3 is not bipartite.
- If we divide the vertex set of K_3 into two disjoint sets, one of the two sets must contain two vertices.
- If the graph were bipartite, these two vertices could not be connected by an edge, but in K_3 each vertex is connected to every other vertex by an edge.
- Can there be bipartite graphs that are complete graphs as well? Bipartite graphs cannot contain a triangle - thats a hint!
- A simple graph is bipartite if and only if it is possible to assign one of two different colors to each vertex of the graph so that no two adjacent vertices are assigned the same color.
- This means 2-colorable, chromatic number is ≤ 2.

Proof of 2-colorable

- ($\Leftarrow$): Let $G = (V, E)$ is a bipartite simple graph. Then $V = V_1 \cup V_2$ where V_1 and V_2 are disjoint sets and every edge in E connects a vertex in V_1 and a vertex in V_2.
- If we assign one color to each vertex in V_1 and a second color to each vertex in V_2 then no two adjacent vertices are assigned the same colour.
- ($\Rightarrow$): Let us consider any graph s.t.it is possible to assign colors to the vertices of the graph using two colours so that no two adjacent vertices are assigned the same color.
- Let V_1 be the set of vertices assigned one color and V_2 assigned the other color. Then, $V_1 \cap V_2 = \emptyset$ and $V = V_1 \cup V2$.
- Also, every edge connects a vertex in V_1 and a vertex in V_2 because no two adjacent vertices are either both in V_1 or both in V_2. Thus, G is bipartite.

$K_{m,n}$

- A complete bipartite graph $K_{m,n}$ has its vertex set partitioned into two subsets of size m and n vertices with an edge between two vertices iff one vertex is in the first subset and the other vertex is in the second subset.



- The number of edges in $K_{m,n}$ is mn.- Exercise!

$K_{m,n}$

- P.T. for a bipartite graph G on n vertices the number of edges in G is at most $\frac{n^2}{4}$.
- In a bipartite graph n vertices can be partitioned into two subsets of size i and $(n-i)$ $0 \leq i \leq n$.
- You can only have edges between vertices of different subsets, so you have a maximum of $i(n-i)$ edges if every member of one subset is connected to every member of the other subset.
- $f(i) = i(n-i)$, $0 \leq i \leq n$ is maximized by $i = \frac{n}{2}$ which leads to $n/2 \cdot n/2 = n^2/4$ being the maximum number of edges.

Bipartite Graphs and Matchings

- Bipartite graphs can be used to model applications that involve matching the elements of one set to elements of another. Most common examples - job assignments and marriage problem.
- Suppose that there are m employees in a group and n different jobs that need to be done, where $m \geq n$.
- Each employee is trained to do one or more of these n jobs.
- We would like to assign an employee to each job.
- We represent each employee and each job by a vertex. We include an edge from each employee to all jobs that the employee has been trained to do.
- We partition vertices into two disjoint sets, the set of employees and the set of jobs, and each edge connects an employee to a job.
- Consequently, this graph is bipartite, where the bipartition is (E, J) where E is the set of employees and J is the set of jobs.

Matchings

- Finding an assignment of jobs to employees can be thought of as finding a matching in the graph model, where a matching M in a simple graph $G = (V, E)$ is a subset of the set E of edges of the graph such that no two edges are incident with the same vertex.
- A matching M in a simple graph $G = (V, E)$ is a subset of the set E of edges of the graph such that no two edges are incident with the same vertex.
- A matching is a subset of edges such that if $\{s, t\}$ and $\{u, v\}$ are distinct edges of the matching, then $s, t, u,$ and v are distinct.
- A vertex that is the endpoint of an edge of a matching M is said to be matched in M; otherwise it is said to be unmatched.
- A maximum matching is a matching with the largest number of edges.

Complete Matchings

- We say that a matching M in a bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) is a complete matching from V_1 to V_2 if every vertex in V_1 is the endpoint of an edge in the matching or equivalently if $|M| = |V_1|$.
- Perfect matching or 1-factor – A perfect matching is a matching involving all the vertices of a graph not necessarily bipartite graphs.
- A perfect matching exhausts all of the vertices, so a bipartite graph that has a perfect matching must have the same number of vertices in each part – a complete matching from one part into the other.
- A perfect matching is therefore a matching containing $n/2$ edges (the largest possible), meaning perfect matchings are only possible on graphs with an even number of vertices.

Hall's Marriage Problem

Theorem (Hall's Marriage Theorem (Philip Hall (1935)))
The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof:

- 'Only if': Assume that there is a complete matching M from V_1 to V_2.
- Let $A \subseteq V_1$ for every vertex $v \in A$, there is an $e \in M$ connecting v to a vertex in V_2.
- This implies there are as at least as many vertices in V_2 that are neighbours of vertices in V_1 as there are vertices in V_1, therefore $|N(A)| \geq |A|$.

Hall's Marriage Problem

- 'If': Assume that if $|N(A)| \geq |A|$ for all $A \subseteq V_1$ then there is a complete matching M from V_1 to V_2.
- Basis step: If $|V_1| = 1$, then V_1 contains a single vertex v_0.
- $|N(\{v_0\})| \geq |\{v_0\}| = 1 \Rightarrow$ there is ≥ 1 edge connecting v_0 and $w_0 \in V_2$ – a complete matching from V_1 to V_2.
- Inductive hypothesis :
  - Let $k \in \mathbb{Z}_{\geq 0}$, $G = (V, E)$, bipartite graph with bipartition (V_1, V_2).
  - Let $|V_1| = j \leq k$, if $|N(A)| \geq |A|, \forall A \subseteq V_1$ then there is a complete matching M from V_1 to V_2.

Hall's Marriage Problem

- Let $H = (W, F)$ is a bipartite graph with bipartition (W_1, W_2) and $|W_1| = k + 1$. We need to consider only the following two cases.
- Case(i) : For all integers j, s.t. $1 \leq j \leq k$, the vertices in every set of j elements from W_1 are adjacent to at least $j + 1$ elements of W_2.
- Case (ii) : For some j with $1 \leq j \leq k$ there is a subset $W_1{}'$ of j vertices such that there are exactly j neighbours of these vertices in W_2.

Why not consider a subset of W_1 of $k + 1$ elements?
Why not consider a subset of W_1 of size j with less than j neighbours?

Hall's Marriage Problem

Case (i): For all integers j with $1 \leq j \leq k$, the vertices in every subset of j elements from W_1 are adjacent to at least $j+1$ elements of W_2.

- Select a vertex $v \in W_1$ and an element $w \in N(\{v\})$ (Must exist as per our assumption that $|N(\{v\})| \geq |\{v\}| = 1$)
- We delete v and w and all edges incident to them from H.
- This produces a bipartite graph H' with bipartition $(W_1 \setminus \{v\}, W_2 \setminus \{w\})$.
- $|W \setminus \{v\}| = k$ from inductive hypothesis we have a complete matching from $W_1 \setminus \{v\}$ to $W_2 \setminus \{w\}$.
- Adding the edge from v to w – a complete matching from W_1 to W_2.

Hall's Marriage Problem

Case (ii): Suppose that for some j with $1 \leq j \leq k$, there is a subset $W_1{}'$ of j vertices such that there are exactly j neighbors of these vertices in W_2.

- Let $W_2{}'$ be the set of these neighbors.
- By I.H.– there is a complete matching from $W_1{}'$ to $W_2{}'$.
- Remove these $2j$ vertices from W_1 and W_2 and all incident edges – we get a bipartite graph K with bipartition $(W_1 \setminus W_1{}', W_2 \setminus W_2{}')$.

Hall's Marriage Problem

- Claim: K satisfies the condition $|N(A)| \geq |A|$ for all subsets A of $W_1 \setminus {W_1}'$.
- If not,there would be a subset of size t of $W_1 \setminus {W_1}'$ where $1 \leq t \leq k+1-j$ with fewer than t vertices of $W_2 \setminus W_2'$ as neighbours.
- But then the set of $j+t$ vertices of W_1 with these t vertices together with the j vertices we removed from W_1 has fewer than $j+t$ neighbors in W_2, contradicting the hypothesis that $|N(A)| \geq |A|$ for all $A \subseteq W_1$.
- From I.H, the graph K has a complete matching. Combining this complete matching with the complete matching from ${W_1}'$ to ${W_2}'$ we get a complete matching from W_1 to W_2.

Matchings

- Read up on a constructive proof for Hall's Marriage problem.
- A graph is said to be regular if every node has the same degree.
- Exercise - Every regular bipartite graph has a perfect matching.

Stable Marriage/ Matching problem /SMP

- The problem of finding a stable matching between two equally sized sets of elements given an ordering of preferences.
- When is a matching not stable?
  - There is an element A of the first set which prefers some element B of the second set over the element it was matched .
  - B also prefers A over the element to which B is already matched.
- Given n men and n women, where each person has ranked all members of the opposite sex in order of preference, marry them s.t. there are no two people of opposite sex who would both rather have each other than their current partners – the set of marriages is deemed stable.
- The Gale-Shapley algorithm– $O(n^2)$ algorithm, n is number of men or women. Ex - Read up on this!

Representing Simple Graphs

- Adjacency lists – specify the vertices that are adjacent to each vertex of the graph.

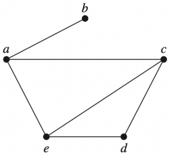


FIGURE 1 **A Simple Graph.**

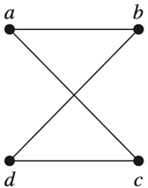
TABLE 1 **An Adjacency List for a Simple Graph.**

Vertex	*Adjacent Vertices*
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

Representing Simple Graphs

- Adjacency matrix A (or A_G) of $G = (V, E)$, w.r.t. the listing of the vertices $v_1, \ldots, v_n$, is the $n \times n$ zero-one matrix with 1 as its (i, j)th entry when v_i and v_j are adjacent, and 0 as its (i, j)th entry when they are not adjacent.
- Its adjacency matrix is $A = [a_{ij}]$, then

$$\begin{aligned} a_{ij} &= 1 \text{ if } \{v_i, v_j\} \text{is an edge of } G \\ &= 0 \text{ otherwise }. \end{aligned}$$



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

Adjacency matrices

- An adjacency matrix of a graph depends on the ordering chosen for the vertices – there may be as many as $n!$ different adjacency matrices for a graph with n vertices!
- For a simple graph it is symmetric, i.e. $a_{ij} = a_{ji}$.
- Since a simple graphs has no loops, each entry a_{ii} is 0.
- Adjacency matrices for undirected graphs multigraphs - a loop at v_i is rep by a 1 at the (i, i)th position.
- For multiple edges connecting the same pair of vertices v_i and v_j , (or multiple loops), the adjacency matrix is no longer a zero-one matrix, the (i, j)th entry equals the number of edges that are associated to $\{v_i, v_j\}$.
- All undirected graphs, including multigraphs and pseudographs, have symmetric adjacency matrices.

Adjacency matrices for directed graphs

- The matrix for a directed graph $G = (V, E)$ has a 1 in its (i, j)th position if there is an edge from v_i to v_j, where $v_1, v_2, \ldots, v_n$ is an arbitrary listing of the vertices of the directed graph.
- If $A = [a_{ij}]$ is the adjacency matrix for the directed graph with respect to this listing of the vertices, then

$$\begin{aligned} a_{ij} &= 1 \text{ if } (v_i, v_j) \text{is an edge of } G \\ &= 0 \text{ otherwise .} \end{aligned}$$

- The adjacency matrix for a directed graph does not have to be symmetric.
- Adjacency matrices can also be used to represent directed multigraphs – again not a zero-one matrix then!

Trade-Offs between Adjacency matrices and Adjacency lists

- For a simple graph contains relatively few edges, that is, sparse, adjacency lists are better.
- For eg, if each vertex has degree not exceeding c, a constant much smaller than n, then each adjacency list contains c or fewer vertices. Hence, there are no more than cn items in all these adjacency lists.
- But the adjacency matrix for the graph has n^2 entries.
- But it is a sparse matrix for which there are special techniques for representing and computing.

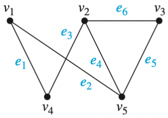
Trade-Offs between Adjacency matrices and Adjacency lists

- For a dense simple graph, (it contains more than half of all possible edges) then we prefer an adjacency matrix over an adjacency list. Consider the complexity of determining whether the possible edge $\{v_i, v_j\}$ is present.
- In an adjacency matrix, its just a look up of (i, j)th entry in the matrix and one comparison with zero or one.
- In case of adjacency lists, we need to search the list of vertices adjacent to either v_i or v_j to determine whether this edge is present. This can require $\Theta(|V|)$ comparisons when many edges are present.

Incidence Matrix

- Another common way to represent graphs is to use incidence matrices.
- Let $G = (V, E)$ be an undirected graph.
- Suppose that $v_1, v_2, \ldots, v_n$ are the vertices and $e_1, e_2, \ldots, e_m$ are the edges of G.
- Then the incidence matrix w.r.t. this ordering of V and E is the $n \times m$ matrix $M = [m_{ij}]$, where

$$\begin{aligned} m_{ij} &= 1 \text{ when edge } e_j \text{ is incident with } v_i \\ &= 0 \text{ otherwise.} \end{aligned}$$



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}.$$

Subgraphs

Sometimes we work with only part of the graph. The smaller graph is called a subgraph .

Definition

A subgraph of a graph $G = (V, E)$ is a graph $H = (W, F)$ where $W \subseteq V$, where $W \subseteq V$ and $F \subseteq E$. A subgraph H of G is a proper subgraph of G if $H \neq G$.

Definition

Let $G = (V, E)$ be a simple graph. The subgraph induced by a subset W of the vertex set V is the graph (W, F), where the edge set F contains an edge in E if and only if both endpoints of this edge are in W.

Every subgraph of a bipartite graph is —–.

Exercise: Every simple graph has a bipartite subgraph with at least $|E|/2$ edges.

Edge Contractions

- Some cases when we remove an edge from a graph, we do not want to retain the endpoints of this edge as separate vertices in the resulting subgraph – we perform an edge contraction .
- We remove an edge e with endpoints u and v and merge u and w into a new single vertex w, and for each edge with u or v as an endpoint replaces the edge with one with w as endpoint in place of u or v and with the same second endpoint.
- For when we have a contraction of the edge e with endpoints u and v what happens to $G = (V, E)$?
- We have a new graph $G' = (V', E')$ (not a subgraph of G), where $V' = V \setminus \{u, v\} \cup \{w\}$ and E' contains the edges in E which do not have either u or v as endpoints and an edge connecting w to every neighbor of either u or v.

Removing Vertices from a Graph & Union of Graphs

- When we remove a vertex v and all edges incident to it from $G = (V, E)$, we produce a subgraph, denoted by $G - v$.
- $G - v = (V \setminus v, E^{'})$, where $E^{'}$ is the set of edges of G not incident to v.
- The union of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with the vertext set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. The union of G_1 and G_2 is denoted by $G_1 \cup G_2$.

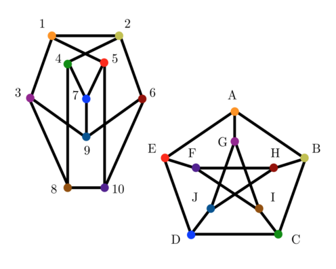
Isomorphism of Graphs

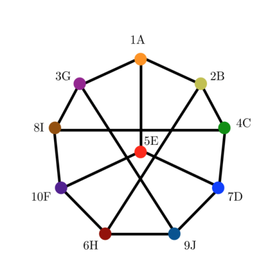
- We often need to know whether if two graphs are essentially the same graph/structure if we ignore the identities of their vertices.
- It has applications in chemistry, for example where the same formula may represent different compounds.

Definition

The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a one- to-one and onto function f from V_1 to V_2 s.t. a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1. Such a function f is called an isomorphism.

Isomorphism of Graphs





Determining if two graphs are isomorphic

- Not that easy to determine if two simple graphs are isomorphic.
- There are $n!$ possible one-to-one correspondences between the vertex sets of two simple graphs with n vertices. If n is large, testing each such correspondence to see whether it preserves adjacency and nonadjacency is impractical!
- Sometimes it is not hard – we can show that two graphs are not isomorphic if we can find a property only one of the two graphs has, but that is preserved by isomorphism.
- Graph Invariant – a property preserved by isomorphism of graphs.
  - Must have the same number of vertices.
  - Must have the same number of edges.
  - Must have the same degrees of the vertices.

Determining if two graphs are isomorphic

- The adjacency matrix of G is the same as the adjacency matrix of H , when rows and columns are labeled to correspond to the images under a bijective/isomorphism f of the vertices in G.
- Note that if f turned out not to be an isomorphism, we would not have established that G and H are not isomorphic, because another correspondence of the vertices in G and H may be an isomorphism.
- The best algorithms known for determining whether two graphs are isomorphic have exponential worst-case time complexity (in the number of vertices of the graphs).
- Linear average-case time complexity algorithms are known that solve this problem.
- One of the few NP problems not known to be either in P or NP-complete.

Moving from *a* to *b* in a graph

Path – a sequence of edges that begins at a vertex of a graph and travels from vertex to vertex along edges of the graph.

Definition

Let $n \in \mathbb{Z}_{\geq 0}$ and G an undirected graph. A path of length n from u to v in G is a sequence of n edges $e_1, \ldots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices such that e_i has for $i = 1, \ldots, n$, the endpoints x_{i-1} and x_i .

- For simple graph, a path is given by its vertex sequence as it uniquely determines the path.
- The path is a circuit if it begins and ends at the same vertex, that is, if $u = v$, and has length greater than zero.
- A path or circuit is simple if it does not contain the same edge more than once.

Moving from a to b in a (directed)graph

Definition

Let $n \in \mathbb{Z}_{\geq 0}$ and G an directed graph. A path of length n from u to v in G is a sequence of n edges $e_1, \ldots, e_n$ of G such that e_i is associated with (x_0, x_1), e_2 with (x_1, x_2) etc, where $x_0 = u$ and $x_n = v$. For a simple directed graph, we denote the path as a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices.

- A path of length greater than zero that begins and ends at the same vertex is called a circuit or cycle.
- A path or circuit is called simple if it does not contain the same edge more than once.
- Walk is also used instead of path, closed walk for circuit and trail for simple path.
- Important examples of paths – Erdős number, the length of the shortest path between a person and Paul Erdős.

Path and Cycles

- A path P_n of length $n-1$ is bipartite.
- A cycle C_n of length n is bipartite iff n is even.
- A graph is bipartite iff it contains no odd cycles.
  - Proof: If bipartite with (V_1, V_2) as bipartition, then every step of a path takes you from V_1 to V_2 or vice-versa.
  - Therefore to reach where you started you need odd steps.
  - Conversely, every cycle of G is even. For a connected component, fix a vertex in it : v_0.
  - For each vertex v in the same component color it red : if its shortest path from v_0 is even, else blue.
  - Do the same for all components.
  - Verify : G has an edge with same colours only if it had an odd cycle.
- Collect the characterizations of bipartite graphs!

Connectivity

- Interested in questions related to : Is there a path between two points in a computer network so as to sent messages through intermediate computers?
- In an undirected graph, connected if there is a path between every pair of distinct vertices of the graph.

Theorem

There is a simple path between every pair of distinct vertices of a connected undirected graph.

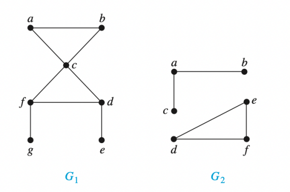
- Let u and v be two distinct vertices of the connected undirected graph $G = (V, E)$.
- G is connected,there is atleast one path between u and v.
- Let $x_0 = u, x_1, \ldots, x_n = v$ be the vertex sequence of a path of least length.

Connectivity

- Claim : This path of least length is simple. If not, then $x_i = x_j$ for some i and j with $0 \leq i \lneq j$.
- But then there is a path from u to v of shorter length – $x_0, x_1, \ldots, x_{i-1}, x_j, \ldots, x_n$.

Connected Components

- A connected component of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G.
- i.e. a maximal connected subgraph of G.
- A graph G that is not connected has two or more connected components that are disjoint and have G as their union.



Connected Components

- Let $\sim$ be the relation on V s.t $v \sim w$ iff v and w are connected by a path. Then $\sim$ is an equivalence relation on V. (Prove!)
- Let V' be an equivalence class of the relation $\sim$ on V. The subgraph induced by V' is called a connected component of the graph.

Vertex Connectivity

- Vertex Cut : A subset $V^{'}$ of V in G if $G - V^{'}$ is disconnected. (defined for noncomplete graphs!)
- Vertex connectivity, $\kappa(G)$: minimum number of vertices in a vertex cut.
- For a complete graph, removing any subset of its vertices and all incident edges still leaves a complete graph.
- We set $\kappa(K_n) = n - 1$, the number of vertices removed to give a graph of one vertex.
- For $0 \leq \kappa(G) \leq n - 1$ for a G with n vertices, $\kappa(G) = 0$ iff G is disconnected or $G = K_1$ and $n - 1$ iff G is complete (Prove!).
- The larger $\kappa(G)$ the more connected G is.
- We say that the graph is k-connected if $\kappa(G) \geq k$. If G is k-connected then G is j-connected for $0 \leq j \leq k$.

Edge Connectivity

- Edge Cut – The minimum number of edges that we can remove to disconnect it.
- Edge connectivity, $\lambda(G)$ – the minimum number of edges in an edge cut of G.
- It is zero if not connected or if G contains a single vertex.
- Ex: $0 \leq \lambda(G) \leq n-1$ and $\lambda(G) = n-1$ iff $G = K_n$
- When $G = (V, E)$ is a noncomplete connected graph with at least three vertices, $\kappa(G) \leq \min_{v \in V} \deg(v)$ and $\lambda(G) \leq \min_{v \in V} \deg(v)$
  - Deleting all the neighbors of a fixed vertex of min degree or edges of that fixed vertex disconnects G.
- Ex: For all graphs $\kappa(G) \leq \lambda(G) \leq \min_{v \in V} \deg(v)$