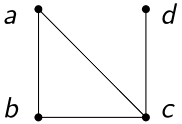
EECS/MATH 1019
Section 10.3:
Representing Graphs
Graph Isomorphism

March 28, 2023

Representing Graphs (subsections 10.3.2–10.3.4)

Some ways to represent a graph (say, in a computer program):

- Adjacency list
- Adjacency matrix
- Incidence matrix



Adjacency list:

Vertex	Adjacent vertices
a	b, c
b	a, c
c	a, b, d
d	c

A matrix is a rectangular array of numbers.

E.g. The matrix $B = \begin{pmatrix} 2 & 0 & -4 & 35 \\ 1 & 88 & 17 & 88 \end{pmatrix}$ has two rows and four columns.

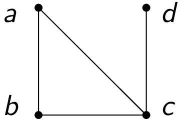
We write b_{ij} to denote the entry of B in the i^{th} row and the j^{th} column. E.g. $b_{11} = 2$, $b_{12} = 0$, $b_{21} = 1$, and so on:

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \end{pmatrix}$$

For a matrix C with m rows and n columns, we have

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{pmatrix}$$

The adjacency matrix of a simple (undirected) graph has one row and one column for each vertex, and the entry in the row of vertex v_i and the column of vertex v_j is 1 if these two vertices are joined by an edge, and 0 otherwise.

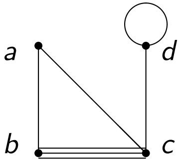


Adjacency matrix: $$\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Important: The order of the rows must be the same as the order of the columns!

If the order is known, then you don't need to label the rows and columns.

For a graph with multiple edges and loops, the adjacency matrix counts the number of edges between pairs of vertices (with each loop counting as one edge):



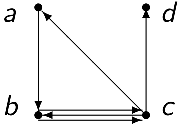
Adjacency matrix:
$$\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 3 & 0 \\ 1 & 3 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

For a graph with no loops: In the row of the adjacency matrix corresponding to a vertex v, what is the sum of the entries of that row?

Answer: The degree of v.

Observe that in the adjacency matrix of a graph (undirected), the entry in row i and column j equals the entry in row j and column i. (In linear algebra, such a matrix is called "symmetric.")

An adjacency matrix of a directed graph is similar, except that for each directed edge (v_i, v_j) that starts at v_i and ends at v_j contributes 1 to the entry in row i and column j (but nothing to row j and column i).

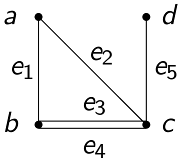


Adjacency matrix: $\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \left(\begin{array}{cccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array} \right) \end{array}$

The sum of the entries in a row equals the out-degree of its vertex.

The *in*-degree of the vertex equals the sum of the entries in the *column* of that vertex.

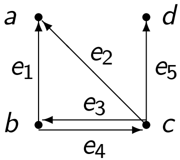
The incidence matrix of an undirected graph has one row for each vertex and one column for each edge. So we need to label edges as well as vertices.



The column corresponding to an edge e_k has a 1 in the rows corresponding to its endpoints, and a 0 in every other row.

Incidence matrix: $$\begin{array}{c c} & \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Remark: We can also define an incidence matrix for directed graphs.



If edge e_k starts at u and ends at v, then the column of e_k has a -1 in the row of u, a $+1$ in the row of v, and 0 in every other row.

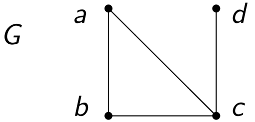
Incidence matrix:

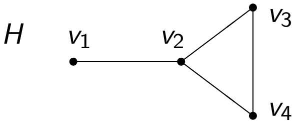
$$\begin{array}{cc} & \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & -1 & 0 \\ 0 & -1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Isomorphisms of Graphs (subsections 10.3.5–10.3.6)

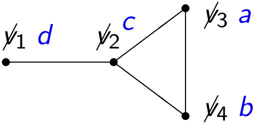
What does it mean to say that two graphs are "essentially the same"? (We shall only discuss undirected simple graphs here.)

For example:





If we identify d with v_1, c with v_2, a with v_3, and b with v_4,



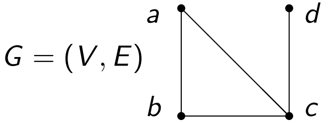
then we get the same graph as G.

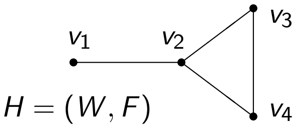
To make this formal, we introduce a function f to "identify" vertices in this way:

$$f(d) = v_1, \quad f(c) = v_2, \quad f(a) = v_3, \quad f(b) = v_4.$$

This f is a one-to-one correspondence from the set of vertices of G to the set of vertices of H. But it is more than that.

Definition: Let $G = (V, E)$ and $H = (W, F)$ be simple graphs.
(i) An isomorphism of G to H is a bijection $f : V \to W$ with the following property: For all vertices t and u in V, t and u are adjacent in G if and only if $f(t)$ and $f(u)$ are adjacent in H.
(ii) The graphs G and H are isomorphic if there exists an isomorphism from G to H.





We claim that our function f is an isomorphism of G to H:

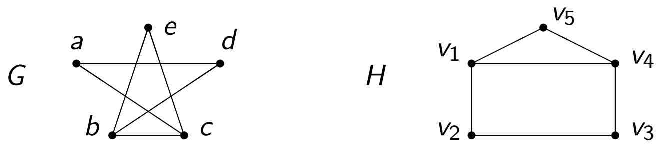
$$f(d) = v_1, \quad f(c) = v_2, \quad f(a) = v_3, \quad f(b) = v_4.$$

This is because $f : V \to W$ is a bijection, and

(t, u)	(a, b)	(a, c)	(a, d)	(b, c)	(b, d)	(c, d)
Adjacent in G?	yes	yes	no	yes	no	yes
$(f(t), f(u))$	(v_3, v_4)	(v_3, v_2)	(v_3, v_1)	(v_4, v_2)	(v_4, v_1)	(v_2, v_1)
Adjacent in H?	yes	yes	no	yes	no	yes

Poll: Are these graphs isomorphic?

(A) Yes (B) No



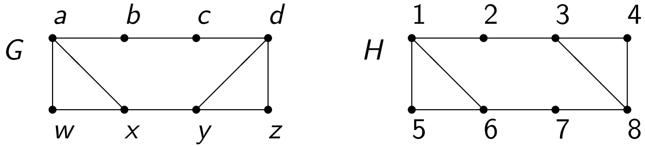
YES. Here is one isomorphism:

$$f(e) = v_5, \quad f(b) = v_1, \quad f(c) = v_4, \quad f(a) = v_3, \quad f(d) = v_2.$$

Some properties: Assume that G and H are isomorphic. Then G and H must have the same number of vertices, and the same number of edges.

Also they must have the same degrees. More precisely, suppose f is an isomorphism from G to H. Let $k \in \mathbb{N}$, and let v be a vertex of degree k in G. Then v is adjacent to exactly k vertices $v_1, \ldots, v_k$ in G, and so $f(v)$ is adjacent to $f(v_1), \ldots, f(v_k)$ in H (and to nothing else). Therefore $f(v)$ has degree k in H.

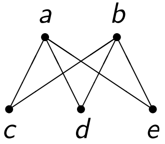
Poll: Are these graphs isomorphic?
(A) Yes (B) No



NO. Here is a proof. Observe that b and c are adjacent in G, and they both have degree 2.
Assume there is an isomorphism f from G to H.
Then $f(b)$ and $f(c)$ must be adjacent in H and they must both have degree 2.
But no two adjacent vertices in H both have degree 2.
This contradiction proves that no isomorphism exists.

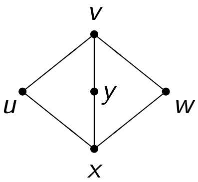
Consider these two graphs and their adjacency matrices:

$K_{2,3}$



$$\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

H



$$\begin{array}{c} u \\ v \\ w \\ x \\ y \end{array} \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Are these two graphs isomorphic? That is, can we re-order (i.e., re-label) the rows (and corresponding columns) of the adjacency matrix of H to make it the same as the adjacency matrix of $K_{2,3}$?

Remark: Deciding whether two large graphs are isomorphic is not easy!

Consider these two problems:

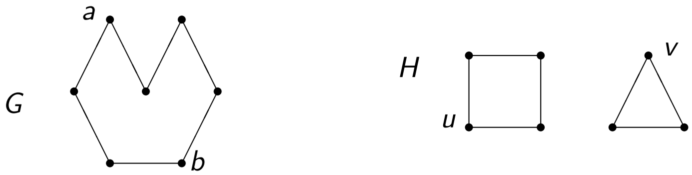
(1) You are given two graphs, each with 1,000 vertices and 50,000 edges. Decide whether they are isomorphic.

(2) You are given a bipartite graph with 1,000 vertices in each part of its bipartition, and 50,000 edges. Decide whether this graph has a complete matching.

Using the best existing algorithms, we expect problem (1) to take much longer than problem (2).

Brief Preview of Section 10.4

Here are two graphs, both of which have 7 vertices, all of degree 2, and 7 edges:



These two graphs seem pretty different. It is not hard to show that G and H are not isomorphic.

We shall introduce some definitions that reflect a major qualitative difference between G and H, namely that G is "connected" while H is "disconnected." Also, the graph H has two "connected components," with vertex u in one component and v in the other.

We say that G contains a "path" from a to b, but H does not contain a path from u to v.

In our next class, we shall give the proper definitions, and show how they can be used.

Next class: Read Section 10.4.

The next Connect assignment is due Sunday April 2.

Please fill out the course evaluations for each of your courses:
http://courseevaluations.yorku.ca/
The course evaluation questions are listed under two separate tabs. The first tab contains the Core Institutional Questions and the second tab has the Course Level Questions.
Please complete both sections of the evaluation form.