Computational Combinatorics

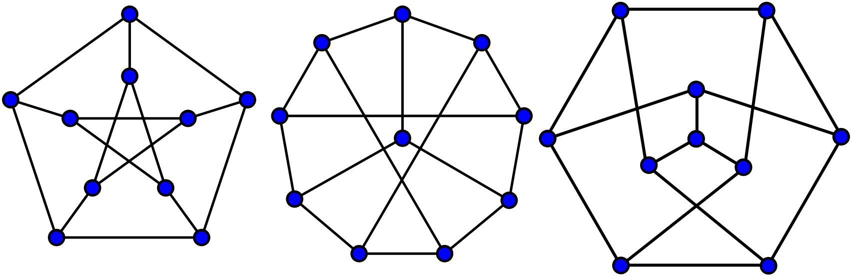
Derrick Stolee

Iowa State University
dstolee@iastate.edu
http://www.math.iastate.edu/dstolee/

November 3 & 5, 2014
Math 101

Combinatorial Object: Graphs

A **graph** G of **order** n is composed of a set $V(G)$ of n vertices and a set $E(G)$ of edges, where the edges are unordered pairs of vertices.



Two Flavors of Graph Theory

Structural Graph Theory:

Two Flavors of Graph Theory

Structural Graph Theory:

What conditions guarantee that certain **substructures** exist?

Two Flavors of Graph Theory

Structural Graph Theory:

What conditions guarantee that certain **substructures** exist?

Extremal Graph Theory:

Two Flavors of Graph Theory

Structural Graph Theory:

What conditions guarantee that certain **substructures** exist?

Extremal Graph Theory:

Given some structure, what **size restrictions** are guaranteed?

TONCAS Theorems

T

O

N

C

A

S

TONCAS Theorems

The

O

N

C

A

S

TONCAS Theorems

The

Obvious

N

C

A

S

TONCAS Theorems

The

Obvious

Neccessary

C

A

S

TONCAS Theorems

The

Obvious

Neccessary

Condition

A

S

TONCAS Theorems

The

Obvious

Neccessary

Condition
is
A

S

TONCAS Theorems

The

Obvious

Neccessary

Condition
is
Also

S

TONCAS Theorems

The

Obvious

Neccessary

Condition
is
Also

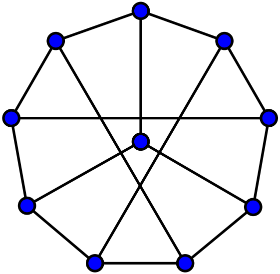
Sufficient

Connectedness

A graph is **connected** if for every pair u, v of vertices in G there exists a **path** from u to v in G.

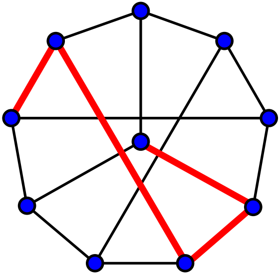
Connectedness

A graph is **connected** if for every pair u, v of vertices in G there exists a **path** from u to v in G.



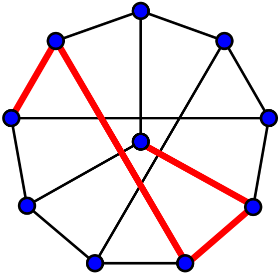
Connectedness

A graph is **connected** if for every pair u, v of vertices in G there exists a **path** from u to v in G.



Connectedness

A graph is **connected** if for every pair u, v of vertices in G there exists a **path** from u to v in G.



Think of the "6-Degrees of Kevin Bacon" game, played on the IMDB graph.

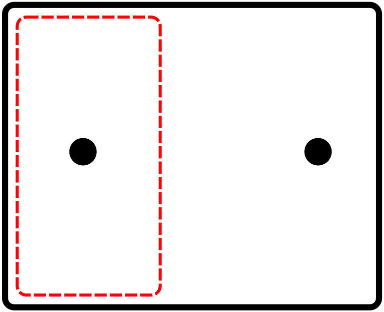
Connectedness : A TONCAS Theorem

A graph is **connected** if for every pair u, v of vertices in G there exists a **path** from u to v in G.

Connectedness : A TONCAS Theorem

A graph is **connected** if for every pair u, v of vertices in G there exists a **path** from u to v in G.

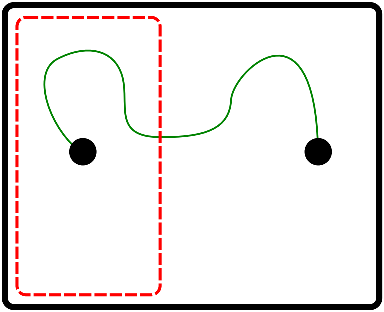
Theorem A graph G is connected **if and only if** for every set $S \subseteq V(G)$ where $S \neq \varnothing$ and $S \neq V(G)$ there exists at least one edge $uv \in E(G)$ where $u \in S$ and $v \notin S$.



Connectedness : A TONCAS Theorem

A graph is **connected** if for every pair u, v of vertices in G there exists a **path** from u to v in G.

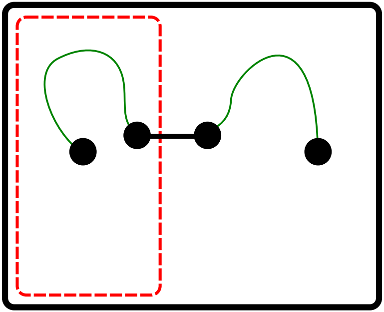
Theorem A graph G is connected **if and only if** for every set $S \subseteq V(G)$ where $S \neq \varnothing$ and $S \neq V(G)$ there exists at least one edge $uv \in E(G)$ where $u \in S$ and $v \notin S$.



Connectedness : A TONCAS Theorem

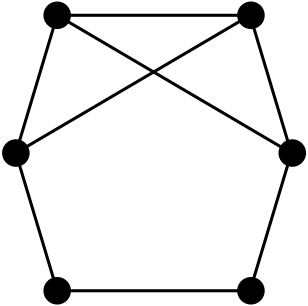
A graph is **connected** if for every pair u, v of vertices in G there exists a **path** from u to v in G.

Theorem A graph G is connected **if and only if** for every set $S \subseteq V(G)$ where $S \neq \varnothing$ and $S \neq V(G)$ there exists at least one edge $uv \in E(G)$ where $u \in S$ and $v \notin S$.



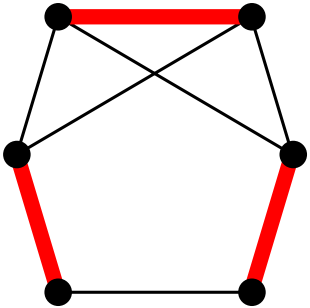
Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.



Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.

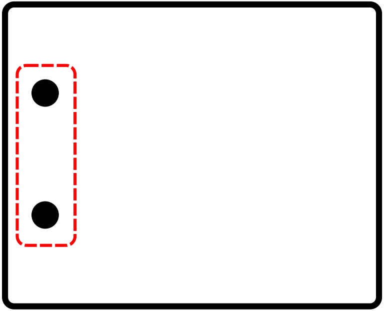


TONCAS for Perfect Matchings

Theorem (Tutte's Theorem) A graph has a perfect matching if and only if there is no set $S \subseteq V(G)$ so that the number of odd components in $G - S$ is greater than $|S|$.

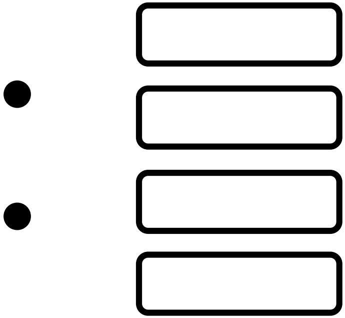
TONCAS for Perfect Matchings

Theorem (Tutte's Theorem) A graph has a perfect matching if and only if there is no set $S \subseteq V(G)$ so that the number of odd components in $G - S$ is greater than $|S|$.



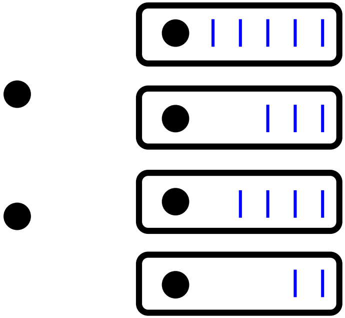
TONCAS for Perfect Matchings

Theorem (Tutte's Theorem) A graph has a perfect matching if and only if there is no set $S \subseteq V(G)$ so that the number of odd components in $G - S$ is greater than $|S|$.



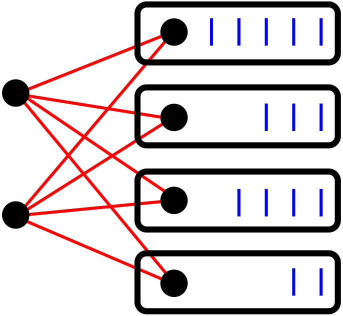
TONCAS for Perfect Matchings

Theorem (Tutte's Theorem) A graph has a perfect matching if and only if there is no set $S \subseteq V(G)$ so that the number of odd components in $G - S$ is greater than $|S|$.



TONCAS for Perfect Matchings

Theorem (Tutte's Theorem) A graph has a perfect matching if and only if there is no set $S \subseteq V(G)$ so that the number of odd components in $G - S$ is greater than $|S|$.



Extremal Graph Theory

Given specified **structure**, determine bounds on **size**.

Extremal Graph Theory

Given specified **structure**, determine bounds on **size**.

"Size" usually means "number of edges"

Extremal Graph Theory

Given specified **structure**, determine bounds on **size**.

“Size” usually means “number of edges”

Another perspective: Specific values of one parameter influence the value of another.

Turán's Theorem

An ***r*-clique** is a set of r vertices that are pairwise adjacent.

Turán's Theorem

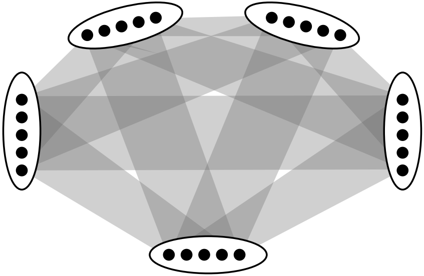
An ***r*-clique** is a set of r vertices that are pairwise adjacent.

Theorem (Turán's Theorem) If G is a graph of order n and has no r-clique, then G has at most $\approx (1 - \frac{1}{r-1})\frac{n^2}{2}$ edges.

Turán's Theorem

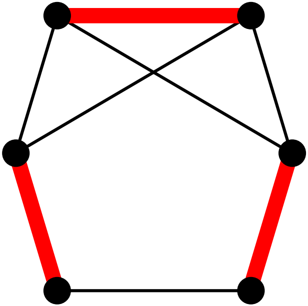
An ***r*-clique** is a set of r vertices that are pairwise adjacent.

Theorem (Turán's Theorem) If G is a graph of order n and has no r-clique, then G has at most $\approx \left(1 - \frac{1}{r-1}\right)\frac{n^2}{2}$ edges.



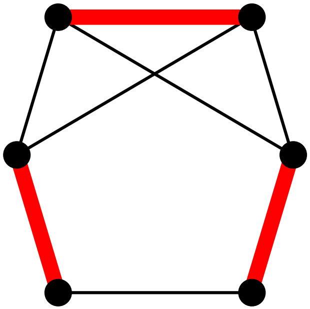
Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.



Perfect Matchings

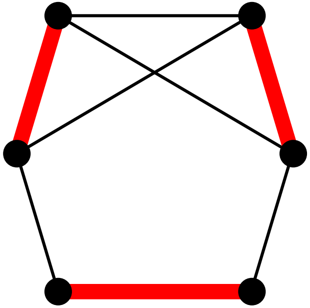
A **perfect matching** is a set of edges which cover each vertex exactly once.
$\Phi(G)$ is the number of perfect matchings in the graph G.



Perfect Matchings

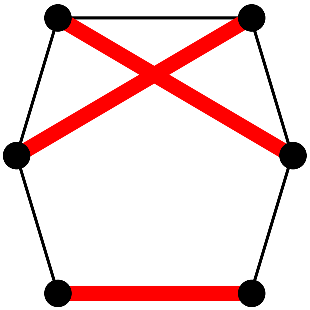
A **perfect matching** is a set of edges which cover each vertex exactly once.

$\Phi(G)$ is the number of perfect matchings in the graph G.



Perfect Matchings

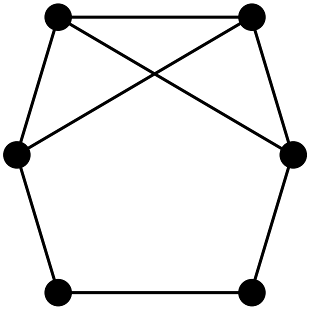
A **perfect matching** is a set of edges which cover each vertex exactly once.
$\Phi(G)$ is the number of perfect matchings in the graph G.



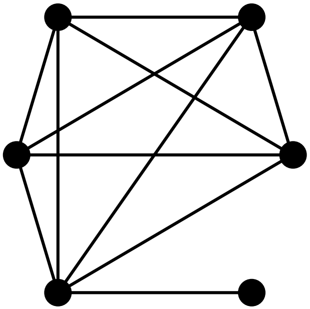
$\Phi(G) = 3$
8 edges

Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.



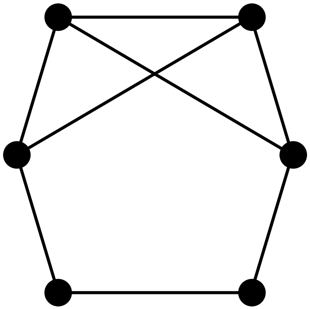
$\Phi(G) = 3$
8 edges



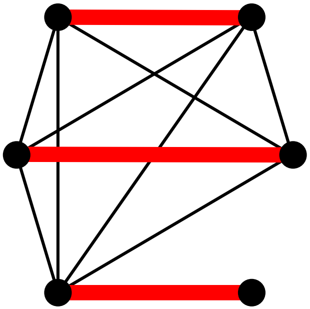
11 edges

Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.



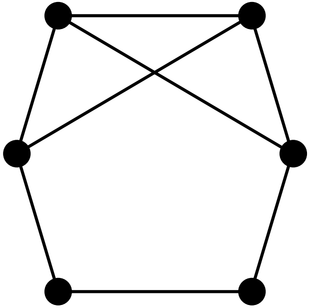
$\Phi(G) = 3$
8 edges



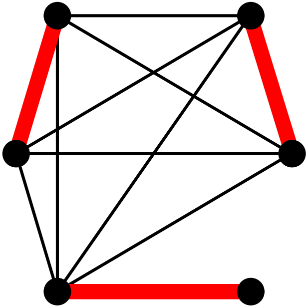
11 edges

Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.



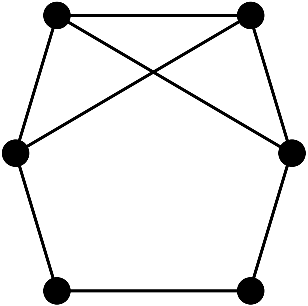
$\Phi(G) = 3$
8 edges



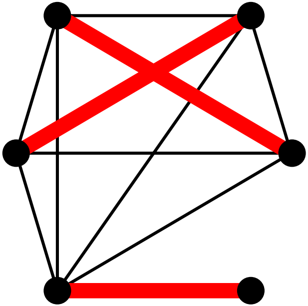
11 edges

Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.



$\Phi(G) = 3$
8 edges



$\Phi(G) = 3$
11 edges

Perfect Matchings

A **perfect matching** is a set of edges which cover each vertex exactly once.

Question (Dudek, Schmitt, 2010) What is the maximum number of edges in a graph with exactly n vertices and p perfect matchings?

Definition Let n be an even number and fix $p \geq 1$.

$$f(n, p) = \max\{|E(G)| : |V(G)| = n, \Phi(G) = p\}.$$

Graphs attaining this number of edges are ***p*-extremal**.

Hetyei's Theorem

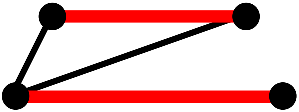
Theorem (Hetyei's Theorem, 1986) For all even $n \geq 2$,

$$f(n, 1) = \frac{n^2}{4}.$$

Hetyei's Theorem

Theorem (Hetyei's Theorem, 1986) For all even $n \geq 2$,

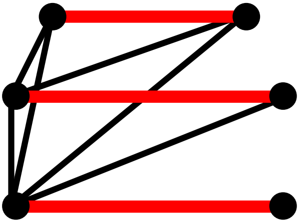
$$f(n, 1) = \frac{n^2}{4}.$$



Hetyei's Theorem

Theorem (Hetyei's Theorem, 1986) For all even $n \geq 2$,

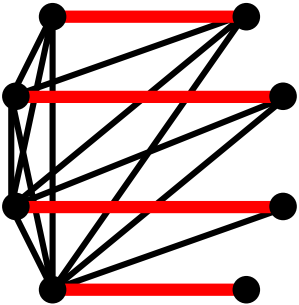
$$f(n, 1) = \frac{n^2}{4}.$$



Hetyei's Theorem

Theorem (Hetyei's Theorem, 1986) For all even $n \geq 2$,

$$f(n,1) = \frac{n^2}{4}.$$



The Form of $f(n, p)$

Theorem (Dudek, Schmitt, 2010) For each p, there exist constants n_p, c_p so that for all $n \geq n_p$,

$$f(n,p) = \frac{n^2}{4} + c_p.$$

The Form of $f(n,p)$

Theorem (Dudek, Schmitt, 2010) For each p, there exist constants n_p, c_p so that for all $n \geq n_p$,

$$f(n,p) = \frac{n^2}{4} + c_p.$$

p	1	2	3	4	5	6
c_p	0	1	2	2	2	3
	H	Dudek, Schmitt, 2010				

Structure Theorem

Theorem (Hartke, Stolee, West, Yancey, 2013) For a fixed p, every graph G with n vertices, p perfect matchings, and $f(n,p) = \frac{n^2}{4} + c_p$ edges is composed of a finite list of **fundamental graphs** combined in specified ways.

Proof involves several classic **structural theorems** from matching theory in an **extremal setting**.

Structure Theorem

Theorem (Hartke, Stolee, West, Yancey, 2013) For a fixed p, every graph G with n vertices, p perfect matchings, and $f(n,p) = \frac{n^2}{4} + c_p$ edges is composed of a finite list of **fundamental graphs** combined in specified ways.

Proof involves several classic **structural theorems** from matching theory in an **extremal setting**.

For $p \leq 10$, the graphs have order at most 12.

Structure Theorem

Theorem (Hartke, Stolee, West, Yancey, 2013) For a fixed p, every graph G with n vertices, p perfect matchings, and $f(n,p) = \frac{n^2}{4} + c_p$ edges is composed of a finite list of **fundamental graphs** combined in specified ways.

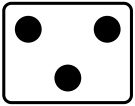
Proof involves several classic **structural theorems** from matching theory in an **extremal setting**.

For $p \leq 10$, the graphs have order at most 12.

We should “simply” list all graphs of this size and pick the best ones!

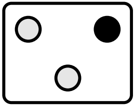
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.



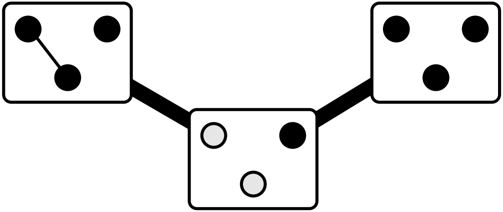
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.



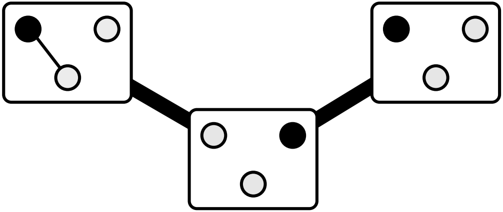
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.



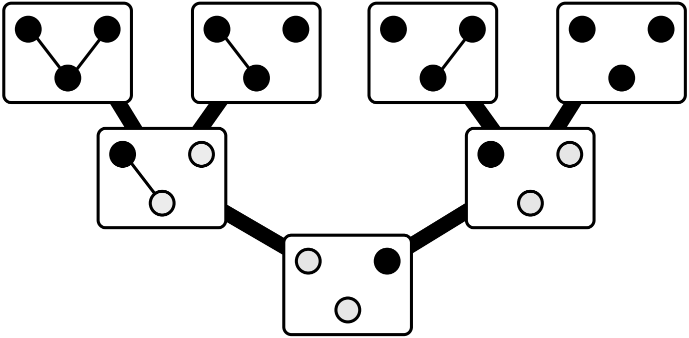
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.



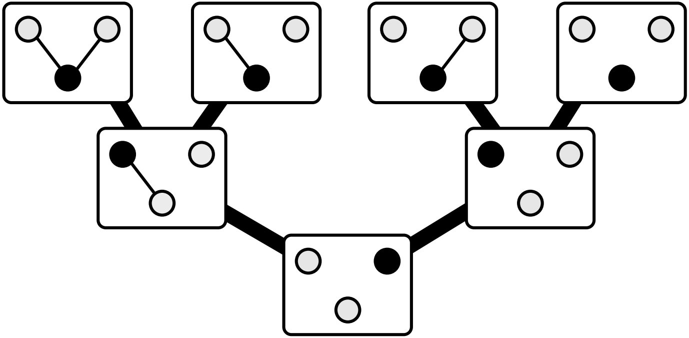
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.



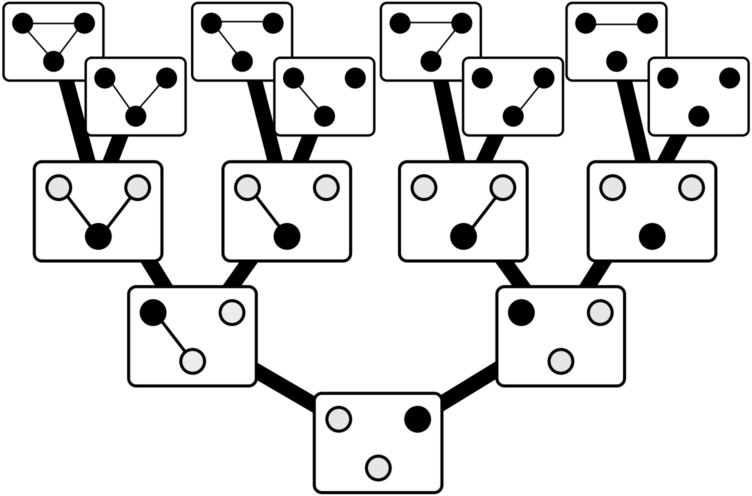
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.



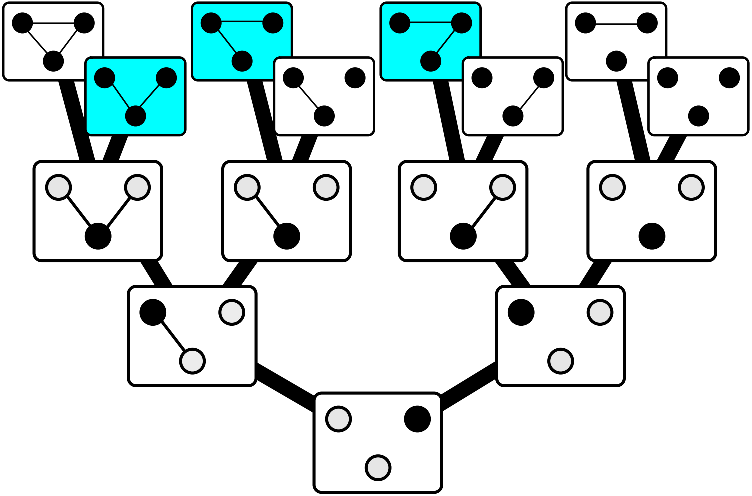
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.



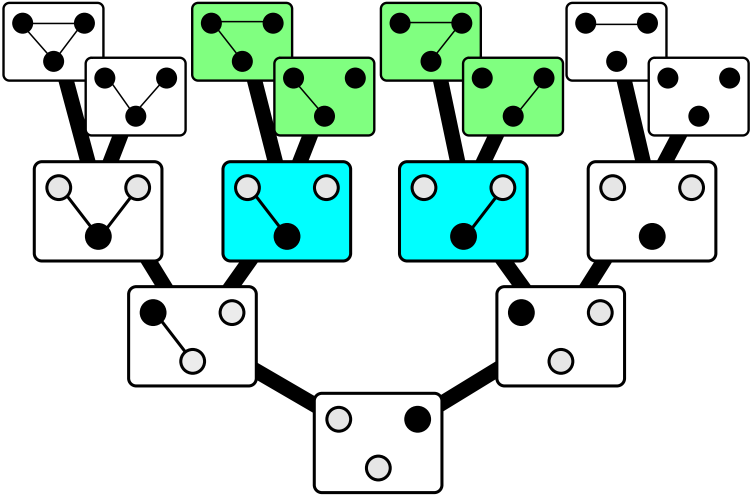
Example: Generating Graphs by Edges

We can build graphs starting at $\overline{K_n}$ by adding edges.

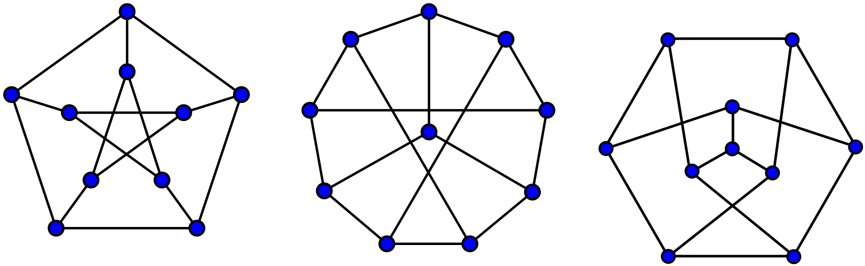


Example: Generating Graphs by Edges

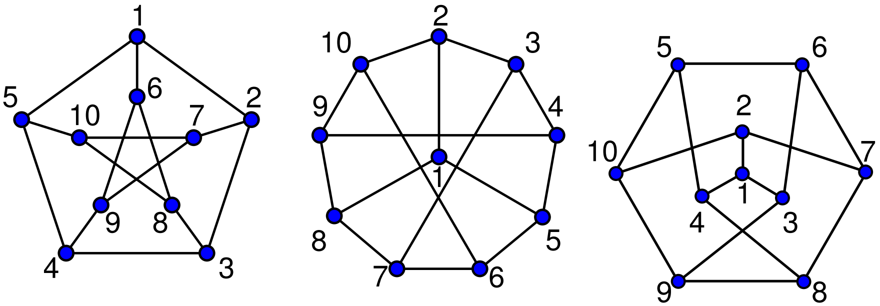
We can build graphs starting at $\overline{K_n}$ by adding edges.



Example: Generating Graphs by Edges

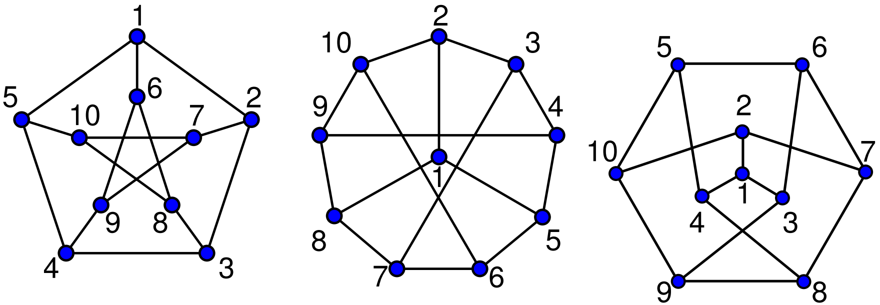


Example: Generating Graphs by Edges



Example: Generating Graphs by Edges

An **isomorphism** between G_1 and G_2 is a bijection from $V(G_1)$ to $V(G_2)$ that induces a bijection from $E(G_1)$ to $E(G_2)$.



Labeled Versus Unlabeled Objects

A **labeled** graph has a linear ordering on the vertices.

An **unlabeled** graph represents an isomorphism class of graphs.

Labeled Versus Unlabeled Objects

A **labeled** graph has a linear ordering on the vertices.

An **unlabeled** graph represents an isomorphism class of graphs.

Most interesting graph properties are **invariant under isomorphism**.

n	Number of unlabeled connected graphs of order n
2	1
3	2
4	5
5	19
6	85
7	509
8	4,060
9	41,301
10	510,489
11	7,319,447
12	117,940,535
13	2,094,480,864
14	40,497,138,011
15	845,480,228,069
16	1,894,152,284,590
17	453,090,162,062,723
18	11,523,392,072,541,432
19	310,467,244,165,539,782
20	8,832,736,318,937,756,165

OEIS Sequence A002851 Grows $2^{\Omega(n^2)}$.

n	Number of unlabeled connected graphs of order n
2	1
3	2
4	5
5	19
6	85
7	509
8	4,060
9	41,301
10	510,489
11	7,319,447
12	**117,940,535**
13	2,094,480,864
14	40,497,138,011
15	845,480,228,069
16	1,894,152,284,590
17	453,090,162,062,723
18	11,523,392,072,541,432
19	310,467,244,165,539,782
20	8,832,736,318,937,756,165

Requires about **1 day** of CPU Time.

n	Number of unlabeled connected graphs of order n
2	1
3	2
4	5
5	19
6	85
7	509
8	4,060
9	41,301
10	510,489
11	7,319,447
12	117,940,535
13	2,094,480,864
14	**40,497,138,011**
15	845,480,228,069
16	1,894,152,284,590
17	453,090,162,062,723
18	11,523,392,072,541,432
19	310,467,244,165,539,782
20	8,832,736,318,937,756,165

Requires over **1 year** of CPU Time.

Structure Theorem

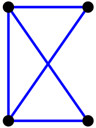
Theorem (Hartke, Stolee, West, Yancey, 2013) For a fixed p, every graph G with n vertices, p perfect matchings, and $f(n, p) = \frac{n^2}{4} + c_p$ edges is composed of a finite list of **fundamental graphs** combined in specified ways.

Proof involves several classic structure theorems from matching theory in an extremal setting.

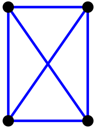
For $p \leq 10$, the graphs have order at most 12.

We should “simply” list all graphs of this size and pick the best ones!

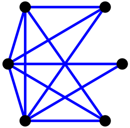
Fundamental Graphs for $2 \leq p \leq 10$



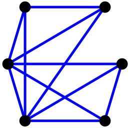
$p = 2$



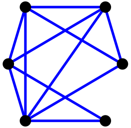
$p = 3$



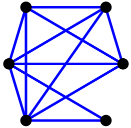
$p = 4$



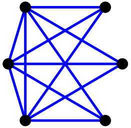
$p = 5$



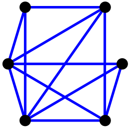
$p = 5$



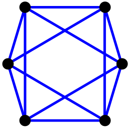
$p = 6$



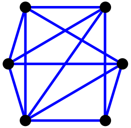
$p = 6$



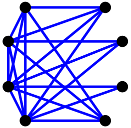
$p = 7$



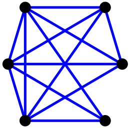
$p = 8$



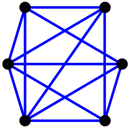
$p = 8$



$p = 8$



$p = 9$



$p = 10$

c_p for small p

p	1	2	3	4	5	6	7	8	9	10
c_p	0	1	2	2	2	3	3	3	4	4
	H	Dudek, Schmitt 2010					HSWY 2011			

c_p for small p

p	1	2	3	4	5	6	7	8	9	10
c_p	0	1	2	2	2	3	3	3	4	4
	H	Dudek, Schmitt 2010					HSWY 2011			

Q: Is c_p monotone in p?

Structural Theorem, Redux

Without more involved computational methods, brute force methods cannot go farther.

Structural Theorem, Redux

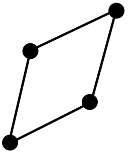
Without more involved computational methods, brute force methods cannot go farther.

The **Lovász Two Ear Theorem (1983)** provides a way to build fundamental graphs using **ear augmentations**.

Structural Theorem, Redux

Without more involved computational methods, brute force methods cannot go farther.

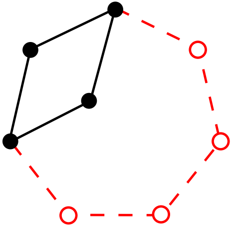
The **Lovász Two Ear Theorem (1983)** provides a way to build fundamental graphs using **ear augmentations**.



Structural Theorem, Redux

Without more involved computational methods, brute force methods cannot go farther.

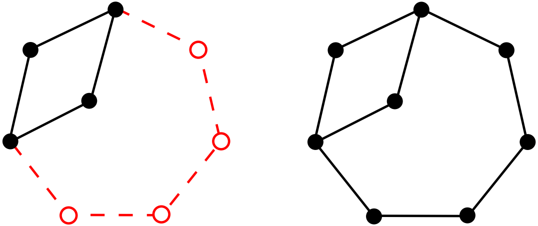
The **Lovász Two Ear Theorem (1983)** provides a way to build fundamental graphs using **ear augmentations**.



Structural Theorem, Redux

Without more involved computational methods, brute force methods cannot go farther.

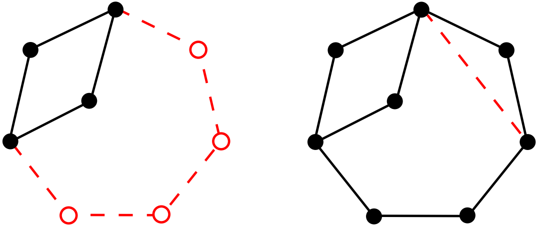
The **Lovász Two Ear Theorem (1983)** provides a way to build fundamental graphs using **ear augmentations**.



Structural Theorem, Redux

Without more involved computational methods, brute force methods cannot go farther.

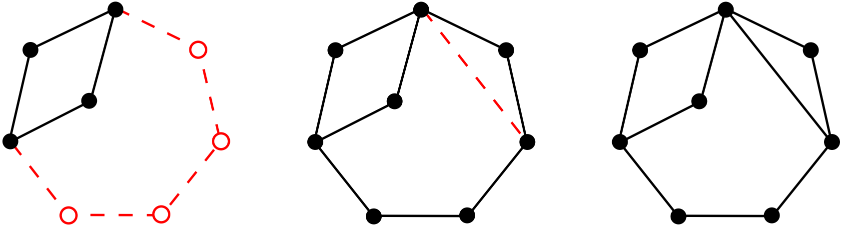
The **Lovász Two Ear Theorem (1983)** provides a way to build fundamental graphs using **ear augmentations**.



Structural Theorem, Redux

Without more involved computational methods, brute force methods cannot go farther.

The **Lovász Two Ear Theorem (1983)** provides a way to build fundamental graphs using **ear augmentations**.



Computational Method

Developed a computational method from:

1. **Augmentations:** Lovász Two Ear Theorem.
2. **Isomorphs:** McKay's Isomorph-Free Generation Method.
3. **Pruning:** Developed new structural and extremal theorems.

Computational Method

Developed a computational method from:

1. **Augmentations:** Lovász Two Ear Theorem.
2. **Isomorphs:** McKay's Isomorph-Free Generation Method.
3. **Pruning:** Developed new structural and extremal theorems.

Before: Stuck at $p \leq 10$ when searching on most 12 vertices.

Computational Method

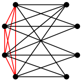
Developed a computational method from:

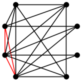
1. **Augmentations:** Lovász Two Ear Theorem.
2. **Isomorphs:** McKay's Isomorph-Free Generation Method.
3. **Pruning:** Developed new structural and extremal theorems.

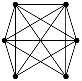
Before: Stuck at $p \leq 10$ when searching on most 12 vertices.

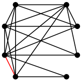
Now: Found graphs for all $p \leq 27$ on up to 22 vertices.

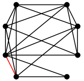
Fundamental Graphs for 11 $\leq p \leq$ 27

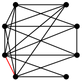

$p = 11$

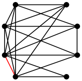

$p = 11$

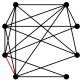

$p = 12$

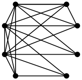

$p = 13$

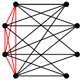

$p = 13$

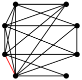

$p = 13$

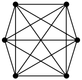

$p = 13$

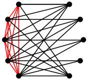

$p = 13$

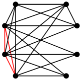

$p = 13$

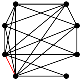

$p = 14$

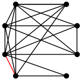

$p = 14$

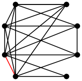

$p = 15$

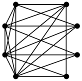

$p = 16$

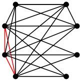

$p = 16$

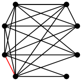

$p = 16$

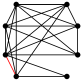

$p = 16$

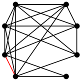

$p = 17$

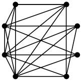

$p = 17$

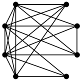

$p = 18$

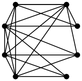

$p = 18$

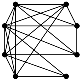

$p = 19$

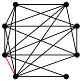

$p = 19$

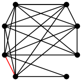

$p = 19$

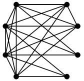

$p = 19$

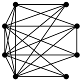

$p = 19$

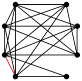

$p = 19$

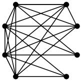

$p = 20$

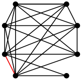

$p = 21$

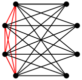

$p = 21$

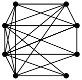

$p = 21$

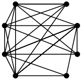

$p = 22$

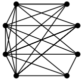

$p = 23$

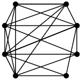

$p = 24$

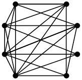

$p = 24$

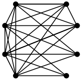

$p = 25$


$p = 25$


$p = 26$


$p = 26$


$p = 26$


$p = 27$

c_p for small p

p	1	2	3	4	5	6	7	8	9	10
c_p	0	1	2	2	2	3	3	3	4	4
	H	Dudek, Schmitt					HSWY			

p	11	12	13	14	15	16	17	18	19	20
c_p	3	5	3	4	6	4	4	5	4	5
	Stolee									

p	21	22	23	24	25	26	27
c_p	5	5	5	6	5	5	6
	Stolee						

c_p for small p

p	1	2	3	4	5	6	7	8	9	10
c_p	**0**	**1**	**2**	**2**	**2**	**3**	**3**	**3**	**4**	**4**
	H	Dudek, Schmitt					HSWY			

p	11	12	13	14	15	16	17	18	19	20
c_p	3	**5**	3	4	**6**	4	4	5	4	5
	Stolee									

p	21	22	23	24	25	26	27
c_p	5	5	5	**6**	5	5	**6**
	Stolee						

c_p not monotonic in p !

Blue numbers match conjectured upper bound.

What can you do?

Does this sound interesting? Here are some things **you** can do:

What can you do?

Does this sound interesting? Here are some things **you** can do:

1. Take MATH 304 (Combinatorics) and/or 314 (Graph Theory)

What can you do?

Does this sound interesting? Here are some things **you** can do:

1. Take MATH 304 (Combinatorics) and/or 314 (Graph Theory)
2. Participate in an REU!

What can you do?

Does this sound interesting? Here are some things **you** can do:

1. Take MATH 304 (Combinatorics) and/or 314 (Graph Theory)
2. Participate in an REU!
3. Participate in the Discrete Mathematics Seminar:
   `http://orion.math.iastate.edu/dept/seminar/dmseminar.htm`

Computational Combinatorics

Derrick Stolee

Iowa State University
dstolee@iastate.edu
http://www.math.iastate.edu/dstolee/

November 3 & 5, 2014
Math 101