HW 3 (Due Wednesday Feb 6)

Create slide(s) for your 1 minute presentation on a graph theory application. Make sure your slide(s) include

(1) Define the problem

(2) What do the vertices represent

(3) What do the edges represent

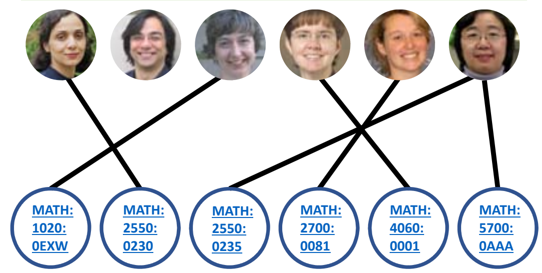
(4) What can graph theory say about your real-life problem? Can you formally state the graph theory problem(s)?

Use large font (best minimum = 24 point, 18 OK)
Figures are helpful. INCLUDE YOUR NAME and affiliation.

1

Application: Assign classes to professors

Isabel Darcy
University of Iowa, Mathematics/AMCS/Informatics
http://homepage.divms.uiowa.edu/~idarcy

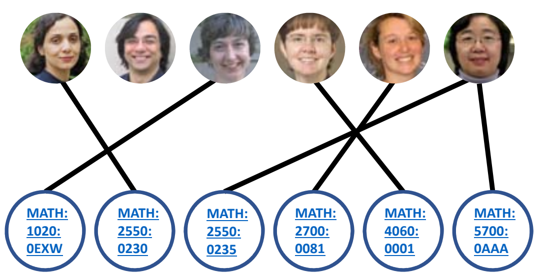


Example: UI's mathbio group (Spr 2018)

2

Application: Assign classes to professors

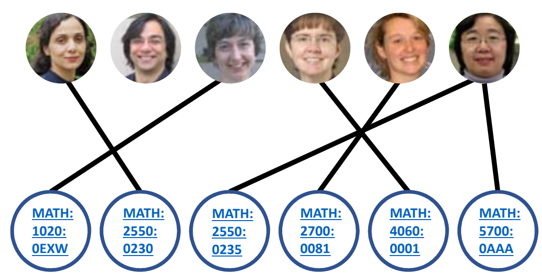
Example: UI's mathbio group (Spr 2018)



3

Application: Assign classes to professors

Example: UI's mathbio group (Spr 2018)

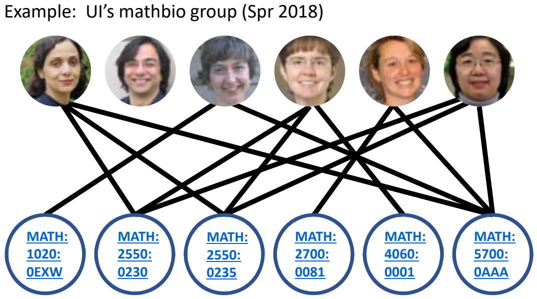


A **vertex** represents either a math professor or a section of a math course

4

Application: Assign classes to professors

Example: UI's mathbio group (Spr 2018)

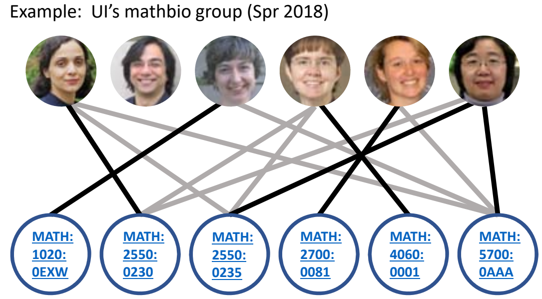


An edge connects a math professor to a section of a math course that professor would like to teach

5

Application: Assign classes to professors

Example: UI's mathbio group (Spr 2018)



Graph theory problem: Select a subset of the edges so that each vertex representing a course section has degree 1 and each vertex representing a professor has degree 0, 1, or 2.

6

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.

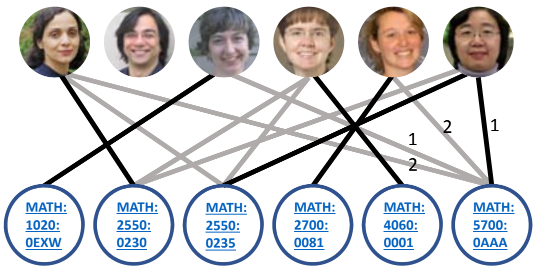
That is a vertex will represent either a math professor or a section of a math class.

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)



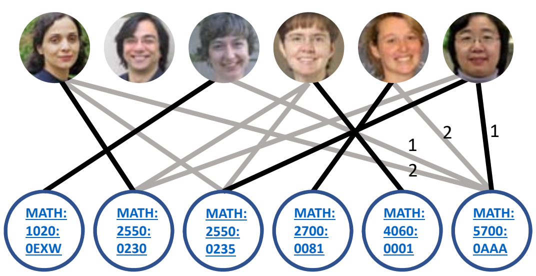
Math professors at UI are asked to provide an ordered list of classes that they would like to teach in a particular semester.

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.

What is the real problem?

Weighted graph

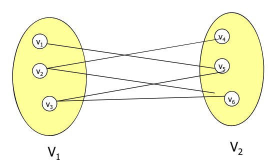
9

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

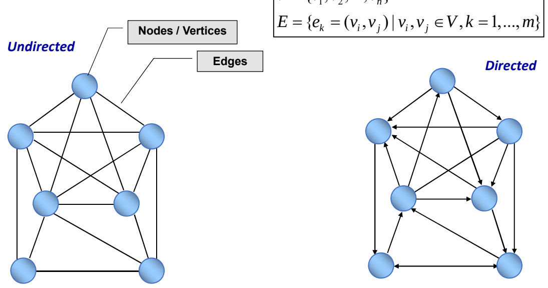
Graphs

Graph with 7 nodes and 16 edges

$G = (V, E)$

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

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

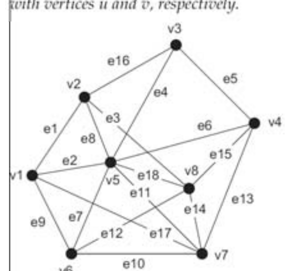


$(v_i, v_j) = (v_j, v_i)$ $\qquad$ $(v_i, v_j) \neq (v_j, v_i)$

11 https://www.csc2.ncsu.edu/faculty/nfsamato/practical-graph-mining-with-R/

11

Definition 2.1: *A **graph** G consists of a collection V of **vertices** and a collection **edges** E, for which we write G = (V, E). Each edge e ∈ E is said to **join** two vertices, which are called its **end points**. If e joins u, v ∈ V, we write e = ⟨u, v⟩. Vertex u and v in this case are said to be **adjacent**. Edge e is said to be **incident** with vertices u and v, respectively.*



$V(G) = \{v_1, \ldots, v_8\}$

$E(G) = \{e_1, \ldots, e_{18}\}$

$e_1 = \langle v_1, v_2 \rangle$ $\quad e_{10} = \langle v_6, v_7 \rangle$

$e_2 = \langle v_1, v_5 \rangle$ $\quad e_{11} = \langle v_5, v_7 \rangle$

$e_3 = \langle v_2, v_8 \rangle$ $\quad e_{12} = \langle v_6, v_8 \rangle$

$e_4 = \langle v_3, v_5 \rangle$ $\quad e_{13} = \langle v_4, v_7 \rangle$

$e_5 = \langle v_3, v_4 \rangle$ $\quad e_{14} = \langle v_7, v_8 \rangle$

$e_6 = \langle v_4, v_5 \rangle$ $\quad e_{15} = \langle v_4, v_8 \rangle$

$e_7 = \langle v_5, v_6 \rangle$ $\quad e_{16} = \langle v_2, v_3 \rangle$

$e_8 = \langle v_2, v_5 \rangle$ $\quad e_{17} = \langle v_1, v_7 \rangle$

$e_9 = \langle v_1, v_6 \rangle$ $\quad e_{18} = \langle v_5, v_8 \rangle$

Figure 2.1: An example of a graph with eight vertices and 18 edges.

12

Definitions – Graph Type

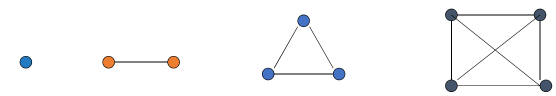
Type	Edges	Multiple Edges Allowed ?	Loops Allowed ?
Simple Graph	undirected	No	No
Multigraph	undirected	Yes	Depends on book (yes for us)
Pseudograph	undirected	Yes	Yes
Directed Graph	directed	No	Yes
Directed Multigraph	directed	Yes	Yes

Modified from https://utdallas.edu/~praba/graph.ppt

13

Simple graphs – special cases

- **Complete graph:** K_n, is the simple graph that contains exactly one edge between each pair of distinct vertices.



K_1 K_2 K_3 K_4

Modified from https://utdallas.edu/~praba/graph.ppt

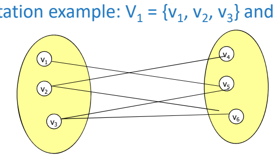
14

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 $V_2 = \{v_4, v_5, v_6\}$,



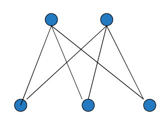
V_1 V_2

Definition 2.14: *A graph G is* **bipartite** *if $V(G)$ can be partitioned into two disjoint subsets V_1 and V_2 such each edge $e \in E(G)$ has one end point in V_1 and the other in V_2: $E(G) \subseteq \{e = \langle u_1, u_2 \rangle | u_1 \in V_1, u_2 \in V_2\}$.*

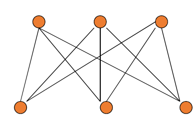
15

Complete Bipartite graphs

- $K_{m,n}$ is the graph that has its vertex set portioned into two subsets of m and n vertices, respectively There is an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.



$K_{2,3}$



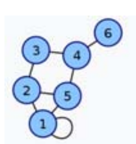
$K_{3,3}$

16

Definition 2.2: *For any graph G and vertex $v \in V(G)$, the* **neighbor set** *$N(v)$ of v is the set of vertices (other than v) adjacent to v, that is*

$$N(v) \stackrel{\text{def}}{=} \{w \in V(G) \mid v \neq w, \exists e \in E(G) : e = \langle u, v \rangle\}$$

N(1) = {2, 5}
N(2) = {1, 3, 5}



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

Definition 2.3: *The number of edges incident with a vertex v is called the* **degree** *of v, denoted as $\delta(v)$. Loops are counted twice.*

Degree of vertex 1 is 4

17

Theorem 2.1: *For all graphs G, the sum of the vertex degrees is twice the number of edges, that is,*

$$\sum_{v \in V(G)} \delta(v) = 2 \cdot |E(G)|$$

Corollary 2.1: *For any graph, the number of vertices with odd degree is even.*

Degree sequence

Listing the vertex degrees of a graph gives us a **degree sequence**. The vertex degrees are usually listed in descending order, in which case we refer to an **ordered degree sequence**.

A sequence is **graphic** iff it is the degree sequence for a simple graph.

If every vertex has the same degree, the graph is called **regular**.

In a **k-regular** graph each vertex has degree k.

Thus its degree sequence is [k, k, …, k]

Theorem 2.2 (Havel-Hakimi): *Consider a list $\mathbf{s} = [d_1, d_2, \ldots, d_n]$ of n numbers in descending order. This list is graphic if and only if $\mathbf{s}^* = [d_1^*, d_2^*, \ldots, d_{n-1}^*]$ of $n-1$ numbers is graphic as well, where*

18

Definition 2.4: *A graph H is a* **subgraph** *of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ such that for all $e \in E(H)$ with $e = \langle u, v \rangle$, we have that $u, v \in V(H)$. When H is a subgraph of G, we write $H \subseteq G$.*

Definition 2.5: *Consider a graph G and a subset $V^* \subseteq V(G)$. The* **subgraph induced by $\mathbf{V^*}$** *has vertex set V^* and edge set E^* defined by*

$$E^* \stackrel{\text{def}}{=} \{e \in E(G) | e = \langle u, v \rangle \text{ with } u, v \in V^*\}$$

Likewise, if $E^* \subseteq E(G)$, the subgraph induced by E^* has edge set E^* and a vertex set V^* defined by

$$V^* \stackrel{\text{def}}{=} \{u, v \in V(G) | \exists e \in E^* : e = \langle u, v \rangle\}$$

The subgraph induced by V^* or E^* is written as $G[V^*]$ or $G[E^*]$, respectively.

19

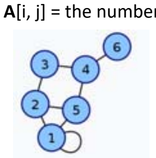
The **complement of a graph** G, denoted as $\overline{G}$ is the graph obtained from G by removing all its edges and joining exactly those vertices that were not adjacent in G.

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

Definition 2.6: *Consider a simple graph $G = (V, E)$. The* **line graph** *of G, denoted as $L(G)$ is constructed from G by representing each edge $e = \langle u, v \rangle$ from E by a vertex v_e in $L(G)$, and joining two vertices v_e and v_{e^*} if and only if edges e and e^* are incident with the same vertex in G.*

20

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



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

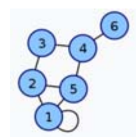
$$\begin{pmatrix} 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

- An adjacency matrix is *symmetric*, that is for all i, j, $\mathbf{A}[i, j] = \mathbf{A}[j, i]$. This property reflects the fact that an edge is represented as an *unordered* pair of vertices $e = \langle v_i, v_j \rangle = \langle v_j, v_i \rangle$.
- A graph G is simple if and only if for all i, j, $\mathbf{A}[i, j] \leq 1$ and $\mathbf{A}[i, i] = 0$. 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, $\delta(v_i) = \sum_{j=1}^{n} \mathbf{A}[i, j]$.

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

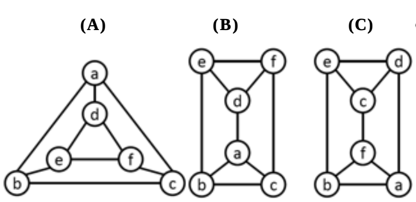
OR

```
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: *Consider two graphs $G = (V, E)$ and $G^* = (V^*, E^*)$. G and G^* are* **isomorphic** *if there exists a one-to-one mapping $\phi : V \to V^*$ such that for every edge $e \in E$ with $e = \langle u, v \rangle$, there is a unique edge $e^* \in E^*$ with $e^* = \langle \phi(u), \phi(v) \rangle$.*



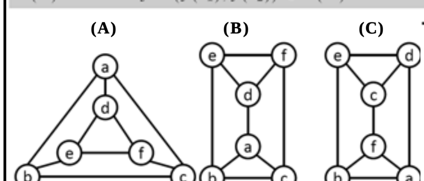
$V(A)$	$V(B')$
a	a
b	b
c	c
d	d
e	e
f	f

23 https://www.csc2.ncsu.edu/faculty/nfsamato/practical-graph-mining-with-R/

23

Graph Isomorphism

Two graphs G and H are **isomorphic** (denoted $G \simeq H$) if there exists a bijection f such that $f : V(G) \to V(H)$ such that an edge $(v_1, v_2) \in E(G)$ if and only if $(f(v_1), f(v_2)) \in E(H)$.



$V(A)$	$V(C)$
a	f
b	b
c	a
d	c
e	e
f	d

Which graphs are isomorphic?

24 https://www.csc2.ncsu.edu/faculty/nfsamato/practical-graph-mining-with-R/

24