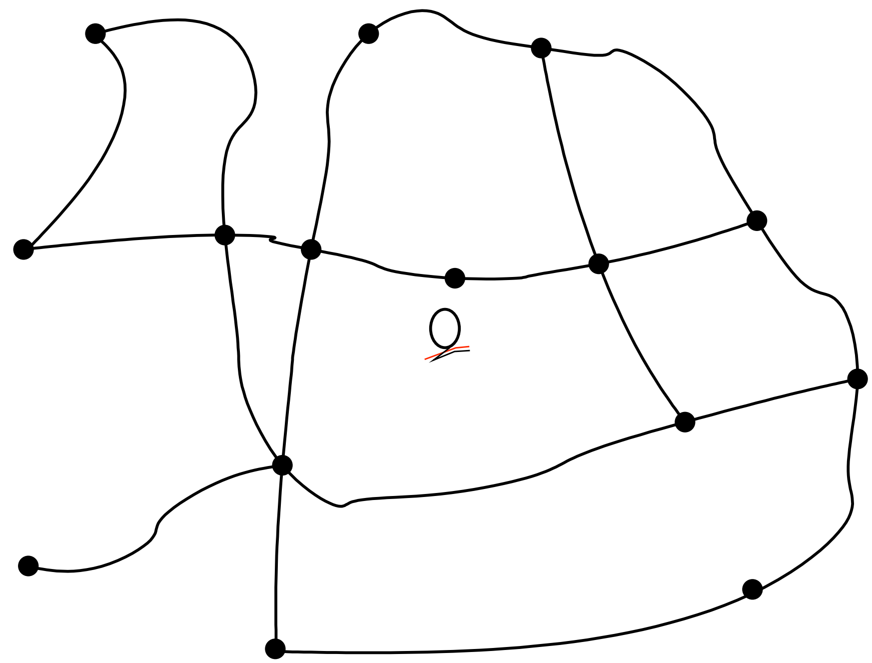
Course Notes

Graph Theory, Fall 2009

What is a graph?

Consider a simplified map of the highways near QC:



We represent each city/intersection with a **vertex** (pl: **vertices**)
(aka **node**, **point**, **junction**).

And each road with an **edge** (aka **arc**).

Note: Every edge connects two vertices.

What is a graph?

Definition: A **graph** G is a pair of sets (V, E), where V is a set of *vertices* and E is set of *edges*, themselves unordered pairs of elements of V.

Sometimes we will write $V(G)$ and $E(G)$ if we want to make clear to which vertex and edge sets we are referring.

Notation: We reserve certain variables for:

____=____=$|V(G)|$ = the number of vertices of G

____=____=$|E(G)|$ = the number of edges of G

Examples of graphs

Example 1. *Matching people and the fruit that they like.*
Erika likes cherries and dates.
Frank likes apples and cherries.
Greg likes bananas and cherries.
Helen likes apples, bananas, dates.

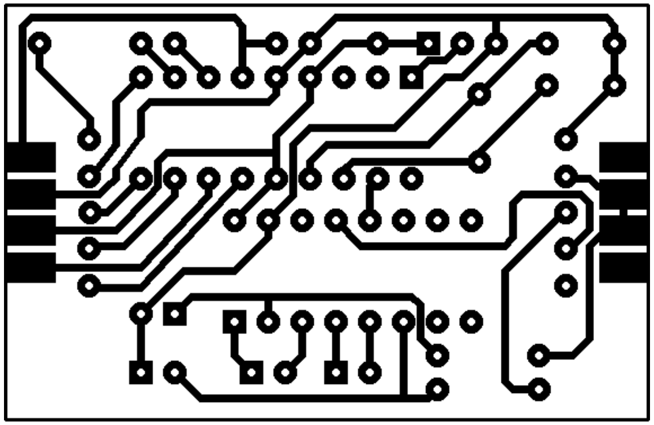
A graph can illustrate these relationships.
Create a node for each fruit and for each person.
Use edges to connect related vertices.

Q. Is there a way for each person to receive a piece of fruit he or she likes?
A.

Related topics: assignments, perfect matchings, counting questions.

Examples of graphs

Example 2. *Graph theory of a circuit board.*

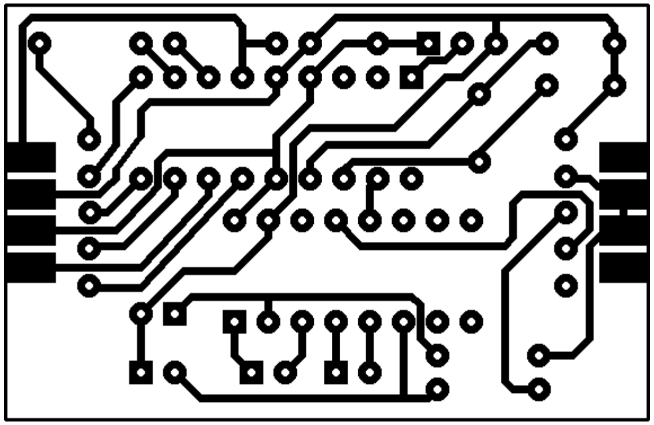


Why does it look like this?
Is the graph planar? How planar is it?
How do we embed in the plane?

Related topics: planarity, non-planarity stats, graph embeddings

Examples of graphs

Example 2. *Graph theory of a circuit board.*



Where should we drill the holes?
How to drill them as fast as possible?

Related topics: Traveling Salesman Problem, computer algorithms, optimization

What is a graph? (for now)

Let's be even more precise about the types of graphs we'll study:
For a graph $G = (V, E)$, *for now*,

- G is **finite**. (That is, $|V| < \infty$.)
  - (Infinite graphs do exist.)
- G is **simple**.
  - No two vertices are connected by more than one edge.
    - Graphs with **multiple edges** are called **multigraphs**.
  - No edge connects a vertex to itself.
    - Graphs with **loops** are called **pseudographs**.

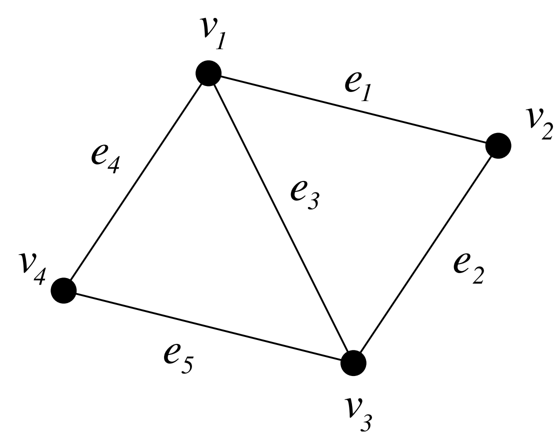
Course Structure

http://qc.edu/∼chanusa/courses/634/09_Fall/

- Written Homework: 25%
  - One is due Thursday!!!
- Class Participation: 10%
  - includes homework presentations
- Final Report: 15%
  - Report on a graph theorist
  - Lesson Plan
  - Report on a topic from graph theory
- Midterm: 25%
- Final Exam: 25%

How to talk about a graph

Example: $G = (V, E)$, where
$V = \{v_1, v_2, v_3, v_4\}$,
$E = \{e_1, e_2, e_3, e_4, e_5\}$, and
$e_1 = \{v_1, v_2\}$, $e_2 = \{v_2, v_3\}$,
$e_3 = \{v_1, v_3\}$, $e_4 = \{v_1, v_4\}$, $e_5 = \{v_3, v_4\}$



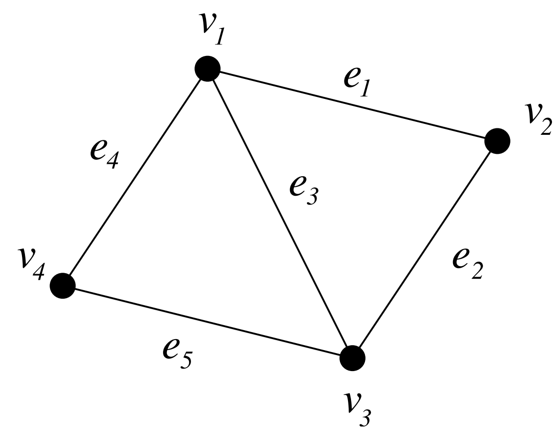
- We often write $e_1 = v_1 v_2$ with the understanding that order does not matter.

We say v_i is **adjacent** to v_2 if there is an edge between v_1 and v_2.
We also say v_1 and v_2 are **neighbors**.
Similarly, edges e_1 and e_2 are adjacent.

We say that v_1 is **incident** to/with e_1 if v_1 is an **endpoint** of e_1.

Degree of a vertex

The **degree** of a vertex v is the number of edges incident with v, and denoted $\deg(v)$.
In our example,
$\deg(v_1) = \_\_\_\_$, $\deg(v_2) = \_\_\_\_$,
$\deg(v_3) = \_\_\_\_$, $\deg(v_4) = \_\_\_\_$.



If $\deg(v) = 0$, we call v an **isolated vertex**.
If $\deg(v) = 1$, we call v an **end vertex** or **leaf**.

Degree sum Exploration:

Q. What is $\deg(v_1) + \deg(v_2) + \deg(v_3) + \deg(v_4)$?

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

Q. How many edges does G have?

A. $m =$

How are these related?

Degree of a vertex

Conjecture: *The degree sum of a graph is equal to twice the number of edges in the graph.*

Proof. Count the number of vertex-edge incidences in two ways.

Vertex-centric: For all v, how many v-e incidences are there? ____.
So the total number of vertex-edge incidences in G is __________.

Edge-centric: For all e, how many v-e incidences are there? _____.
So the total number of vertex-edge incidences in G is __________.

Since we have counted the same quantity in two different ways, the two values are equal. Therefore, for any graph $G = (V, E)$,

$\square$

Corollary: *The degree sum of a graph is always even.*

Field trip

Degree sequence of a graph

For any graph G, list the degrees of its vertices in weakly decreasing order. This is the graph's **degree sequence**.

In the graph above, the degree sequence is: ____________.

D. Every simple graph has a degree sequence.

Q. Does every sequence have a simple graph?

A.

Degree sequence of a graph

Definition: A weakly decreasing sequence of non-negative numbers is **graphic** if there is some graph that has this sequence as its degree sequence.
You can prove that a sequence is graphic by showing such a graph.

Q. How can we tell if a sequence is graphic?
A. Use the **Havel–Hakimi algorithm** from Theorem 1.1.2.

Initialization. Start with *Sequence 1*.
Step 1. Take the first number (call it s) and remove it.
Step 2. Subtract 1 from each of the next s numbers in the list.
Step 3. Order the list in non-increasing order and call the resulting sequence *Sequence 2*.

Theorem: Sequence 1 is graphic iff Sequence 2 is graphic.

⋆ Apply this algorithm multiple times to reduce to a simple case. ⋆

Proof of the Havel–Hakimi algorithm

Setup: standardize the degree sequences:

Sequence 1	$(s$	t_1	t_2	$\ldots$	t_s	d_1	$\ldots$	$d_k)$.
Sequence 2	$($	$t_1 - 1$	$t_2 - 1$	$\ldots$	$t_s - 1$	d_1	$\ldots$	$d_k)$.

Theorem: Sequence 1 is graphic **iff** *Sequence 2 is graphic.*

Proof: (S_2 graphic $\Rightarrow$ S_1 graphic) Suppose that S_2 is graphic. Therefore, there exists a graph G_2 with degree sequence S_2. We will construct a graph G_1 that has S_1 as its degree sequence.

Can this argument work in reverse?

Proof of the Havel–Hakimi algorithm

Proof: (S_1 graphic $\Rightarrow$ S_2 graphic) Suppose that S_1 is graphic. Therefore, there exists a graph G_1 with degree sequence S_1.
We can not directly construct a graph G_2 with degree sequence S_2.
However, we will construct a graph with degree sequence S_2 in stages.

Game plan:

$$G_1 \longrightarrow G_2 \longrightarrow G_3 \longrightarrow \cdots \longrightarrow G_a$$

- Start with G_1 which we know exists.
- At each stage, create a new graph G_i from G_{i-1} such that
  - G_i has degree sequence S_1.
  - The vertex of degree s in G_i is adjacent to MORE of the highest degree vertices than G_{i-1}.
- After some number of iterations, the vertex S of degree s in G_a will be adjacent to the next s highest degree vertices.
- Peel off vertex S to reveal a graph with degree sequence S_2.

Proof of the Havel–Hakimi algorithm