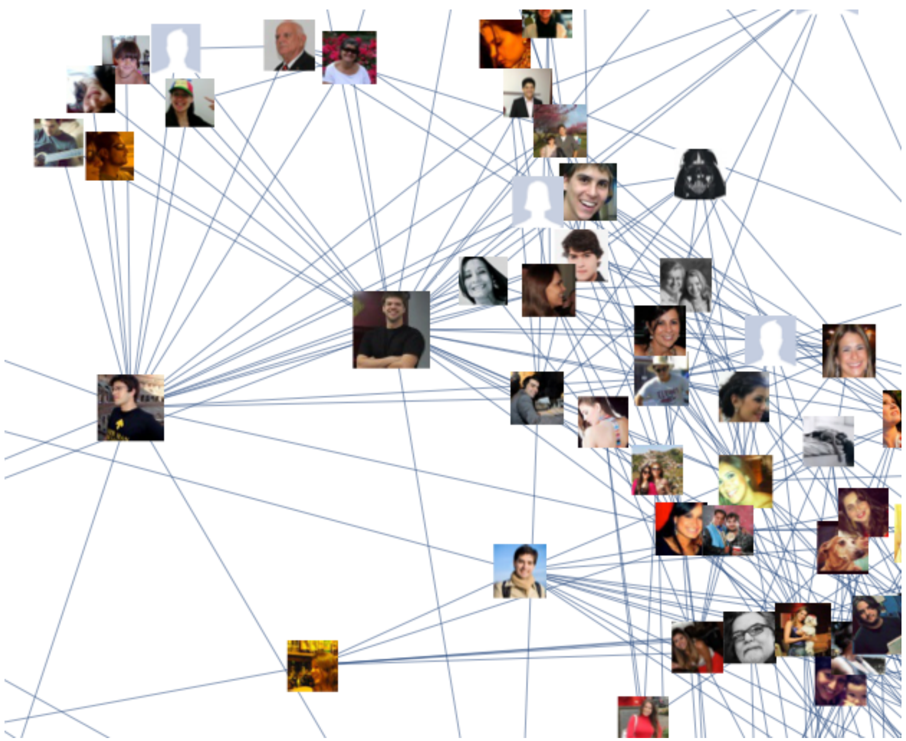
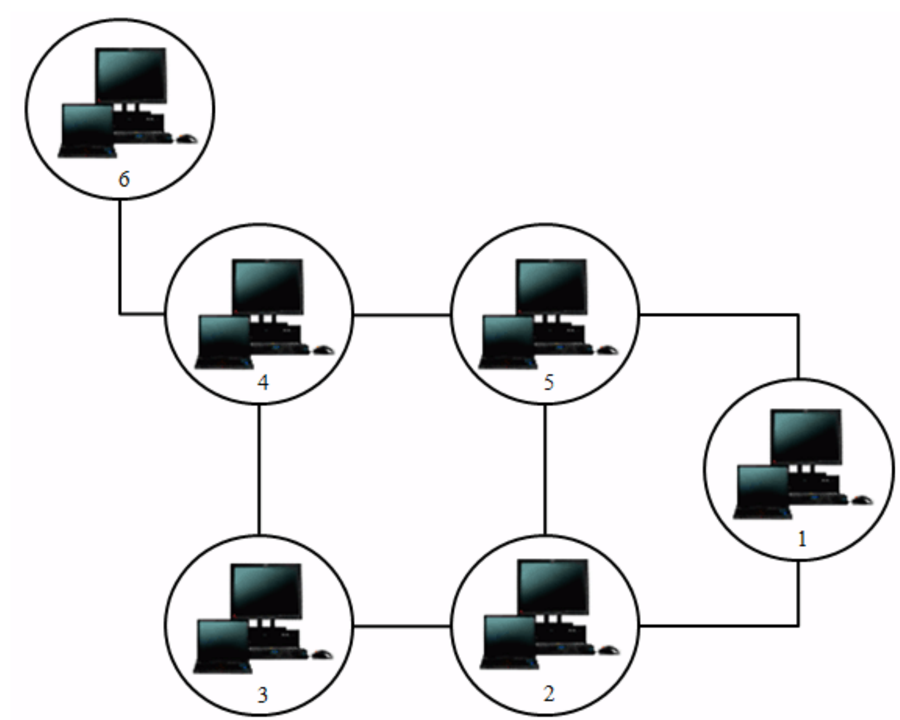
15-251
Great Theoretical Ideas in Computer Science

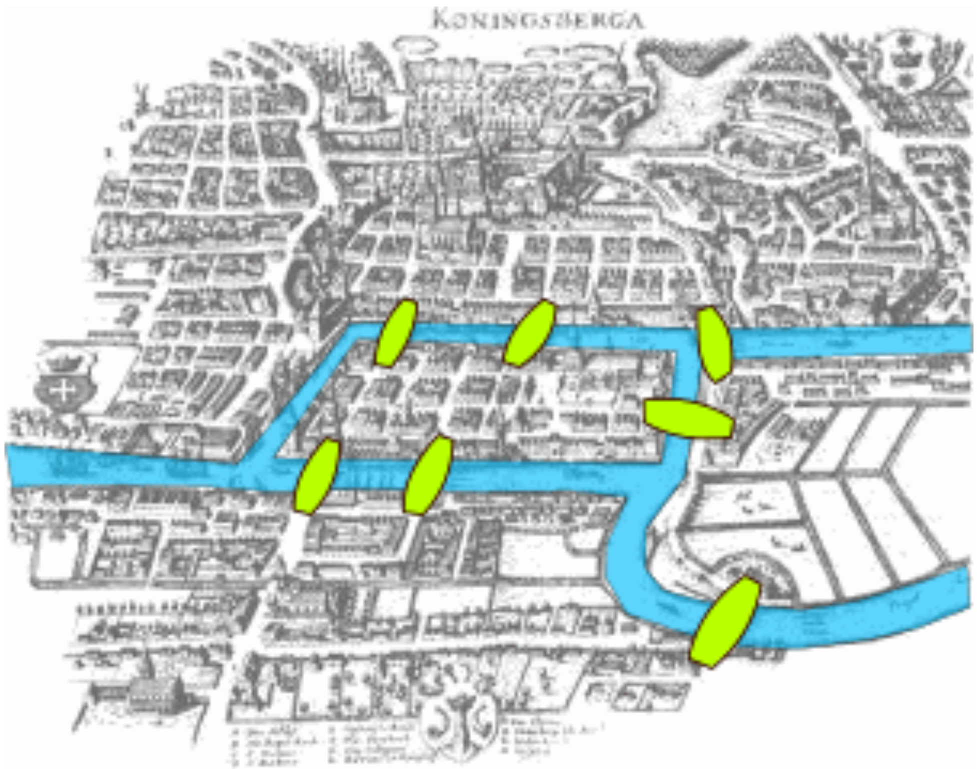
Lecture 9:
Graphs I: The Basics





September 27th, 2016

Crossing bridges



Königsberg (Prussia)

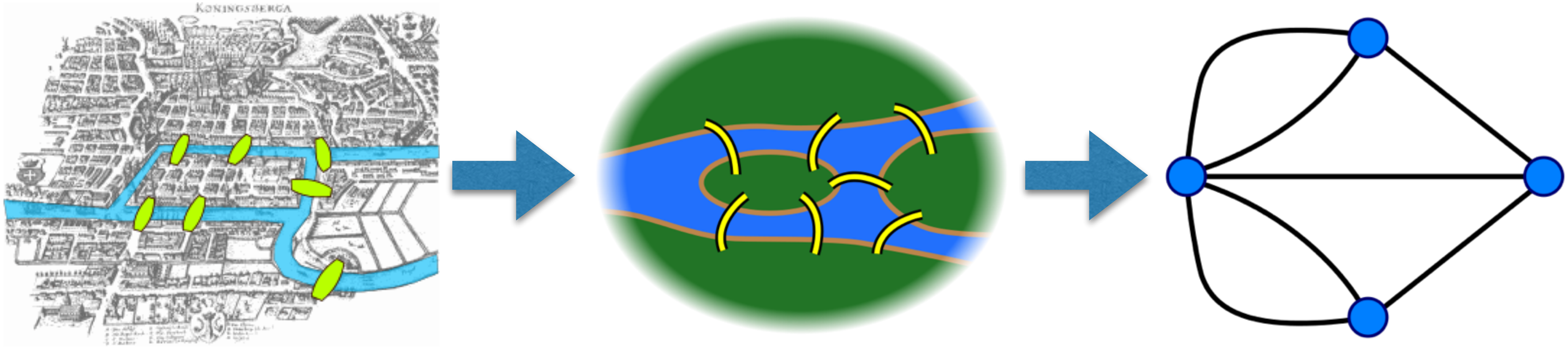
Now
Kaliningrad (Russia)

Is there a way to walk through the city that would cross each bridge **exactly** once?

Leonhard Euler
(1735)

This is not possible!

Crossing bridges



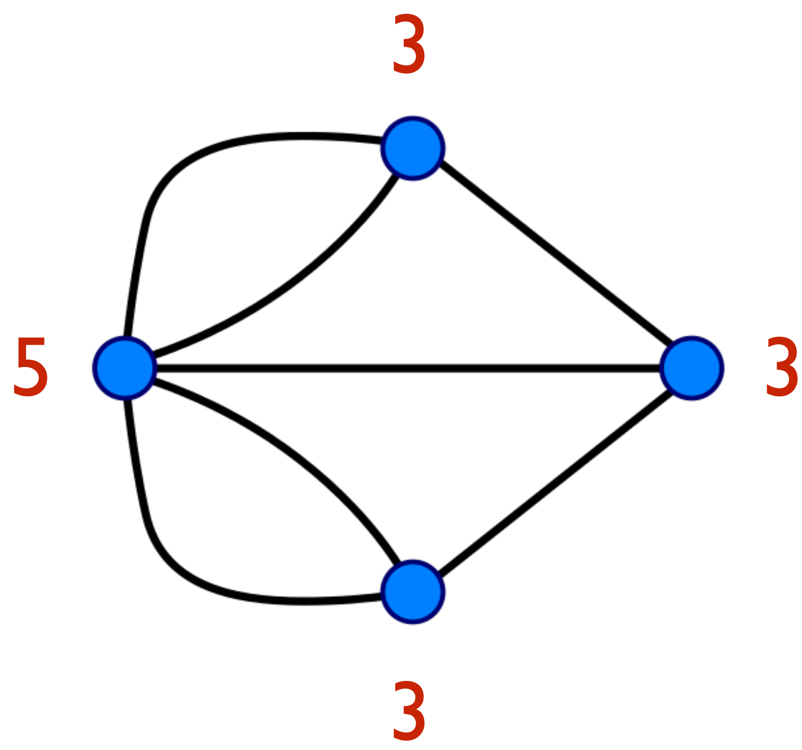
A "graph" with 4 *nodes/vertices* and 7 *edges*.

Except for the start and end vertices:

- Whenever one *enters* a vertex by an edge, one must *leave* by another edge.
- # edges incident to a vertex must be even.

(start and end vertex must not have this property, unless start = end)

Crossing bridges



Every vertex is incident to an odd number of edges.

So this graph does not have an "*Eulerian tour*".

Crossing bridges

Ok, that wasn't too bad.

But 7 bridges.. Come on. Pittsburgh has **446**.

What if it *is* the case that exactly 0 or 2 nodes are incident to an odd number of vertices?

Does that imply the graph must have an Eulerian tour?

Why graphs?

Why now?

Facebook


Graph is big and changing
1 billion people
240 billion photos
1 trillion connections

Enemybook

Kevin Matulef

Enemybook remedies the one-sided perspective of Facebook, by allowing you to manage enemies as well as friends. With Enemybook you can **add people** as Facebook enemies, **specify why** they are your enemies, **notify** your enemies, **see who lists you** as an enemy, and even **become friends with the enemies of your enemies**.

Enemybook

Browse Your Enemies:

Kevin Matulef's Enemybook

(See who's listed you! ➡)

Enemy List | **Add Enemies**

You have 2 enemies in your enemybook.

Name: Mark Zuckerberg

Networks: Facebook
Harvard
San Francisco, CA

Enemy Details: Facebook ruined Mark's and your relationship.
You don't even know Mark, but hate him already.

[edit details]

View Enemies
Remove as Enemy
Tell Friends to "Enemy"
View Friends
Add as Friend
Send Message
Flip Off Mark!

Name: George Bush

Networks: Boston, MA

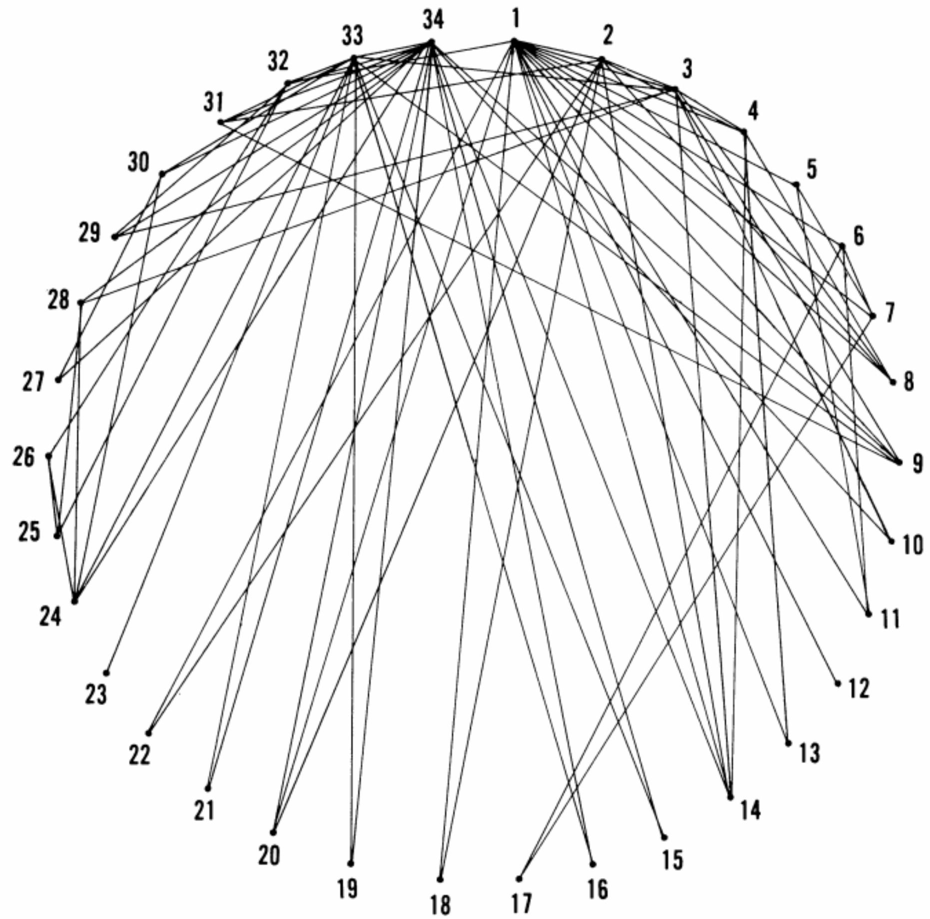
Enemy Details: George insulted your intelligence.

[edit details]

View Enemies
Remove as Enemy
Tell Friends to "Enemy"
View Friends
Add as Friend
Send Message
Flip Off George!

Zachary Karate Club

FIGURE 1
Social Network Model of Relationships in the Karate Club



This is the graphic representation of the social relationships among the 34 individuals in the karate club. A line is drawn between two points when the two individuals being represented consistently interacted in contexts outside those of karate classes, workouts, and club meetings. Each such line drawn is referred to as an edge.

Zachary Karate Club CLUB

networkkarate.tumblr.com

Google PageRank

1998 paper

2.2 Link Structure of the Web

While estimates vary, the current graph of the crawlable Web has roughly 150 million nodes (pages) and 1.7 billion edges (links). Every page has some number of forward links (outedges) and backlinks (inedges) (see Figure 1). We can never know whether we have found all the backlinks of a particular page but if we have downloaded it, we know all of its forward links at that time.

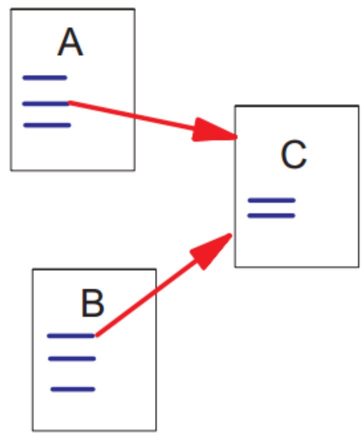


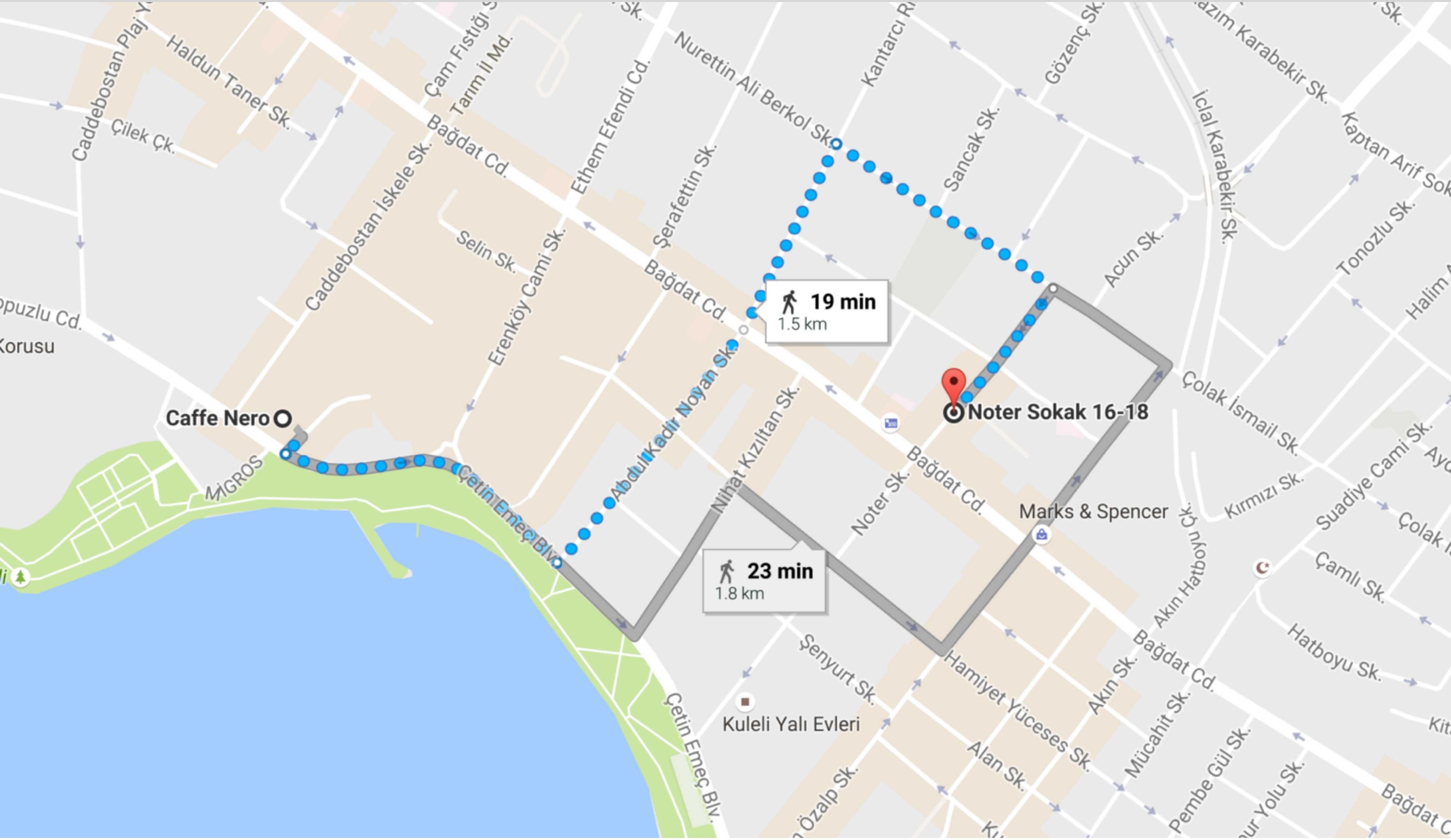
Figure 1: A and B are Backlinks of C

Web pages vary greatly in terms of the number of backlinks they have. For example, the Netscape home page has 62,804 backlinks in our current database compared to most pages which have just a few backlinks. Generally, highly linked pages are more "important" than pages with few links. Simple citation counting has been used to speculate on the future winners of the Nobel Prize [San95]. PageRank provides a more sophisticated method for doing citation counting.

Larry Page

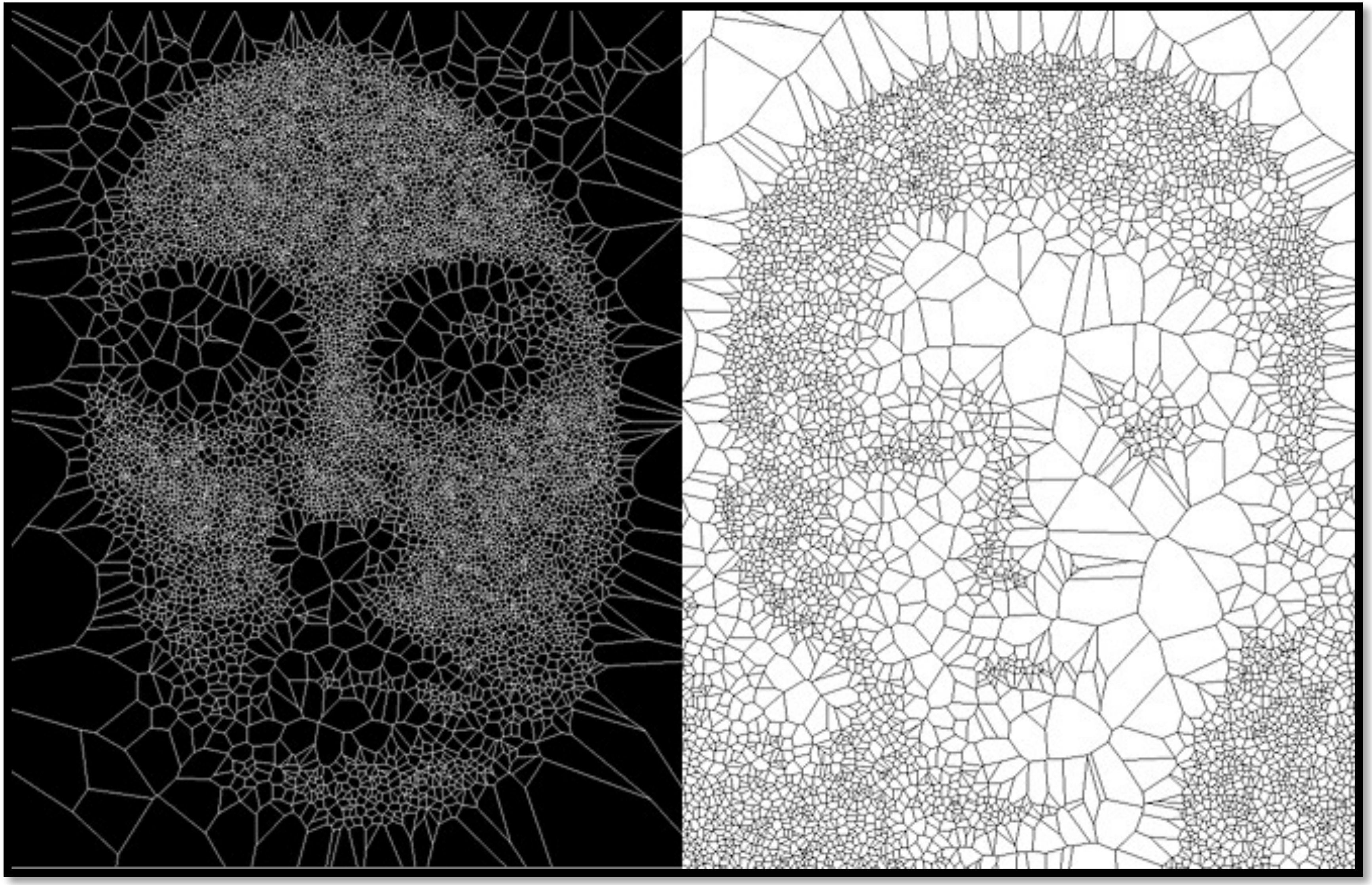
Sergey Brin

Street Maps

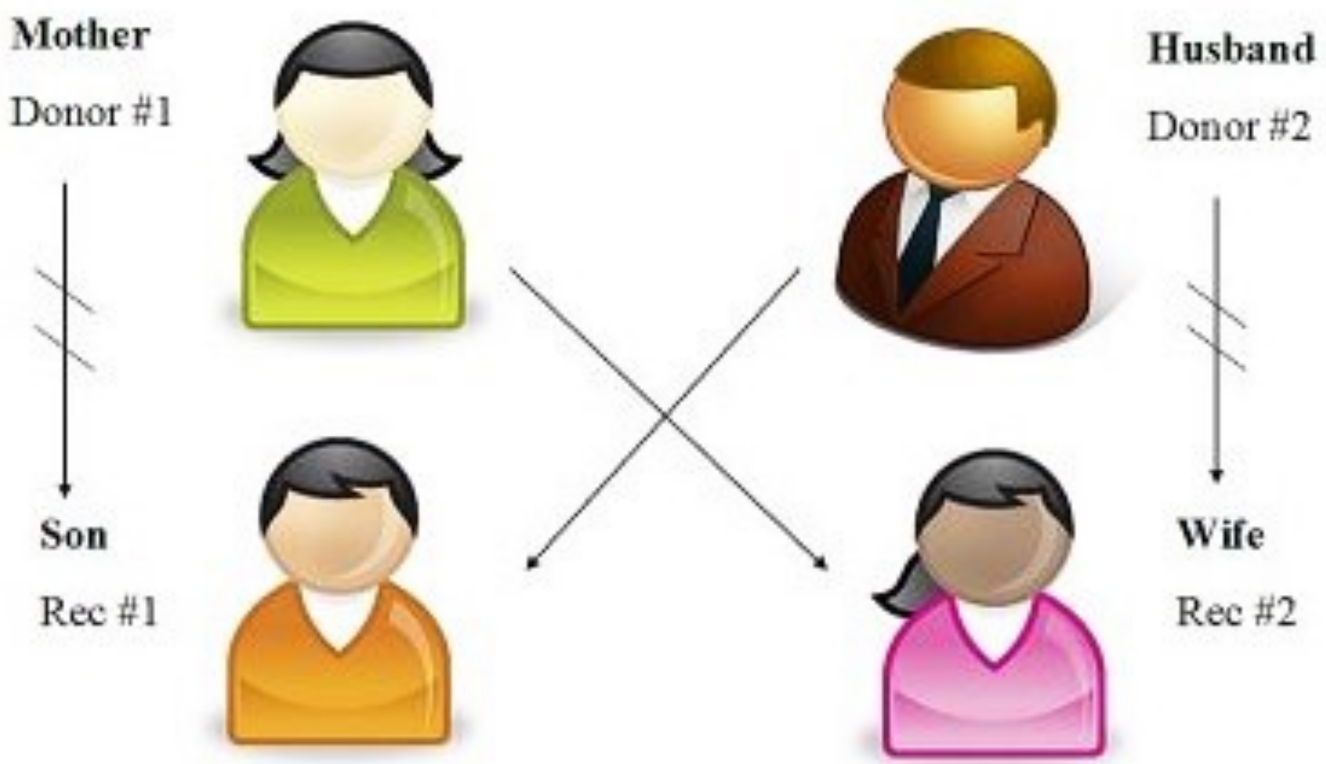


Caffe Nero
Noter Sokak 16-18
19 min
1.5 km
23 min
1.8 km
Marks & Spencer
Kuleli Yalı Evleri
MIGROS
Bağdat Cd.
Çetin Emeç Blv.
Haldun Taner Sk.
Çilek Çk.
Caddebostan İskele Sk.
Selin Sk.
Erenköy Cami Sk.
Ethem Efendi Cd.
Şerafettin Sk.
Nurettin Ali Berkol Sk.
Kantarcı R
Sancak Sk.
Gözenç Sk.
Acun Sk.
İclal Karabekir Sk.
Kaptan Arif Sok
Tonozlu Sk.
Çolak İsmail Sk.
Kırmızı Sk.
Suadiye Cami Sk.
Çamlı Sk.
Hatboyu Sk.
Akın Hatboyu Çk.
Akın Sk.
Mücahit Sk.
Pembe Gül Sk.
Hamiyet Yüceses Sk.
Alan Sk.
Şenyurt Sk.
Nihat Kızıltan Sk.
Noter Sk.
Abdulkadir Noyan Sk.
Tarım İl Md.
Korusu

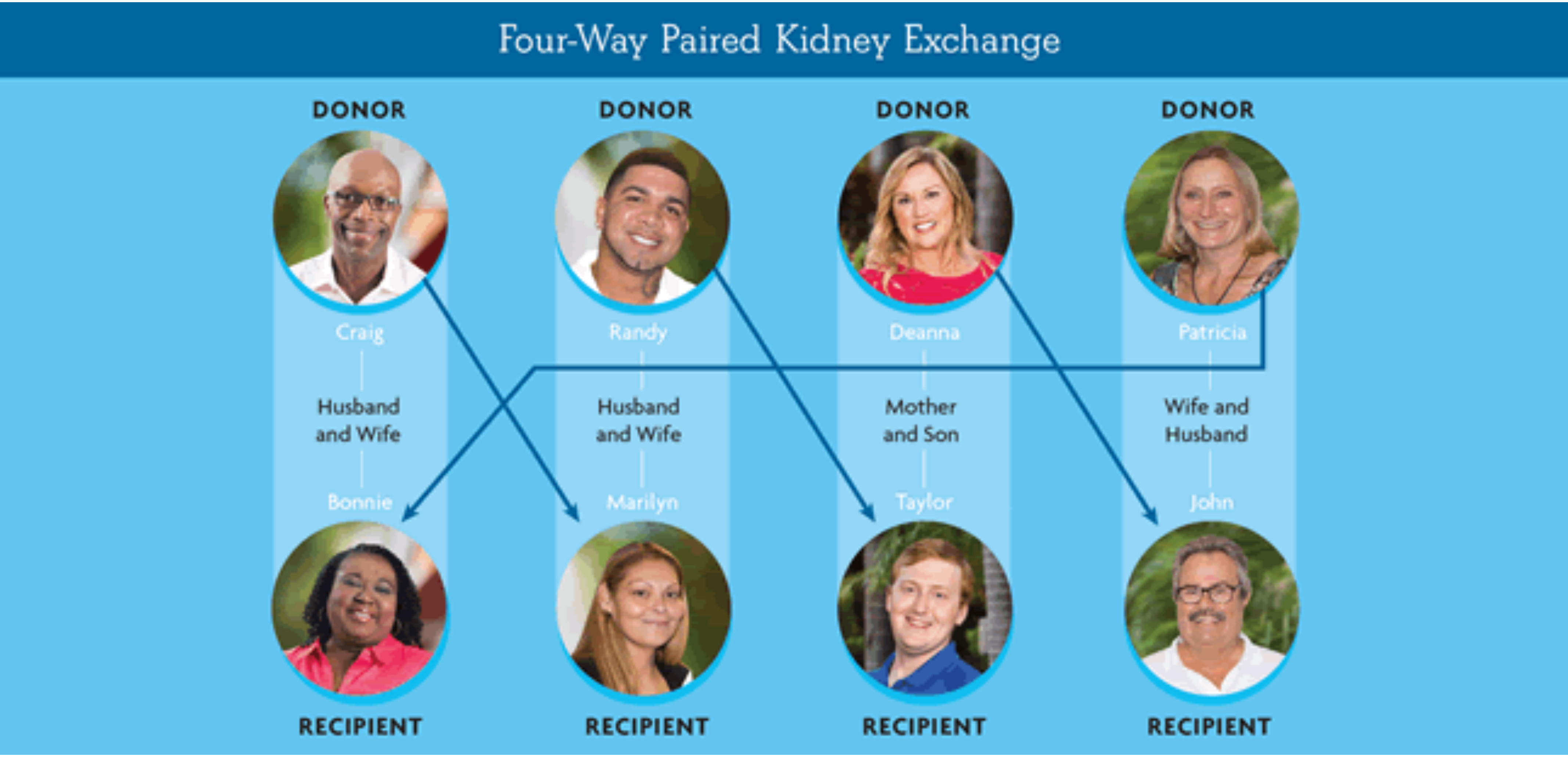
Images



Kidney Exchange

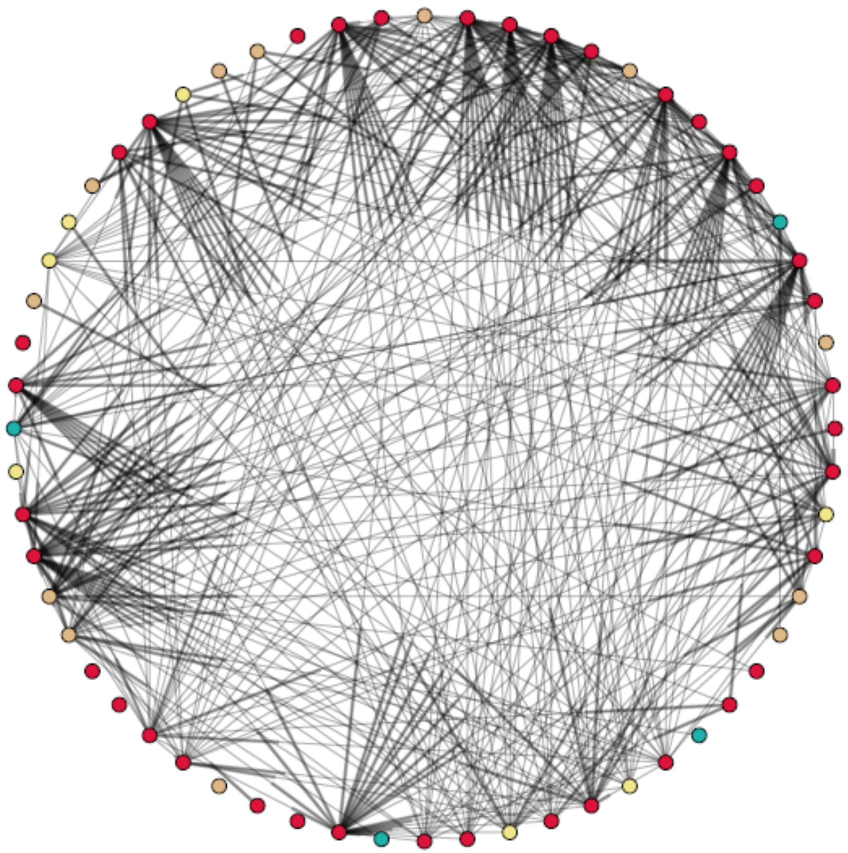


Kidney Exchange



Kidney Exchange

Vertices = patient-donor pairs, edges = compatibility



UNOS pool, Dec 2010 [Courtesy John Dickerson, CMU]

Tuomas Sandholm (CMU prof.)

Computer Science Life Lesson

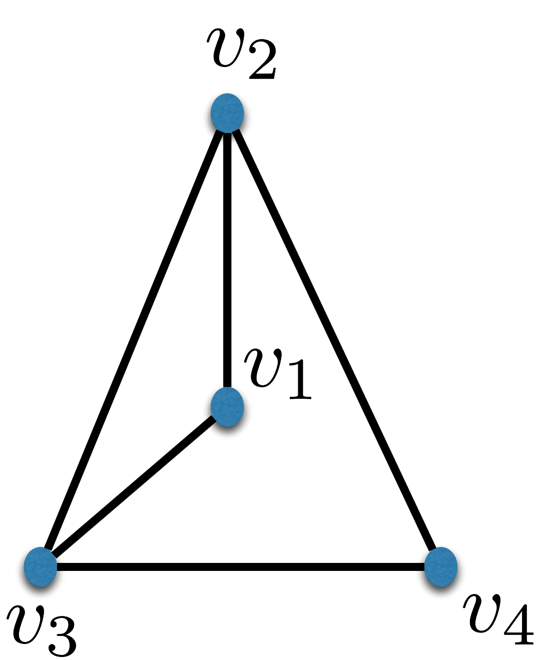
If your problem has a graph, 😃 👍 .

If not, try to make it have a graph.

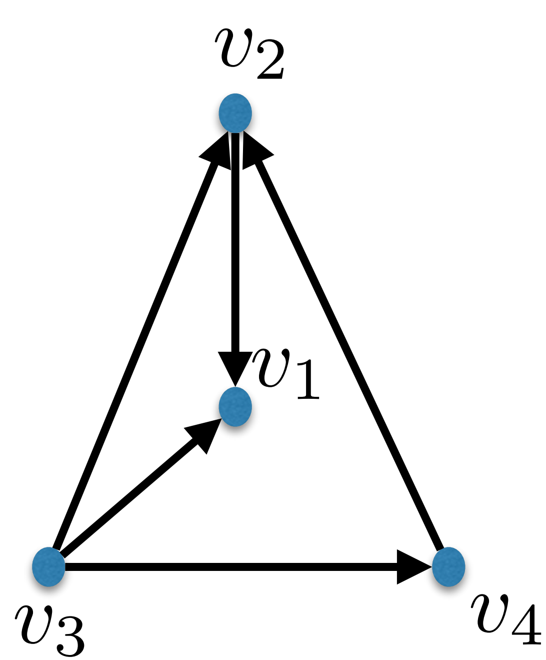
What is a graph?

(A hundred) definitions and some basic properties

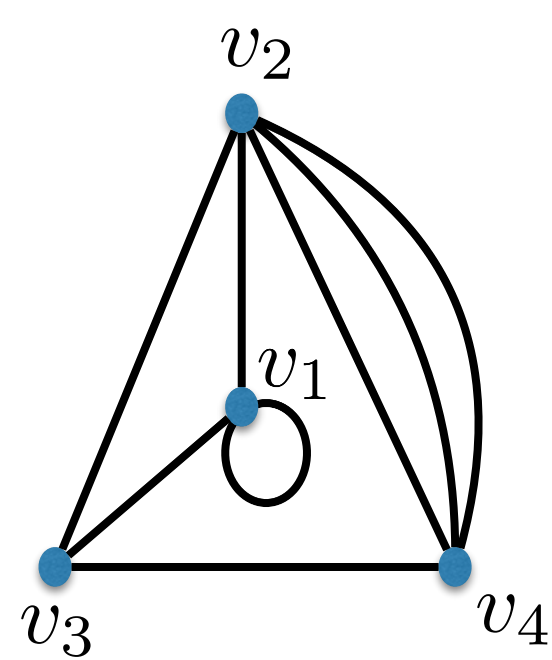
Types of Graphs



Simple
Undirected
Graph



Directed
Graph



Multigraph

Formal Definition: (undirected) graph

A **graph** G is a tuple (V, E), where

- V is a finite set called the set of vertices (or nodes).
- E is a set called the set of edges.

Each edge $e \in E$ is of the form $\{u, v\}$ for distinct $u, v \in V$.

Example:

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$$

$$E = \{\{v_1, v_2\}, \{v_1, v_4\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_5, v_6\}\}$$

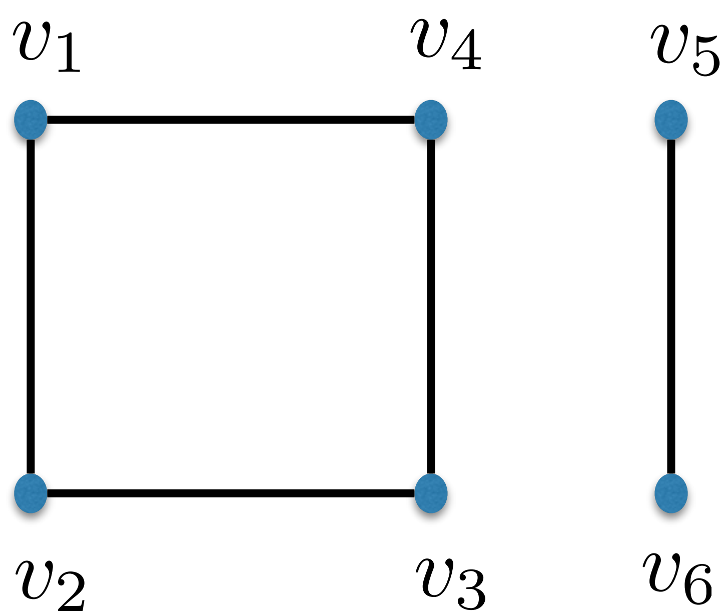
Formal Definition: (undirected) graph

Example:

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$$

$$E = \{\{v_1, v_2\}, \{v_1, v_4\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_5, v_6\}\}$$

Graphs can be drawn:



Formal Definition: (undirected) graph

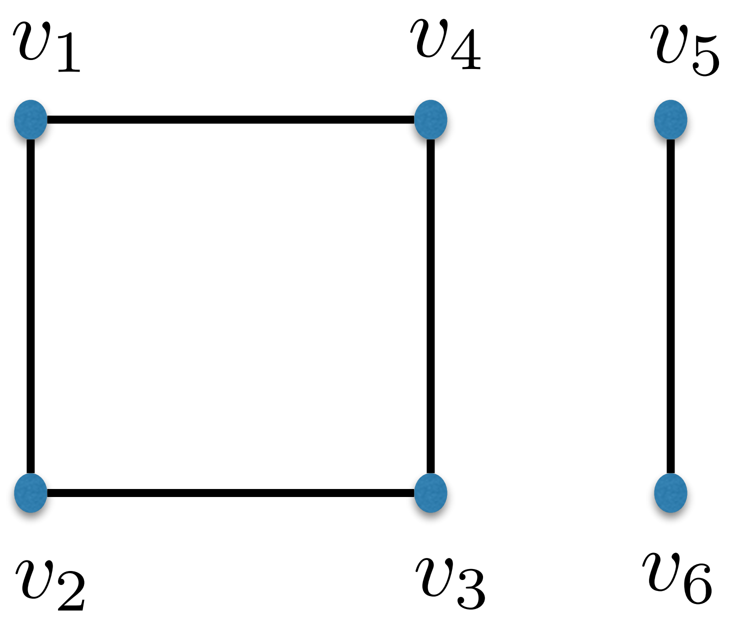
Example:

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$$

$$E = \{\{v_1, v_2\}, \{v_1, v_4\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_5, v_6\}\}$$

Matrix representation (adjacency matrix):

$$\begin{array}{c c} & \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$



IMPORTANT Notation

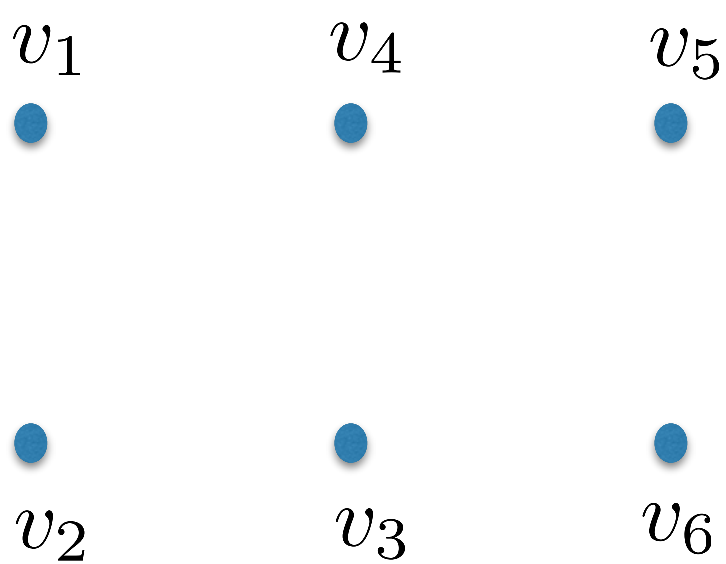
Almost always:

n = number of vertices in the graph, $|V|$

m = number of edges, $|E|$

Edge cases

Is it possible that $E = \emptyset$?

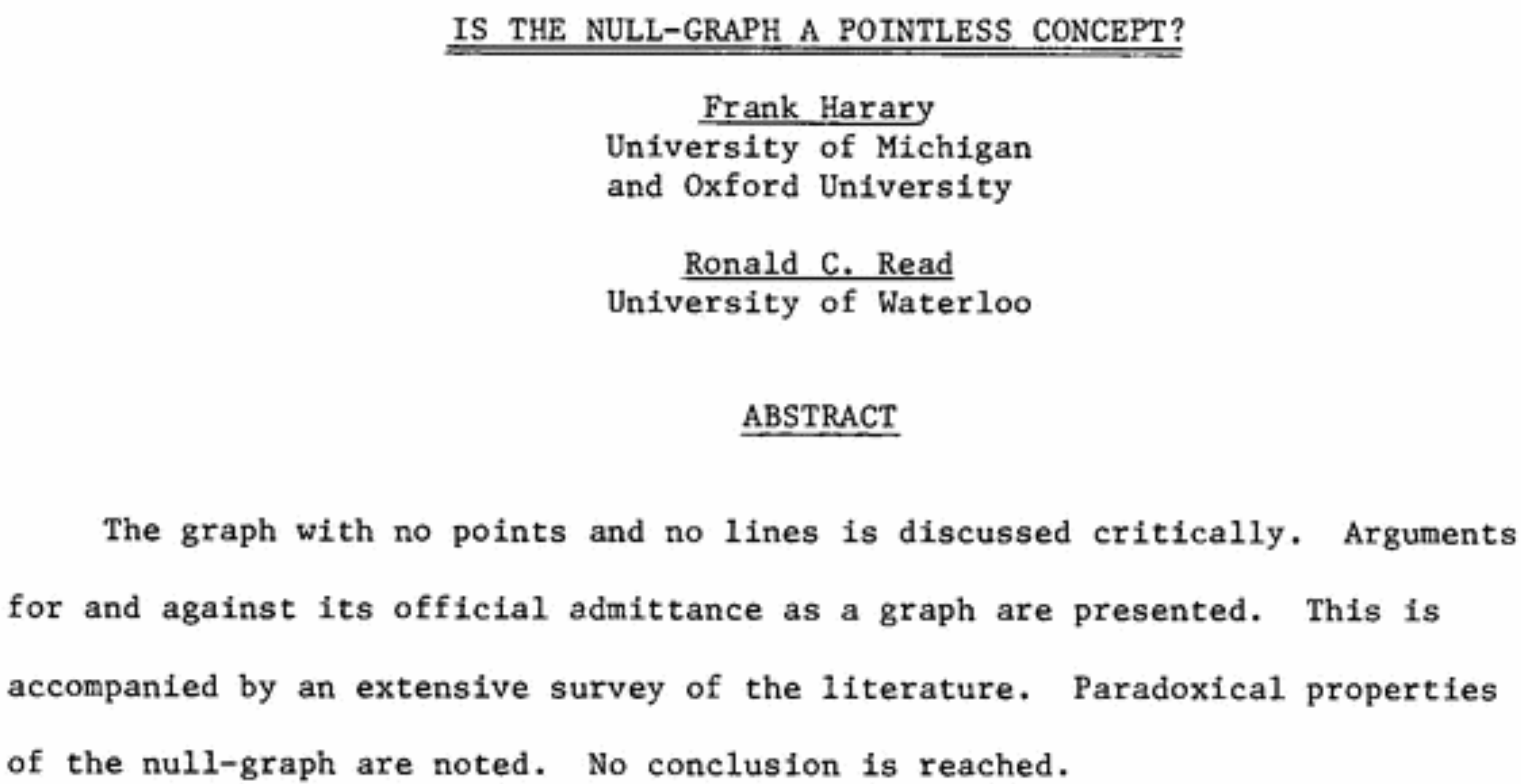


"Empty graph" with 6 vertices.

6 "isolated" vertices.

Is it possible that $V = \emptyset$?

The Null Graph


IS THE NULL-GRAPH A POINTLESS CONCEPT?

Frank Harary
University of Michigan
and Oxford University

Ronald C. Read
University of Waterloo

ABSTRACT

The graph with no points and no lines is discussed critically. Arguments for and against its official admittance as a graph are presented. This is accompanied by an extensive survey of the literature. Paradoxical properties of the null-graph are noted. No conclusion is reached.

The Null Graph

Figure 1. The Null Graph

1st Challenge

Is it possible to have a party with 251 people in which everyone knows exactly 5 other people in the party?

Is it possible to have a graph with 251 vertices in which each vertex is adjacent to exactly 5 other vertices?

Terminology: Neighbor

Suppose $e = \{u, v\} \in E$ is an edge.

We say:

u and v are **endpoints** of e

u and v are **adjacent**

u and v are **incident** on e

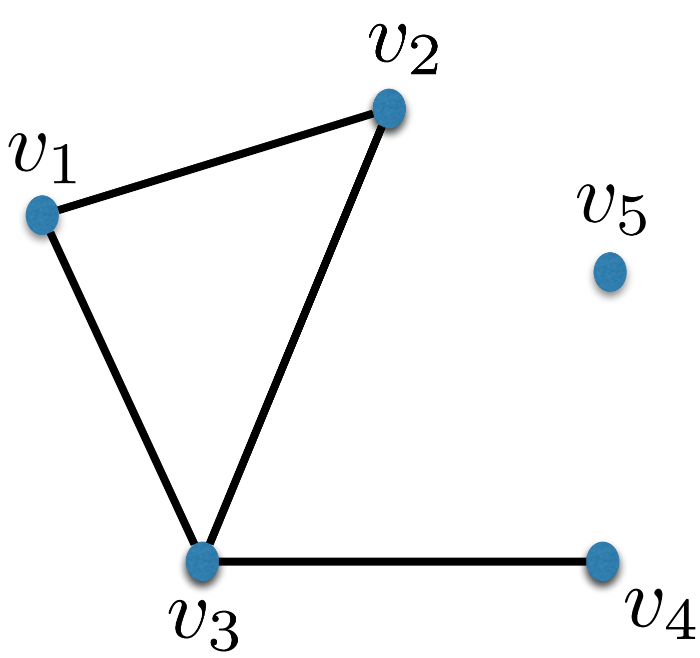
u is a **neighbor** of v

v is a **neighbor** of u

Terminology: Neighborhood

For $v \in V$, the **neighborhood** of v is defined as

$$N(v) = \{u \in V : \{v, u\} \in E\}.$$



$$N(v_1) = \{v_2, v_3\}$$

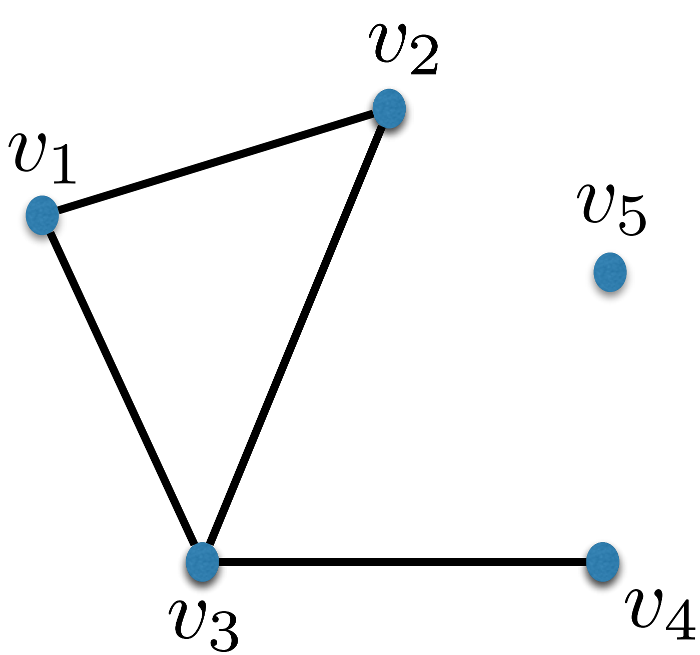
$$N(v_3) = \{v_1, v_2, v_4\}$$

$$N(v_5) = \emptyset$$

Terminology: Degree

For $v \in V$, the **degree** of v is defined as

$$\deg(v) = |N(v)|.$$



$$\deg(v_1) = 2$$

$$\deg(v_3) = 3$$

$$\deg(v_5) = 0$$

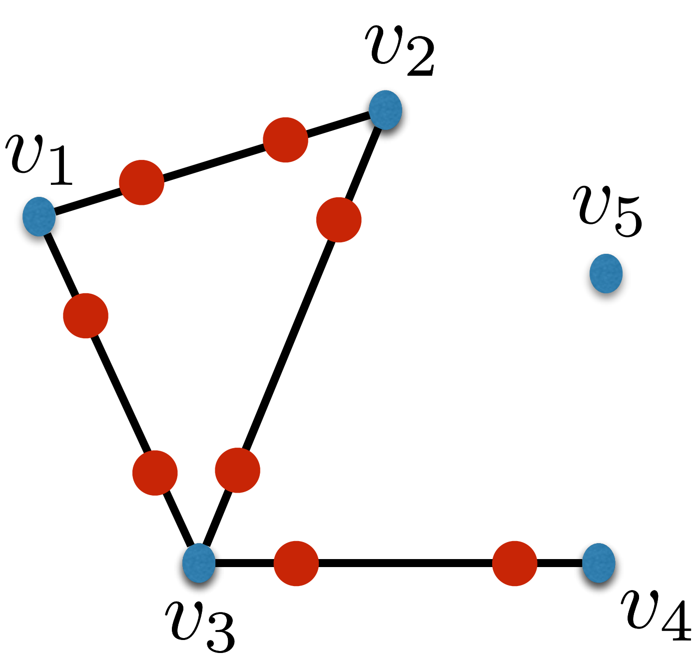
A graph is called **d-regular** if $\forall v \in V, \quad \deg(v) = d.$

1st Theorem

Theorem: Let $G = (V, E)$ be a graph. Then

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

Proof:



Place tokens on edges:

- each vertex puts a token on each edge it's incident to.

Observations:

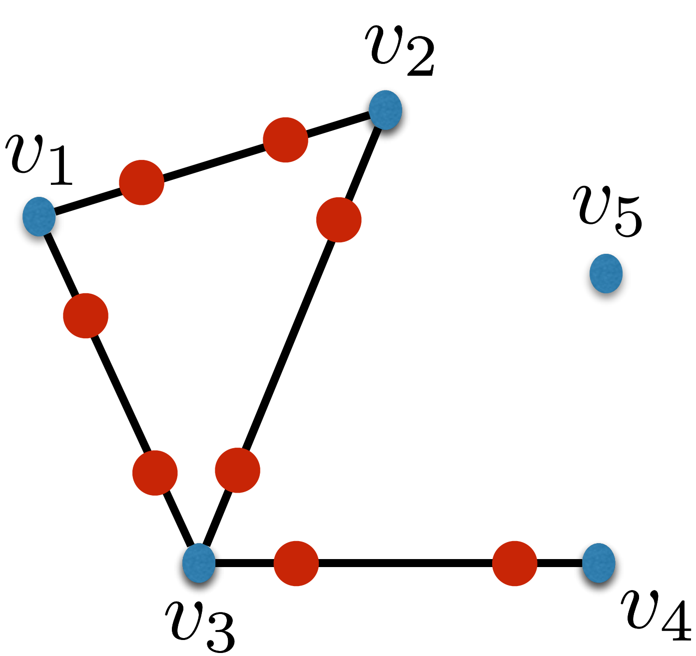
- vertex v puts $\deg(v)$ tokens.
- each edge gets 2 tokens.

1st Theorem

Theorem: Let $G = (V, E)$ be a graph. Then

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

Proof:



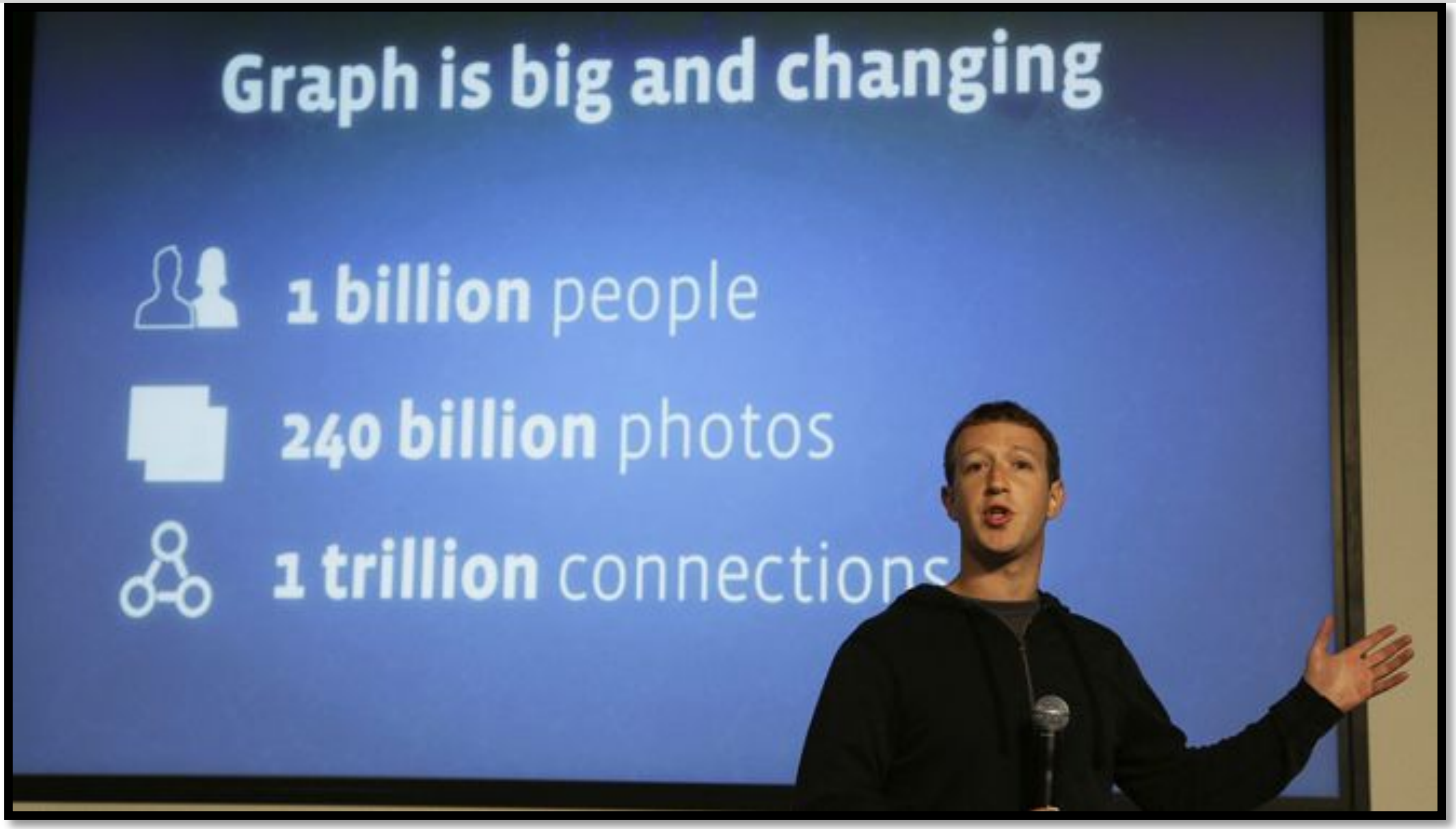
Count the total # tokens:

1st way: $\sum_{v \in V} \deg(v)$

2nd way: $2m$

$\square$

Back to Facebook for a sec



$$m = 1000000000000 \qquad n = 1000000000$$

$$2m = 2000000000000 \quad = \sum_{v \in V} \deg(v)$$

$\implies$ on average, people have 2000 friends.

Poll

Is it possible to have a graph with 251 vertices in which each vertex is adjacent to exactly 5 other vertices?

Yes

No

Beats me

2nd Challenge

We have n computers that we want to connect.

We can put a link between any two computers, but the links are expensive.

What is the least number of links we can use?

What is the least number of edges needed to connect n vertices?

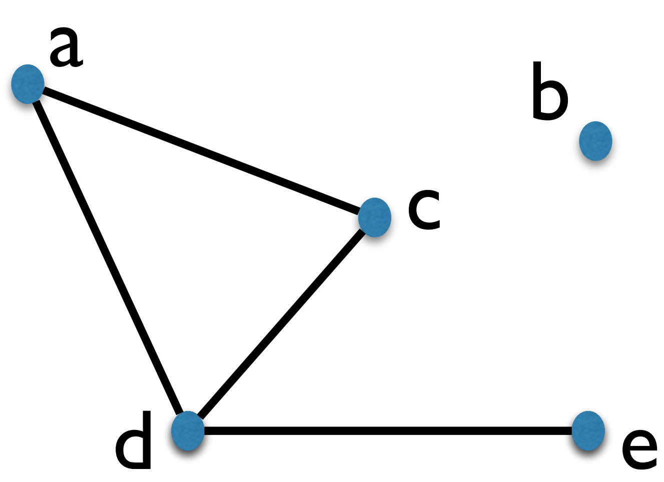
Walks and Paths

A **walk** in a graph G = (V, E) is a sequence of vertices

$$v_0, v_1, v_2, \ldots, v_k \qquad\qquad (k \geq 0)$$

such that $\{v_{i-1}, v_i\} \in E$ for all $i \in \{1, \ldots, k\}$.

We say that this is a walk from v_0 to v_k ,
and its length is k .



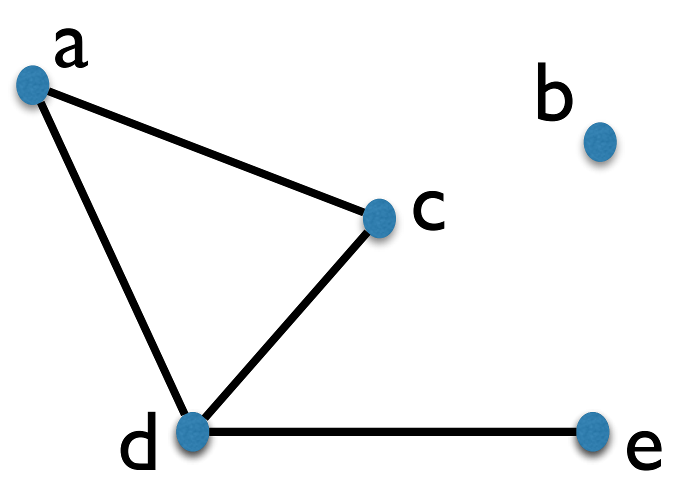
(a, c, d, a, d, e)

is a walk from a to e
of length 5.

Walks and Paths

A **path** in a graph G = (V, E) is
a walk with **<u>no repeated vertices</u>**.

<u>Fact:</u> There is a path from u to v iff
there is a walk from u to v



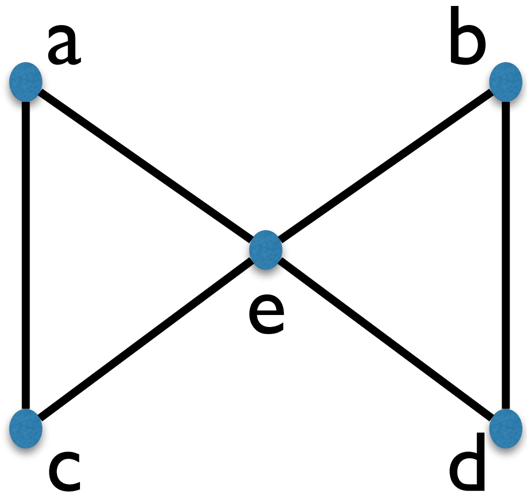
(a, c, d, a, d, e)

"shortcut"
repeated vertices

(a, d, e)

Circuits and Cycles

A **circuit** in a graph G = (V, E) is a walk from u to u (for some u).

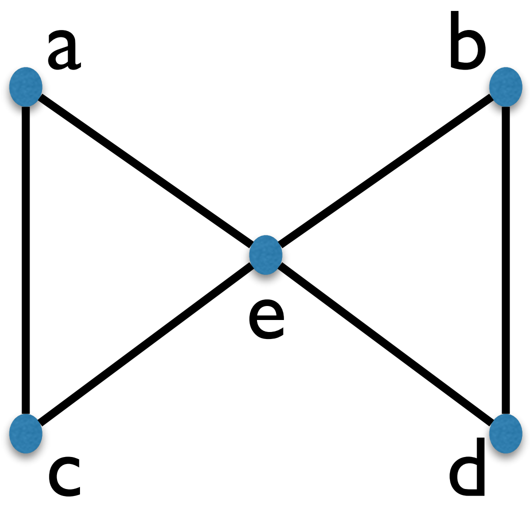


(a, e, b, d, e, c, a)

is a circuit

Circuits and Cycles

A **cycle** in a graph G = (V, E) is a walk from u to u with **no repeated vertices** (except for u). (of length ≥ 3)



(a, c, e, a) is a cycle

(a, e, c, a) is considered the same cycle

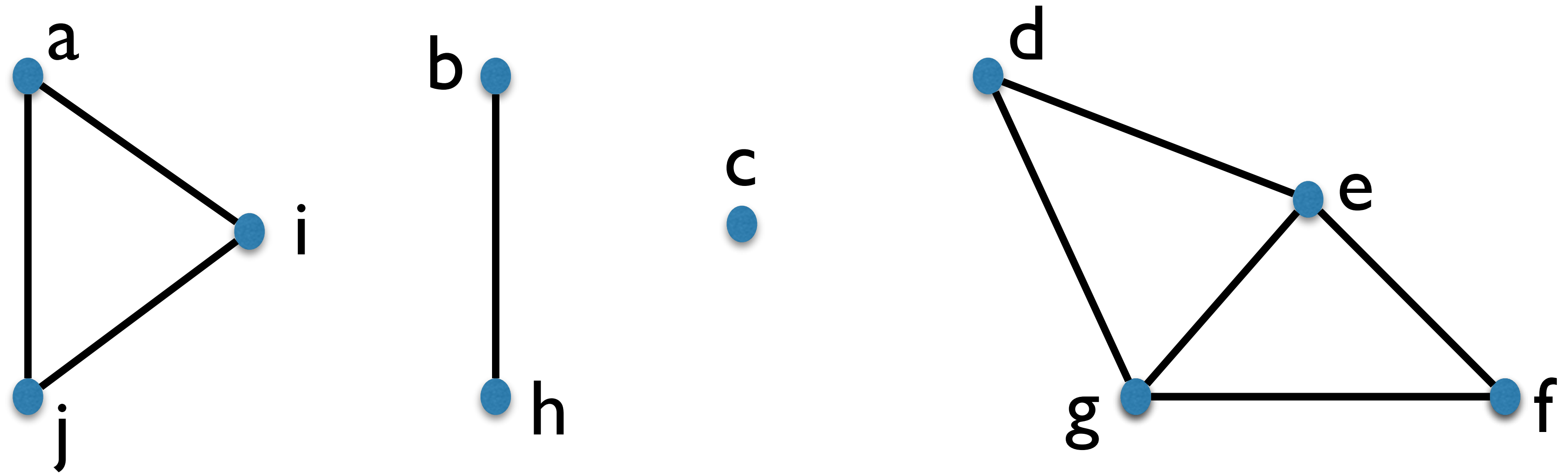
(e, a, c, e) is considered the same cycle

(e, b, d, e) is a cyle

A graph with no cycles is called **acyclic**.

Connected Graphs

A graph is **connected** if there is a path between any two vertices of the graph.



This 10-vertex graph is **not** connected.

It has 4 **connected components**:

{a, i, j}, {b, h}, {c}, {d, e, f, g}

A graph is connected iff it has 1 connected component.

Back to the challenge

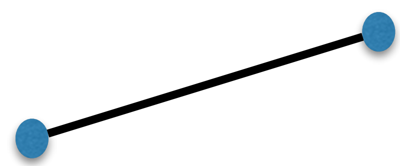
What is the least number of edges needed to connect n vertices?

n = 1

m = 0

necessary and sufficient

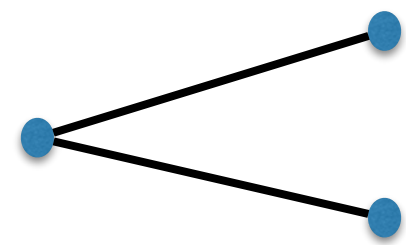
n = 2



m = 1

necessary and sufficient

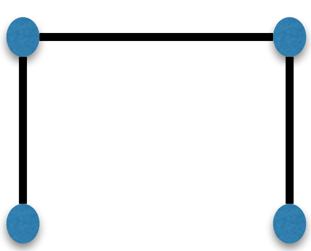
n = 3

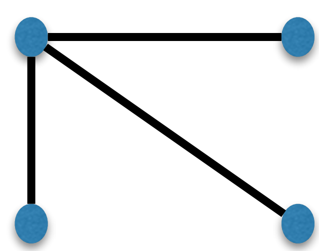


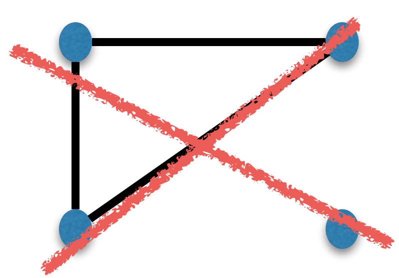
m = 2

necessary and sufficient

n = 4







m = 3

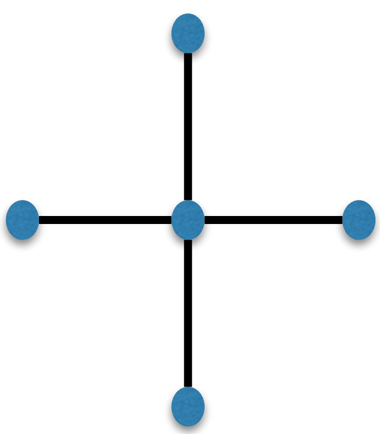
necessary and sufficient

Back to the challenge

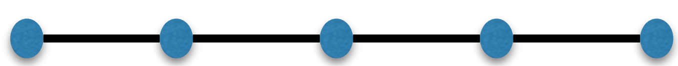
What is the least number of edges needed to connect n vertices?

n-1 edges are always sufficient

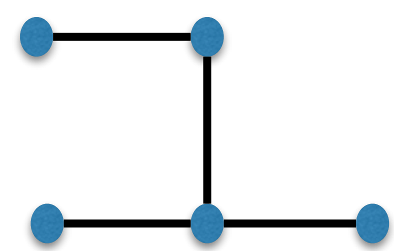
“star graph”



“path graph”



“something else”



n-1 edges always necassary?

Poll

Are n-1 edges always necassary to connect n vertices?

Yes

No

No opinion

2nd Theorem

Theorem: Let $G = (V, E)$ be a **connected** graph.

Then $m \geq n - 1$.

Furthermore,

$$m = n - 1 \iff G \text{ is acyclic.}$$

Proof:

Imagine the following process:

- remove all the edges of G.
- add them back one by one (in an arbitrary order).

n isolated vertices $\rightarrow$ G

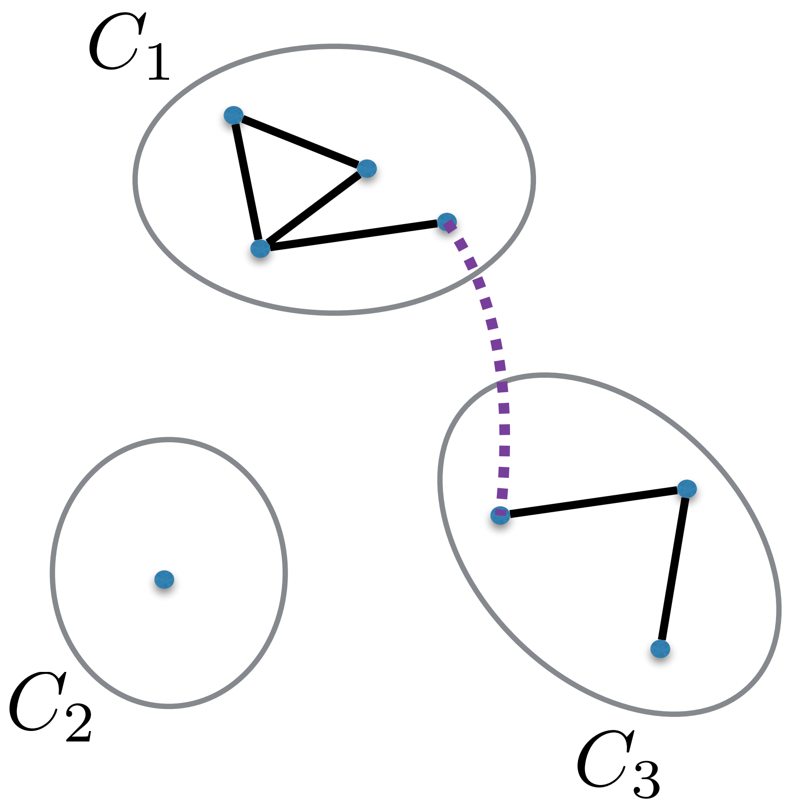
n CCs $\rightarrow$ 1 CC

CC = connected component

2nd Theorem

Proof (continued):

Consider a step of adding an edge back.



2 possibilities:

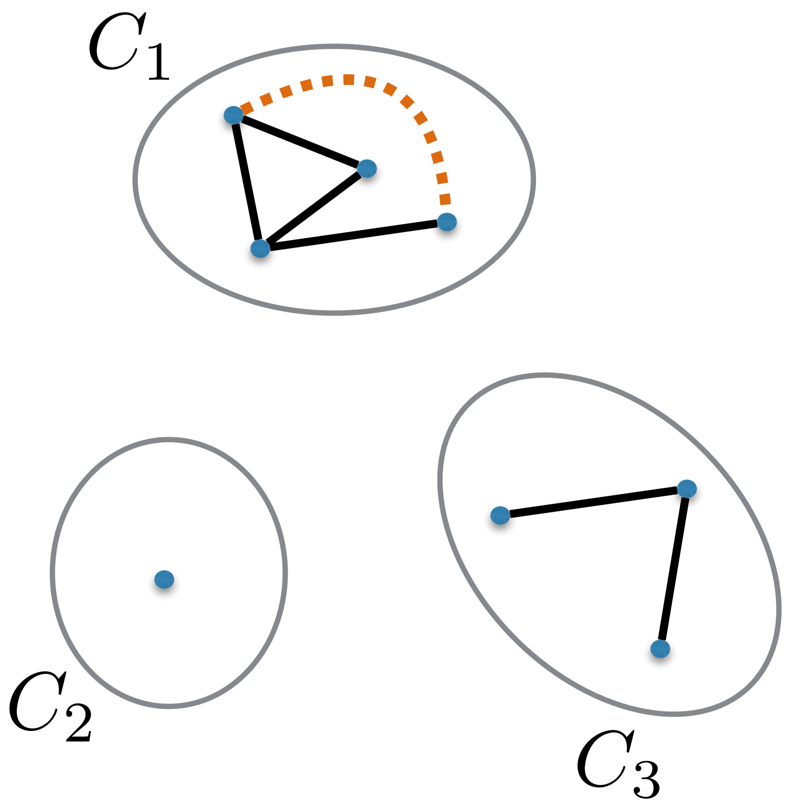
(i) connector edge

- connects 2 CCs.
- # CCs goes down by 1.
- cannot create a new cycle.

2nd Theorem

Proof (continued):

Consider a step of adding an edge back.



2 possibilities:

(i) connector edge

- connects 2 CCs.
- # CCs goes down by 1.
- cannot create a new cycle.

(ii) cycle creator edge

- an edge within a CC.
- # CCs stays the same.
- creates a new cycle.

2nd Theorem

Proof (continued):

Consider a step of adding an edge back.

2 possibilities:

(i) connector edge — # CCs goes down by 1.

(ii) cycle creator edge — # CCs stays the same.

n CCs → 1 CC

So we must add at least $n-1$ edges.

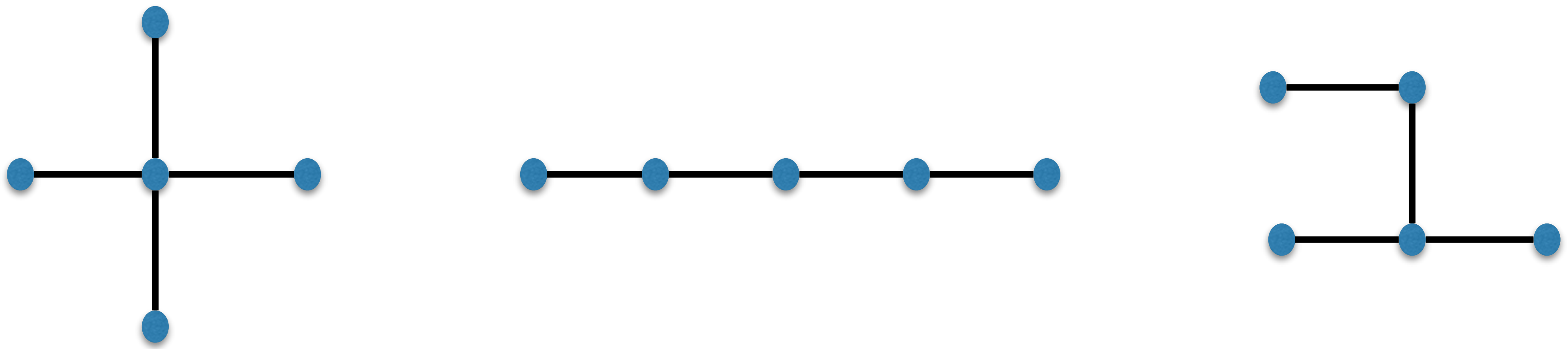
i.e. we must have $m \geq n-1$.

If $m = n-1$: all type (i) edges $\Longrightarrow$ no cycles.

If $m > n-1$: at least one type (ii) edge $\Longrightarrow$ a cycle. $\square$

Trees

Some examples with 5 vertices



Definition:

An n-vertex **tree** is any graph with at least 2 of the following 3 properties:

(i) connected

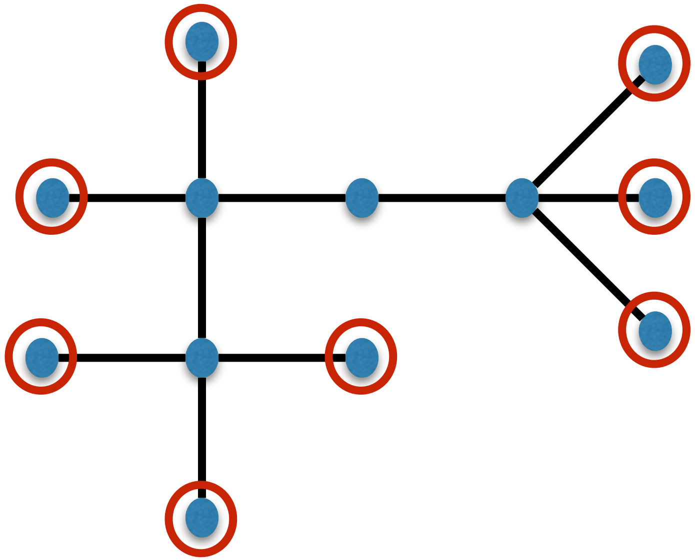
(ii) $m = n-1$

(iii) acyclic

Exercise:

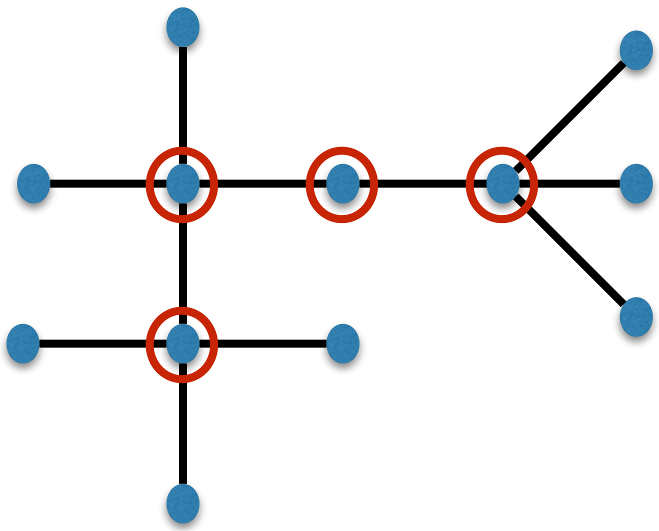
if it has two of the properties, it automatically has the third too.

Trees



Leaf: a vertex of degree 1

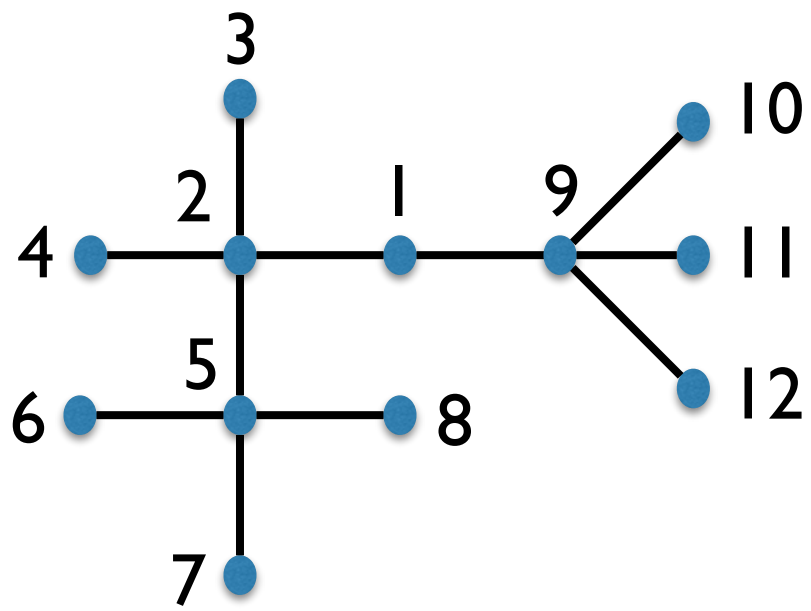
Trees



Leaf: a vertex of degree 1

Internal node: a vertex of degree > 1

Trees



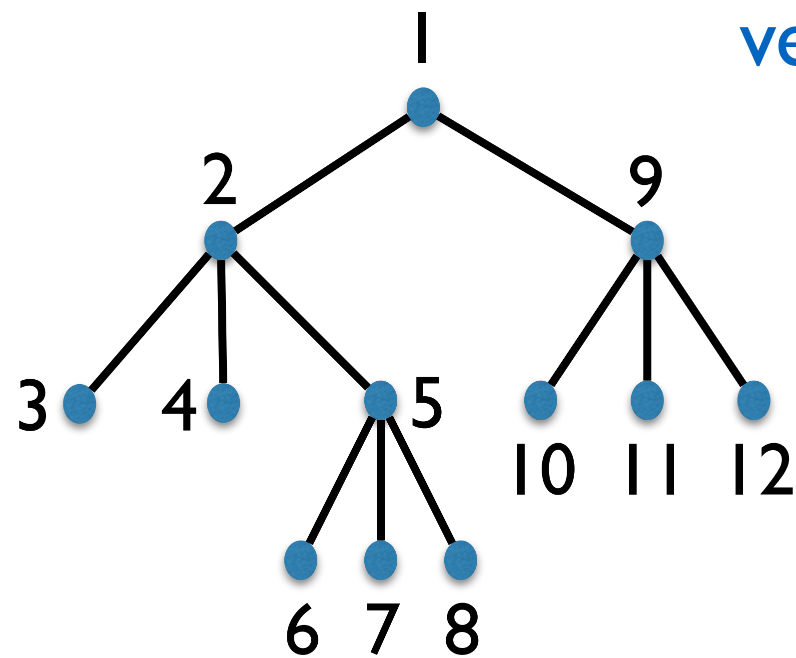
Leaf: a vertex of degree 1

Internal node: a vertex of degree > 1

Rooted tree: a tree with one vertex designated as “root”

Trees

vertex 1 is the root

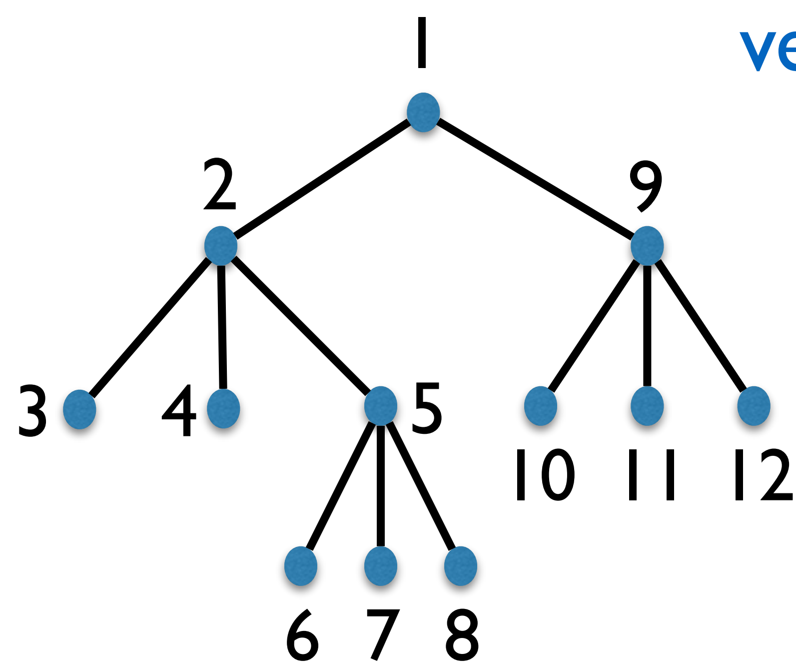


Leaf: a vertex of degree 1

Internal node: a vertex of degree > 1

Rooted tree: a tree with one vertex designated as "root"

Trees



vertex 1 is the root

For rooted trees, we use "*family tree*" terminology:

- parent
- child
- sibling

- ancestor
- descendant

etc…

Binary tree:

- rooted tree
- each node has at most 2 children.

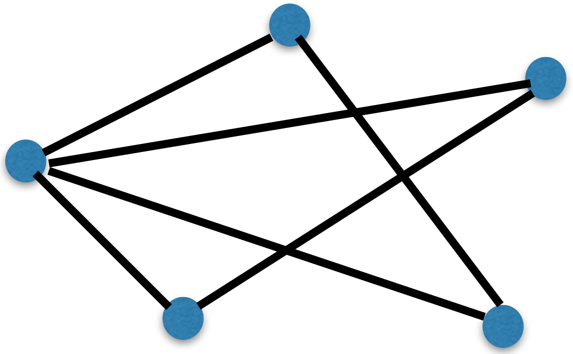
Back to Köningsberg's Bridges

Eulerian circuit

Eulerian Circuit Problem

Input: a graph G = (V, E)

Output: Yes if there is a circuit visiting each edge exactly once. No otherwise.

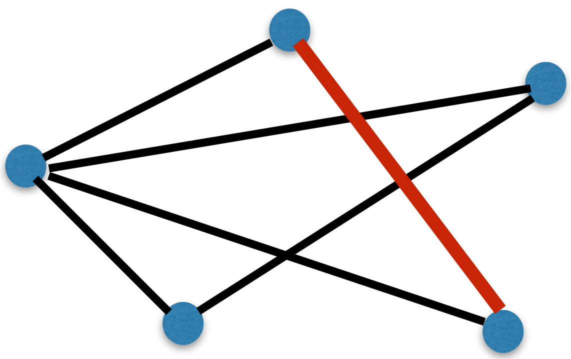


Eulerian circuit

Eulerian Circuit Problem

Input: a graph G = (V, E)

Output: Yes if there is a circuit visiting each edge exactly once. No otherwise.

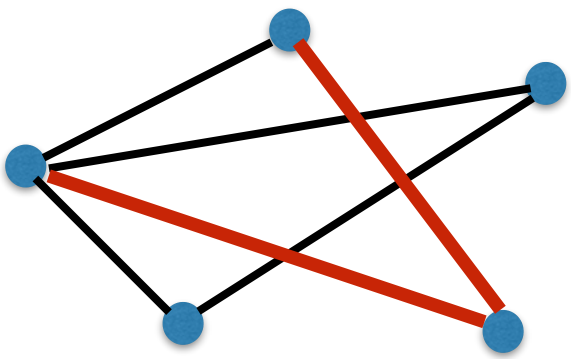


Eulerian circuit

Eulerian Circuit Problem

Input: a graph G = (V, E)

Output: Yes if there is a circuit visiting each edge exactly once. No otherwise.

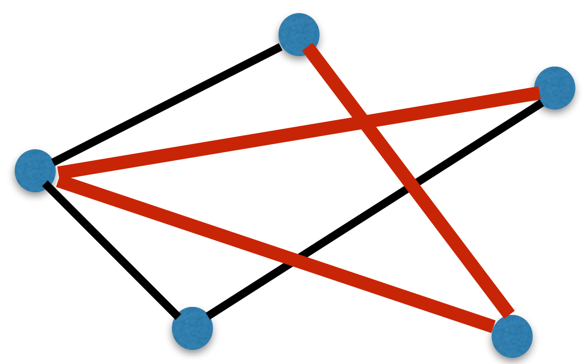


Eulerian circuit

Eulerian Circuit Problem

Input: a graph $G = (V, E)$

Output: Yes if there is a circuit visiting each edge exactly once. No otherwise.

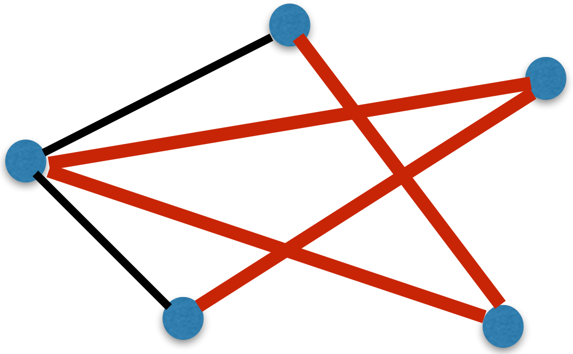


Eulerian circuit

Eulerian Circuit Problem

<u>Input</u>: a graph G = (V, E)

<u>Output</u>: Yes if there is a circuit visiting each edge exactly once. No otherwise.

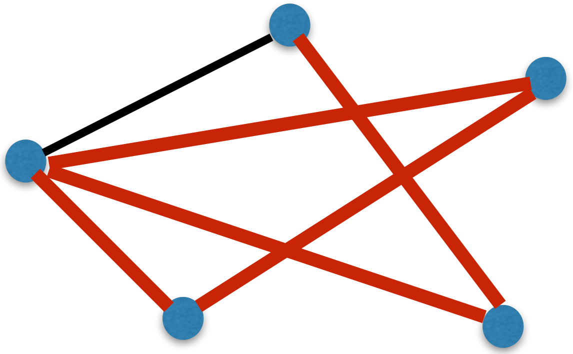


Eulerian circuit

Eulerian Circuit Problem

Input: a graph G = (V, E)

Output: Yes if there is a circuit visiting each edge exactly once. No otherwise.

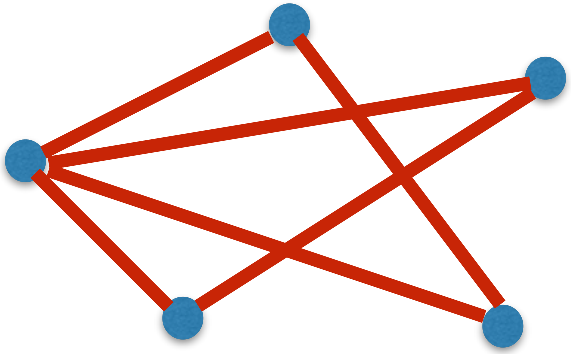


Eulerian circuit

Eulerian Circuit Problem

Input: a graph G = (V, E)

Output: Yes if there is a circuit visiting each edge exactly once. No otherwise.



Eulerian circuit

Eulerian Circuit Problem

Input: a graph G = (V, E)

Output: Yes if there is a circuit visiting each edge exactly once. No otherwise.

Euler claimed (but did not provide a proof):

A connected graph has an Eulerian circuit **iff** deg(v) is even for all v.

proved by Hierholzer

Efficient algorithm:

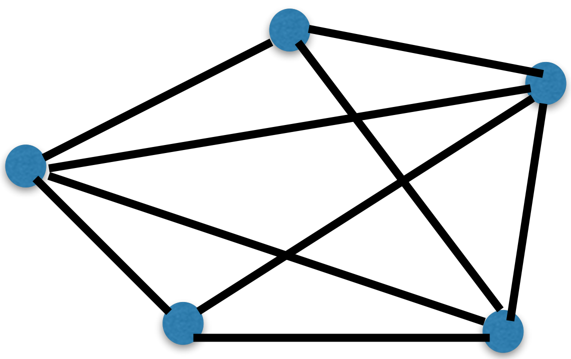
- Check that the graph is connected.
- Check that every vertex has even degree.

Hamiltonian cycle

Hamiltonian Cycle Problem

Input: a graph G = (V, E)

Output: Yes if there is a cycle visiting each **vertex** exactly once. No otherwise.

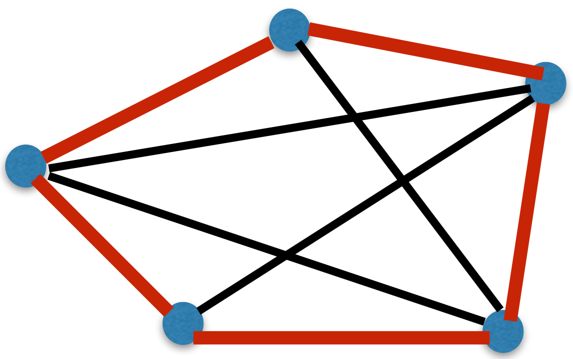


Hamiltonian cycle

Hamiltonian Cycle Problem

Input: a graph G = (V, E)

Output: Yes if there is a cycle visiting each **vertex** exactly once. No otherwise.



Hamiltonian cycle

Hamiltonian Cycle Problem

<u>Input</u>: a graph G = (V, E)

<u>Output</u>: Yes if there is a cycle visiting each **vertex** exactly once. No otherwise.

Brute-Force Algorithm:

- Try all cycles $O(n!)$

Dynamic Programming Algorithm: $O(2^n)$

Clever Algebraic Brute-Force: $O(1.657^n)$

Anything better?