Application: Assign classes to professors

Problem description: Math professors at UI are asked to provide an ordered list of classes that they would like to teach in a particular semester.

The goal is to assign classes to these professors which fit their preferences as much as possible.

Vertices: The set of professors union the set of classes.

I.e., each math professor is represented by a vertex and each section of a math class is represented by a vertex.

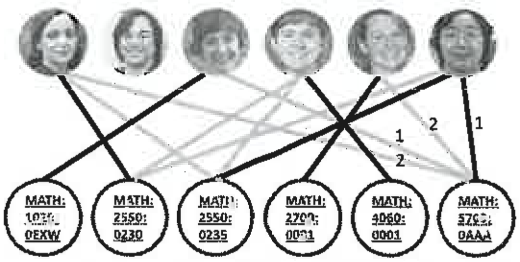
That is a vertex will represent either a math professor or a section of a math class.

Edges: An edge is drawn between a vertex representing a math professor and all sections of a math class if that professor has listed that math class as one of the courses they would like to teach.

7

Application: Assign classes to professors

Example: UI's mathbio group (Spr 2018)



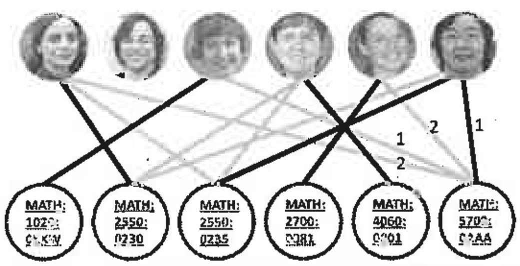
Math professors at UI are asked to provide an ordered list of classes that they would like to teach in a particular semester.

Weighted graph

8

Application: Assign classes to professors

Example: UI's mathbio group (Spr 2018)



Math professors at UI are asked to provide an ordered list of classes that they would like to teach in a particular semester.

Weighted graph

What is the real problem?

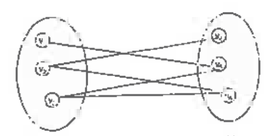
9

Bipartite graphs

- In a simple graph G, if 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 V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2)

  Application example: Representing Relations

  Representation example: $V_1 = \{v_1, v_2, v_3\}$ and $V_2 = \{v_4, v_5, v_6\}$,



10

EX

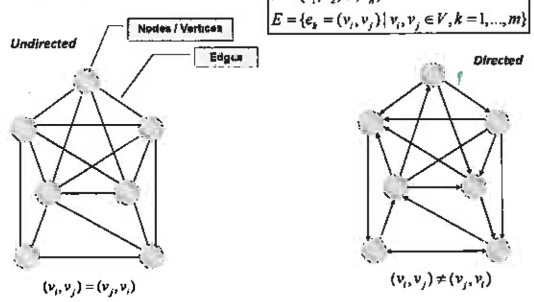
Graphs

Graph with 7 nodes and 16 edges

$G = (V, E)$

$V = \{v_1, v_2, \ldots, v_n\}$

$E = \{e_k = (v_i, v_j) \mid v_i, v_j \in V, k = 1, \ldots, m\}$



11

Definition 2.1: A *graph* [illegible] *vertices* [illegible] *edges* [illegible] *join* [illegible] *end points* [illegible] *adjacent* [illegible] *incident* [illegible]

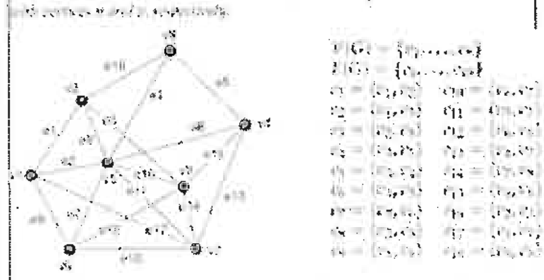


Figure 2.1: An example of a graph with eight vertices and [illegible] edges.

12

Note vocabulary

induced by V*
maximal subgraph G_1 of G st $V(G_1) = V^*$

Definition 2.4: A graph H is a *subgraph* of G if V(H) ⊆ V(G) and E(H) ⊆ E(G) [illegible] When H is a subgraph of G, we write H ⊆ G.

Definition 2.5: Consider a graph G and a subset V* ⊆ V(G). *The subgraph induced by* V* [illegible]

[illegible]

19

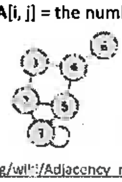
The **complement of a graph** G, denoted as $\overline{G}$ is the graph obtained from G by removing all its edges and joining exactly those vertices that were not adjacent in G.

It should be clear that if we take a graph G and its complement G "together," we obtain a complete graph.

Definition 2.6 [illegible] *The line graph* [illegible]

20

Adjacency matrix: **A[i, j]** = the number of edges joining vertex v_i and v_j.



https://en.wikipedia.org/wiki/Adjacency_matrix

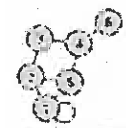
$$\begin{pmatrix} 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

- An adjacency matrix is symmetric, that is for all i, j, A[i, j] = A[j, i]. This property reflects the fact that an edge is represented as an unordered pair of vertices [illegible]
- A graph G is simple if and only if for all i, j, A[i, j] ≤ 1 and A[i, i] = 0. In other words, there can be at most one edge joining vertices v_i and v_j and, in particular, no edge joining a vertex to itself.
- The sum of values in row i is equal to the degree of vertex v_i, that is, [illegible]

21

Incidence matrix:
M[i, j] = the number of times that edge e_j is incident with vertex v_i.

2	1	1	0	0	0	0	0
0	1	0	1	1	0	0	0
0	0	0	0	1	0	1	0
0	0	0	0	0	1	1	1
0	0	1	1	0	1	0	0
0	0	0	0	0	0	0	1



https://en.wikipedia.org/wiki/Adjacency_matrix

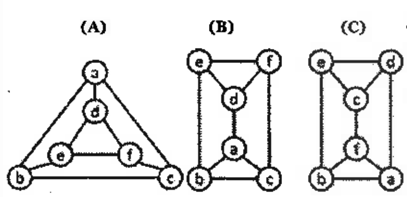
OR

2	1	0	0	0	0	0	1
0	1	1	0	0	0	1	0
0	0	1	1	0	0	0	0
0	0	0	1	1	1	0	0
0	0	0	0	1	0	1	1
0	0	0	0	0	1	0	0

OR ...

22

Definition 2.7: [illegible] *isomorphic* [illegible]



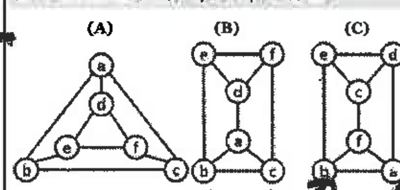
V(A)	V(B)
a	a
b	b
c	c
d	d
e	e
f	f

[illegible]/practical-graph-mining-with-R/

23

Graph Isomorphism

Two graphs G and H are isomorphic (denoted G ≅ H) if there exists a bijection f such that f : V(G) → V(H) such that an edge [illegible] ∈ E(G) if and only if [illegible] ∈ E(H).



V(A)	V(C)
a	f
b	b
c	e
d	c
e	[illegible]
f	d

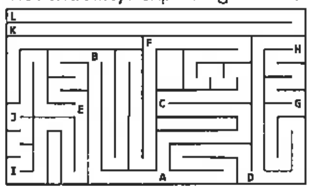
Which graphs are isomorphic?

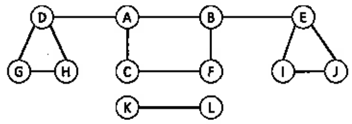
https://www.csc2.ncsu.edu/faculty/[illegible]/practical-graph-mining-with-R/

24

https://www.cs.upc.edu/~jordicf/Teach..g/AP2/pptx/10_Graphs_Connectivity.pptx

Reachability: exploring a maze

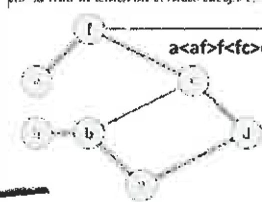




Which vertices of the graph are reachable from a given vertex?

2.3 Connectivity

Definition 2.8: *Consider a graph G. A* $(\mathbf{v_0}, \mathbf{v_k})$*-**walk** in G is an alternating sequence* $[v_0, e_1, v_1, e_2 \ldots v_{k-1}, e_k, v_k]$ *of vertices and edges from G with* $e_i = (v_{i-1}, v_i)$*. In a **closed walk**,* $v_0 = v_k$*. A **trail** is a walk in which all edges are distinct; a **path** is a trail in which also all vertices are distinct. A **cycle** is a closed trail in which all vertices except* v_0 *and* v_k *are distinct.*



a<af>f<fc>c<cd>d<de>e<eb>b<bh>h

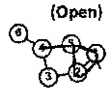
afcdebh

Definition 2.9: *Two distinct vertices u and v in graph G are **connected** if there exists a* (u, v) *− path in G. G is **connected** if all pairs of distinct vertices are connected.*

Walk: Vertices may repeat. Edges may repeat (Open or Closed)

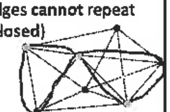
645234523

Trail: Vertices may repeat. Edges **cannot** repeat (Open)

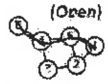


45125

Circuit: Vertices may repeat. Edges **cannot** repeat (Closed)

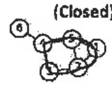


Path: Vertices **cannot** repeat. Edges **cannot** repeat (Open)



6451

Cycle: Vertices **cannot** repeat. Edges **cannot** repeat (Closed)



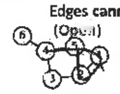
451234

https://en.wikipedia.org/wiki/File:6n-graf.svg

Walk: Vertices may repeat. Edges may repeat (Open or Closed)

645234523

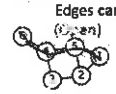
Trail: Vertices may repeat. Edges **cannot** repeat (Open)



45125

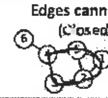
Circuit: Vertices may repeat. Edges **cannot** repeat (Closed)

Path: Vertices **cannot** repeat. Edges **cannot** repeat (Open)



6451

Cycle: Vertices **cannot** repeat. Edges **cannot** repeat (Closed)



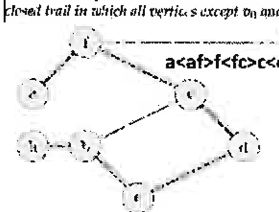
451234

Path ⊂ **Trail** ⊂ **Walk**

Cycle ⊂ **Circuit** ⊂ **Closed Walk**

2.3 Connectivity

Definition 2.8: *Consider a graph G. A* $(\mathbf{v_0}, \mathbf{v_k})$*-**walk** in G is an alternating sequence* $[v_0, e_1, v_1, e_2 \ldots v_{k-1}, e_k, v_k]$ *of vertices and edges from G with* $e_i = (v_{i-1}, v_i)$*. In a **closed walk**,* $v_0 = v_k$*. A **trail** is a walk in which all edges are distinct; a **path** is a trail in which also all vertices are distinct. A **cycle** is a closed trail in which all vertices except* v_0 *and* v_k *are distinct.*

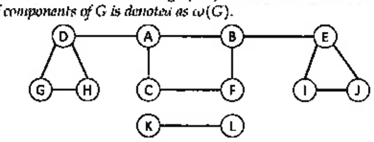


a<af>f<fc>c<cd>d<de>e<eb>b<bh>h

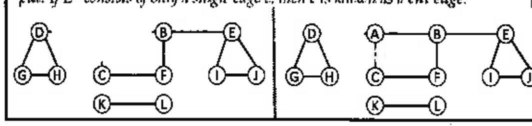
afcdebh

Definition 2.9: *Two distinct vertices u and v in graph G are **connected** if there exists a* (u, v) *− path in G. G is **connected** if all pairs of distinct vertices are connected.*

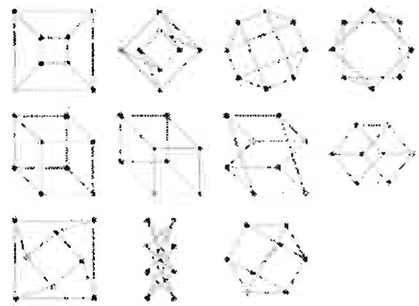
Definition 2.10: *A subgraph H of G is called a **component** of G if H is connected and not contained in a connected subgraph of G with more vertices or edges. The number of components of G is denoted as* $\omega(G)$*.*



Definition 2.11: *For a graph G let* $V^* \subset V(G)$ *and* $E^* \subset E(G)$*.* V^* *is called a **vertex cut** if* $\omega(G - V^*) > \omega(G)$*. If* V^* *consists of a single vertex v, then v is called a **cut vertex**. Likewise, if* $\omega(G - E^*) > \omega(G)$ *then* E^* *is called an **edge cut**. If* E^* *consists of only a single edge e, then e is known as a **cut edge**.*

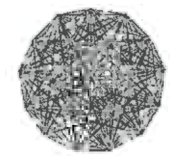


Drawing a graph

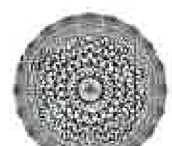


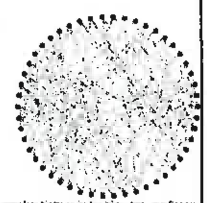
http://mathworld.wolfram.com/...

GraphPlot[Table[1, {20}, {20}]] https://reference.wolfram.com/language/...

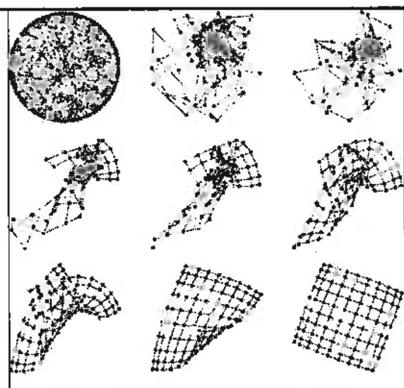


GraphPlot[Table[1, {20}, {20}], Method → "Circ..."]





Graph Theory and Complex Networks by Maarten van Steen



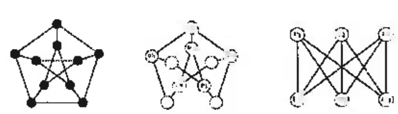
Graph Theory and Complex Networks by Maarten van Steen

Figure 2.17: The evolution of applying a spring embedding to a graph.

What is a planar embedding?

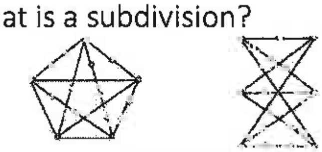
Kuratowski's Theorem (1930)

A graph is planar if and only if it does not contain a subdivision of K_5 or $K_{3,3}$.



Kuratowski Subgraphs

What is a subdivision?



Kuratowski Subgraphs

not on mini-Exam 1

Euler characteristic (simple form):

χ = number of vertices – number of edges + number of faces

Or in short-hand,

$$\chi = |V| - |E| + |F|$$

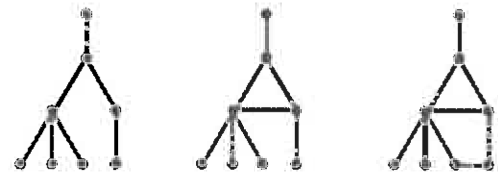
where V = set of vertices
E = set of edges
F = set of faces = set of regions

& the notation |X| = the number of elements in the set X.

For a planar connected graph |V| - |E| + |F| = 2

Defn: A *tree* is a connected graph that does **not** contain a cycle.

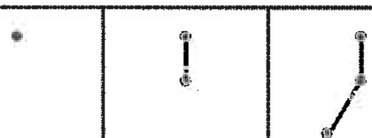
A *forest* is a graph whose components are trees.



$\chi = 8 - 7 + 1 = 2$ $\quad \chi = 8 - 8 + 2 = 2$ $\quad \chi = 8 - 9 + 3 = 2$

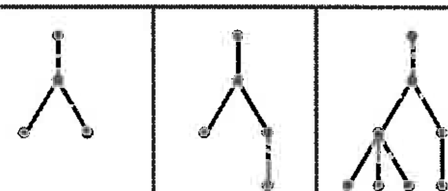
Lemma 2.1: Any tree with *n* vertices has *n-1* edges.

$\chi = |V| - |E| + |F|$



$\chi = 1 - 0 + 1 = 2$ $\quad \chi = 2 - 1 + 1 = 2$ $\quad \chi = 3 - 2 + 1 = 2$

$\chi = |V| - |E| + |F|$



$\chi = 4 - 3 + 1 = 2$ $\quad \chi = 5 - 4 + 1 = 2$ $\quad \chi = 8 - 7 + 1 = 2$

$\chi = |V| - |E| + |F|$

$\chi = 8 - 8 + 2 = 2$ $\quad \chi = 8 - 9 + 3 = 2$

Not a tree.

For the b[illegible]ve of heart, consider graphs drawn on other sur[illegible] such as a torus or K[illegible] bottle. For fun, see [illegible] or www.geometry[illegible].org

Euler's fomula: For a planar connected graph |V| - |E| + |F| = 2 where V = set of vertices, E = set of edges, F = set of faces = set of regions

Defn: A *tree* (or *acyclic graph*) is a connected graph that does **not** contain a cycle.

A *forest* is a graph whose components are trees.

Lemma 2.1: Any tree with *n* vertices has *n-1* edges.

Thm 2.9: For any connected planar graph with $|V| \geq 2$,
$$|E| \leq 3|V| - 6$$

Cor 2.4: K_5 is nonplanar.

Thm 2.10: $K_{3,3}$ is nonplanar.

Cor: A graph is planar if and only if it does not contain a subdivision of K_5 or $K_{3,3}$.