Listing All Maximal Cliques in Sparse Graphs in Near-Optimal Time

Darren Strash
Department of Computer Science
UC Irvine

Joint work with David Eppstein and Maarten Löffler

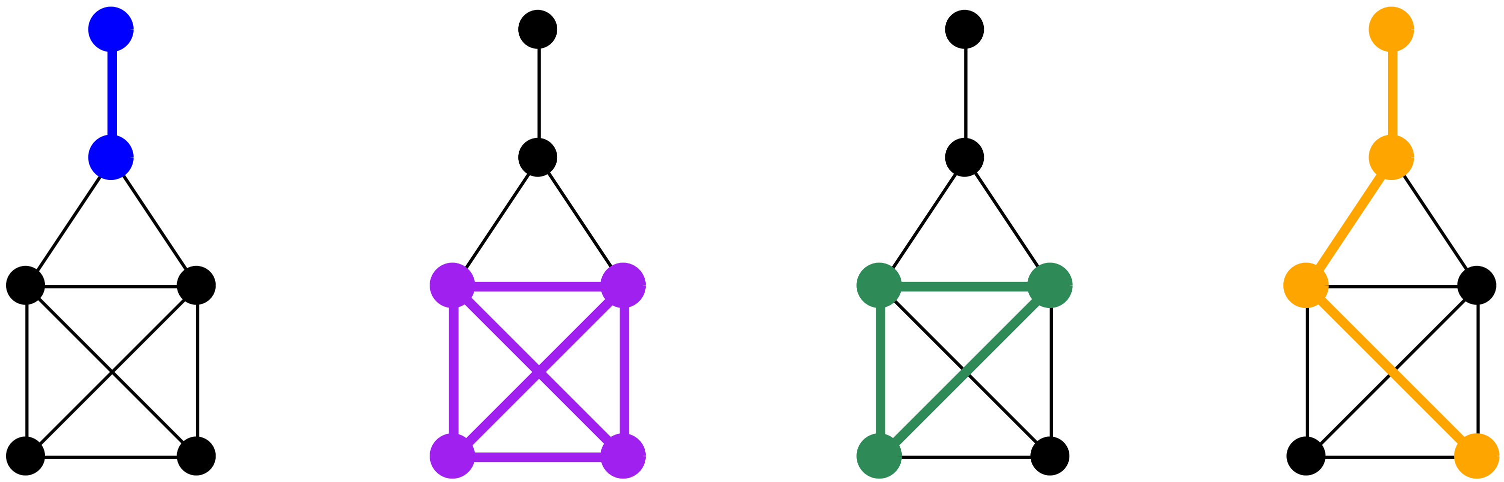
What is a Maximal Clique?

A clique that cannot be extended by adding more vertices

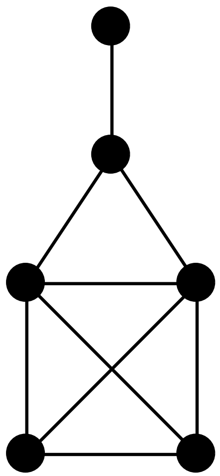
What is a Maximal Clique?

A clique that cannot be extended by adding more vertices

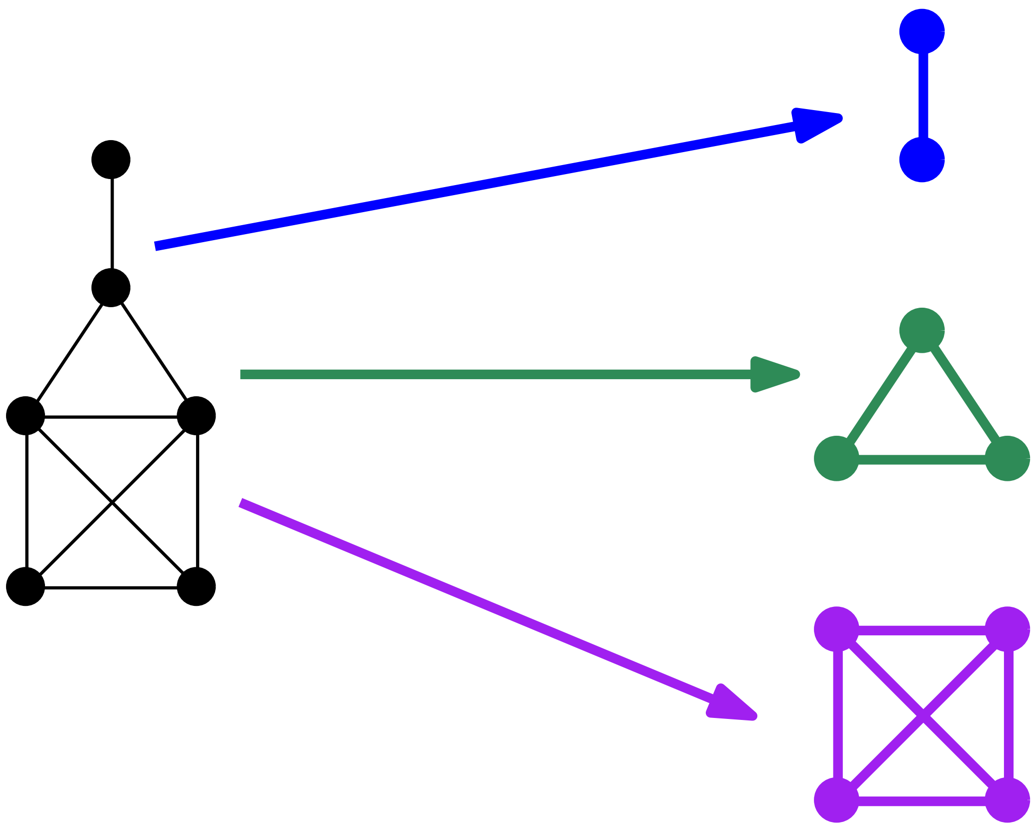
Maximal | Maximal, Maximum | Not Maximal | Not Clique



Goal: Design an algorithm to list all maximal cliques



Goal: Design an algorithm to list all maximal cliques



Motivation

Motivation

Features in ERGM

Motivation

Features in ERGM

Detect structural motifs from similarities between proteins

Motivation

Features in ERGM

Detect structural motifs from similarities between proteins

Determine the docking regions between biomolecules

Motivation

Features in ERGM

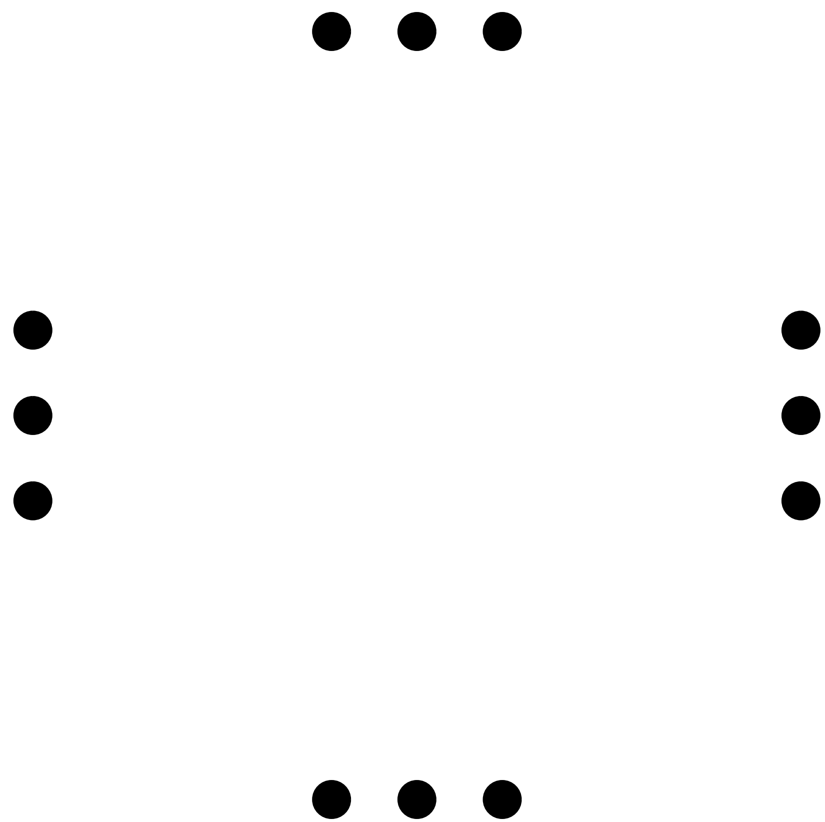
Detect structural motifs from similarities between proteins

Determine the docking regions between biomolecules

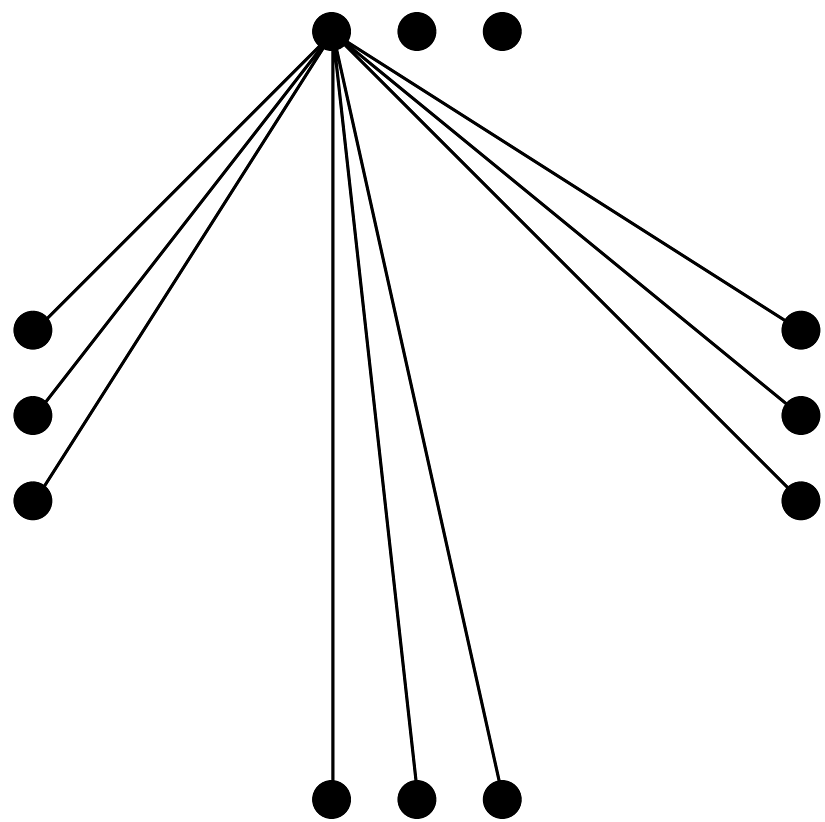
Document clustering for information retrieval

There may be many maximal cliques.

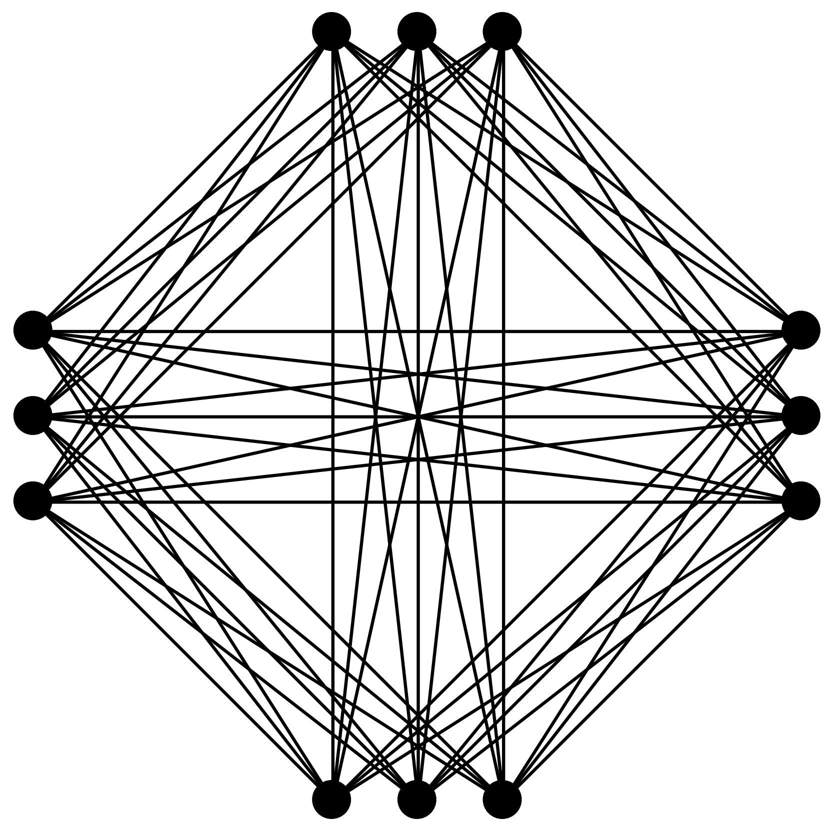
There may be many maximal cliques.



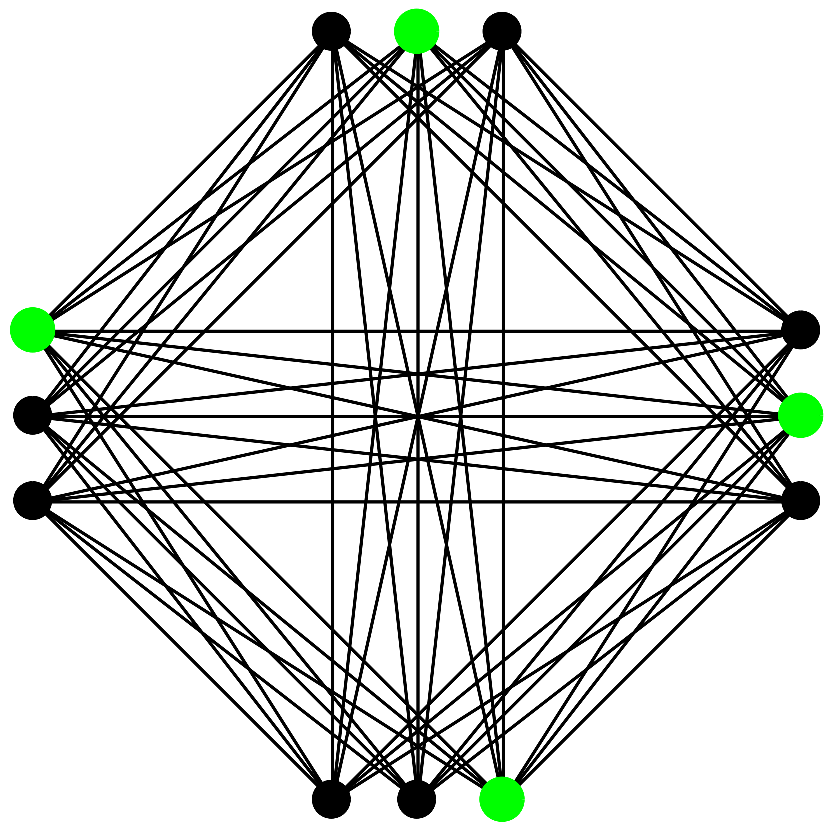
There may be many maximal cliques.



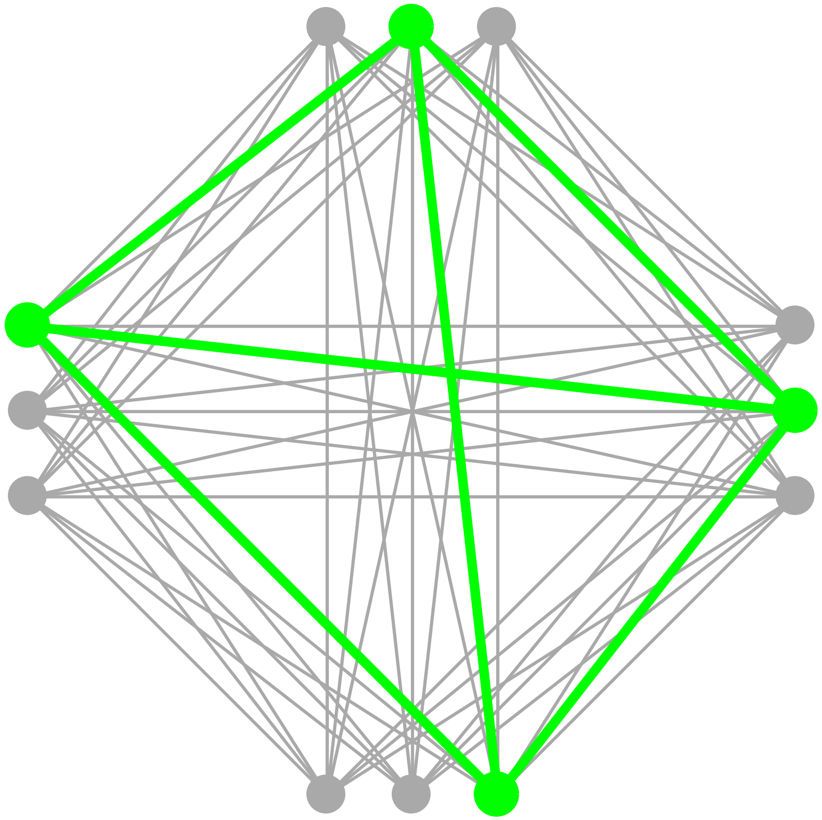
There may be many maximal cliques.



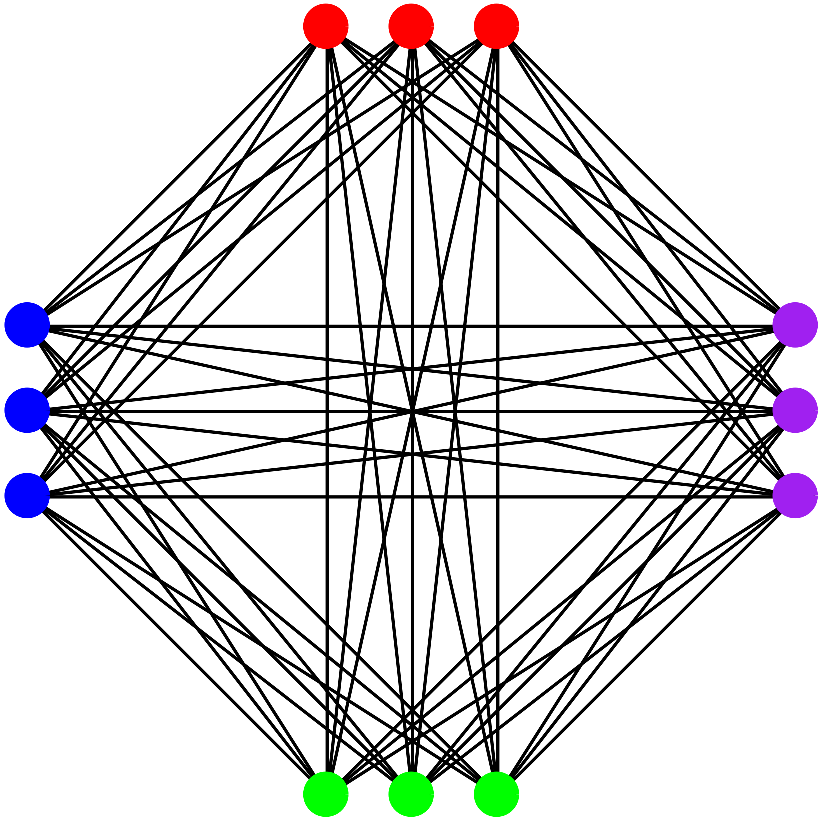
There may be many maximal cliques.



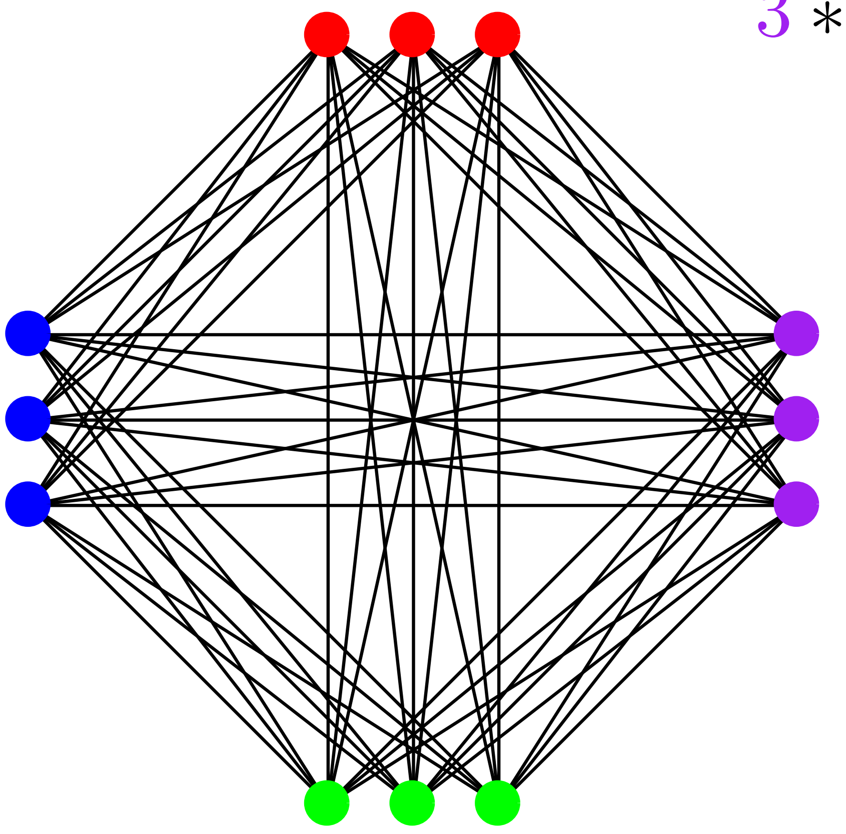
There may be many maximal cliques.



There may be many maximal cliques.

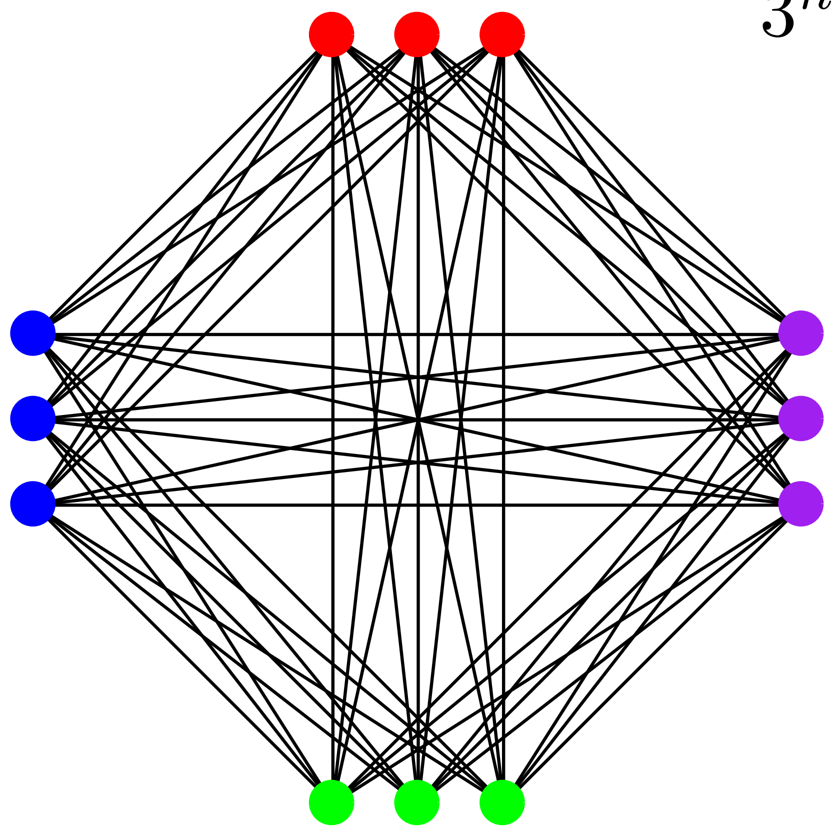


There may be many maximal cliques.



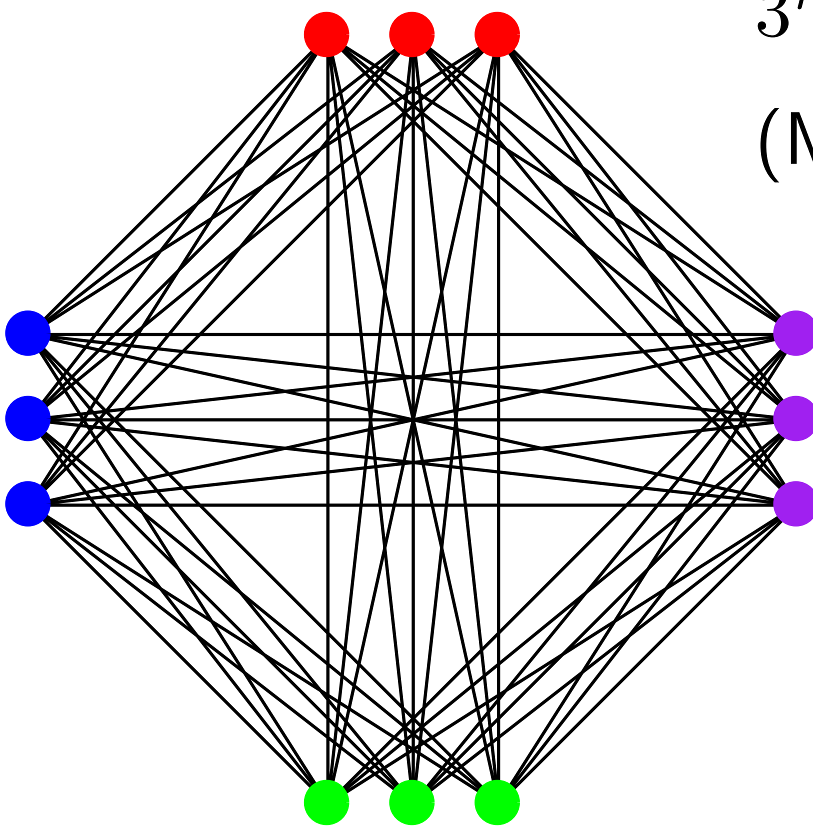
$3 * 3 * 3 * 3$

There may be many maximal cliques.



$3^{n/3}$ maximal cliques

There may be many maximal cliques.



$3^{n/3}$ maximal cliques

(Moon–Moser bound)

Maximal Clique Listing Algorithms

Author	Year	Running Time
Bron and Kerbosch	1973	???
Tsukiyama et al.	1977	$O(nm\mu)$
Chiba and Nishizeki	1985	$O(\alpha m\mu)$
Makino and Uno	2004	$O(\Delta^4\mu)$

n = number of vertices
m = number of edges
μ = number of maximal cliques
α = arboricity
Δ = maximum degree of the graph

Tomita et al. (2006)

Tomita et al. (2006)

Worst-case optimal running time $O(3^{n/3})$

Tomita et al. (2006)

Worst-case optimal running time $O(3^{n/3})$

Computational experiments:

AMC [14]	AMC* [14]	CLIQUES
261.27	**9.51**	10.49
952.25	49.45	**10.20**
3,601.09	130.76	**9.90**
14,448.21	431.20	**10.95**
35,866.69	530.53	**12.97**
$>$ 24 h	1,066.62	**16.85**
$>$ 24 h	4,350.94	**33.75**
$>$ 24 h	15,655.05	**65.06**
$>$ 24 h	$>$ 24 h	**293.97**

Tomita et al. (2006)

Worst-case optimal running time $O(3^{n/3})$

Computational experiments:

slow

AMC [14]	AMC* [14]	CLIQUES
261.27	**9.51**	10.49
952.25	49.45	**10.20**
3,601.09	130.76	**9.90**
14,448.21	431.20	**10.95**
35,866.69	530.53	**12.97**
> 24 h	1,066.62	**16.85**
> 24 h	4,350.94	**33.75**
> 24 h	15,655.05	**65.06**
> 24 h	> 24 h	**293.97**

fast!

The Bron–Kerbosch Algorithm

Easy to understand

Easy to implement

There are many heuristics, which make it faster

Its variations work well in practice.

Confirmed through computational experiments
Johnston (1976), Koch (2001), Baum (2003)

One variation is worst-case optimal ($O(3^{n/3})$ time)
Tomita et al. (2006)

The Bron–Kerbosch Algorithm

Easy to understand

Easy to implement

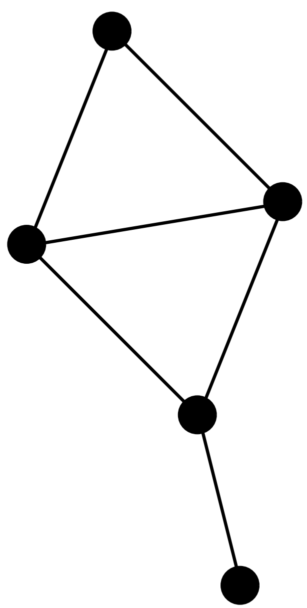
There are many heuristics, which make it faster

Its variations work well in practice.

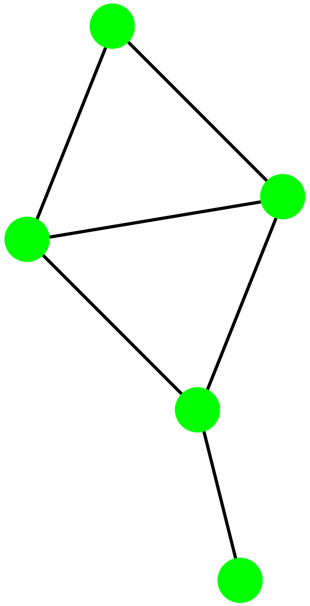
Confirmed through computational experiments
Johnston (1976), Koch (2001), Baum (2003)

One variation is worst-case optimal ($O(3^{n/3})$ time)
Tomita et al. (2006)

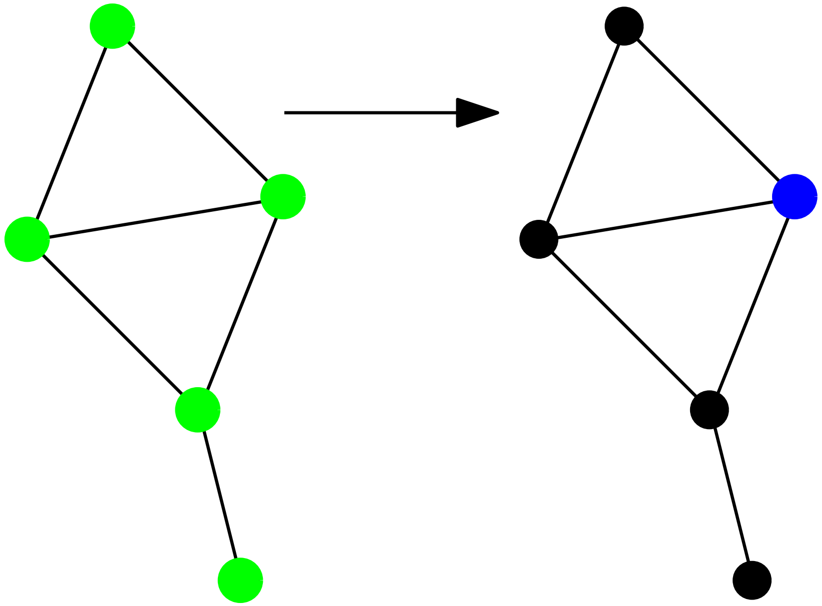
Finding one maximal clique



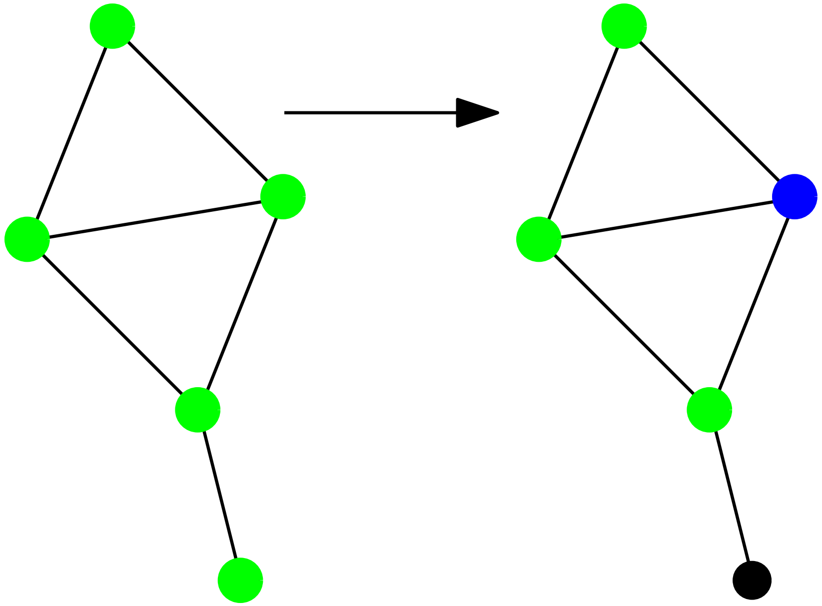
Finding one maximal clique



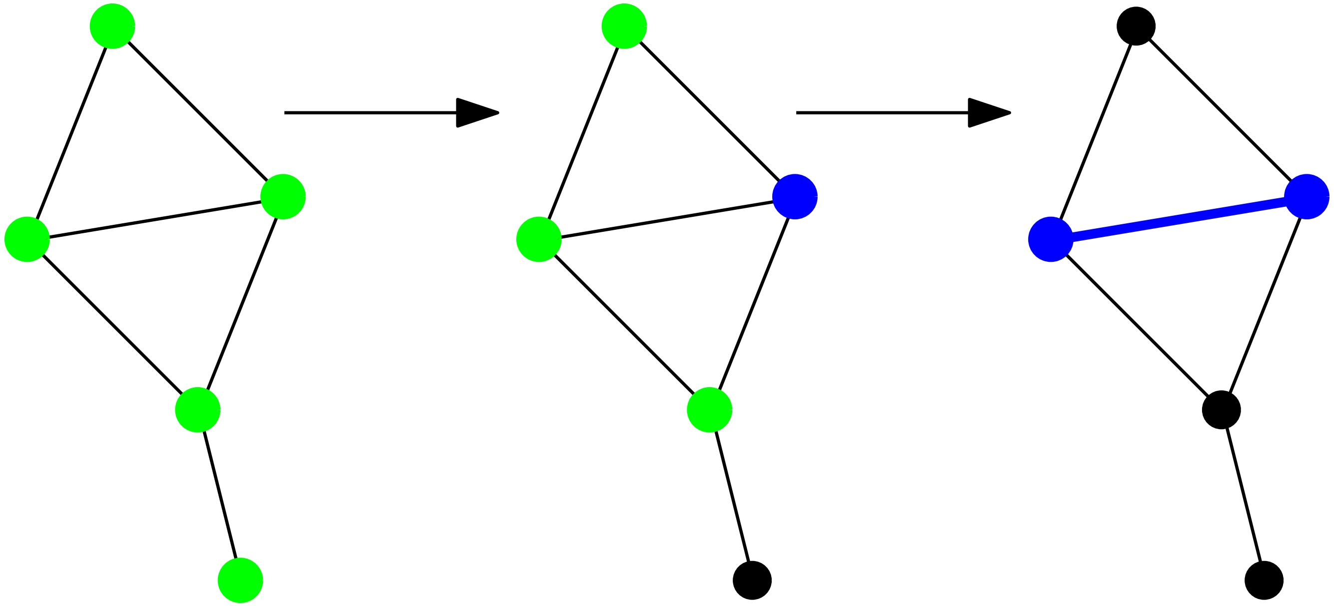
Finding one maximal clique



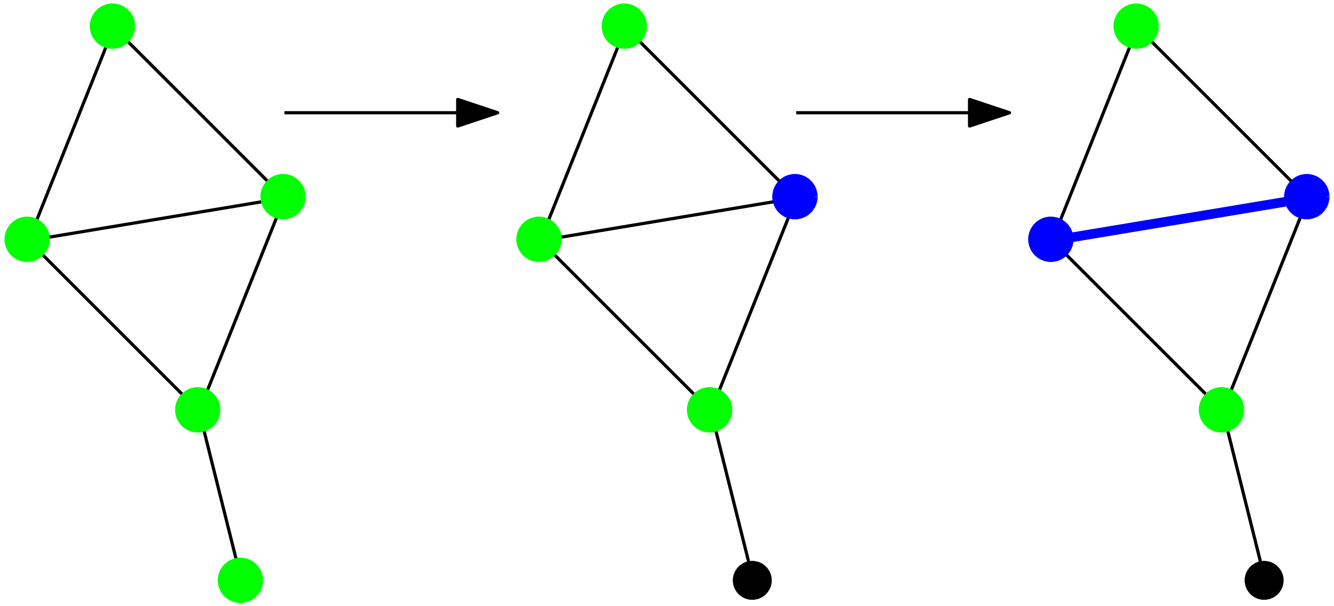
Finding one maximal clique



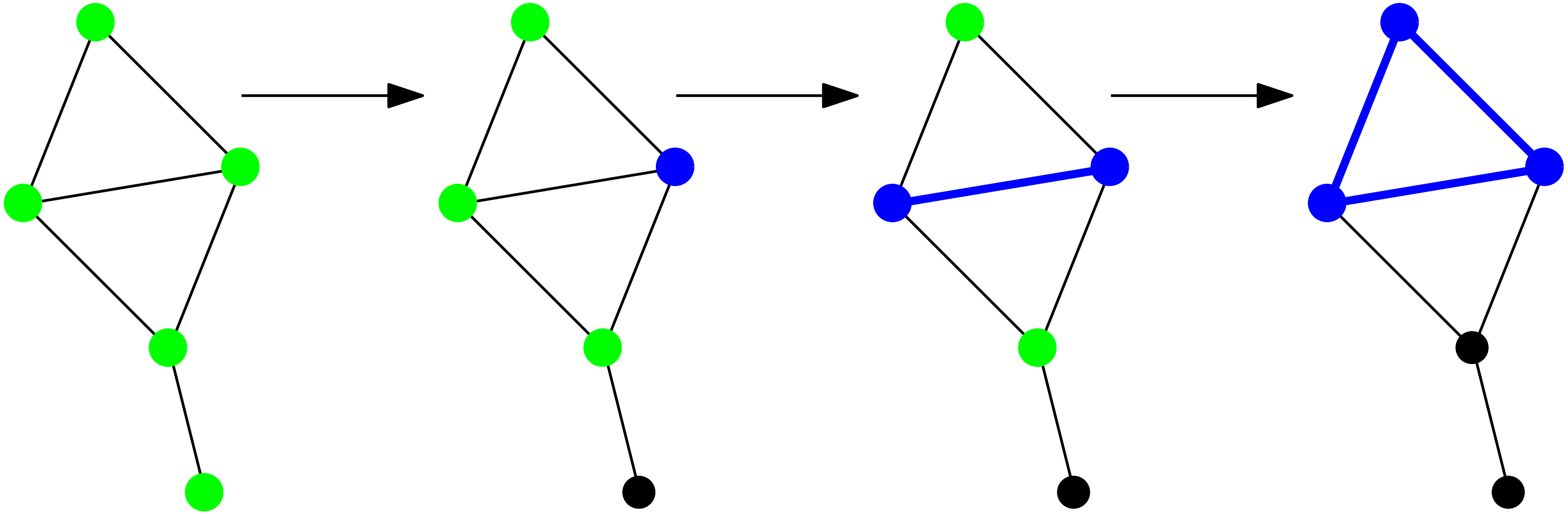
Finding one maximal clique



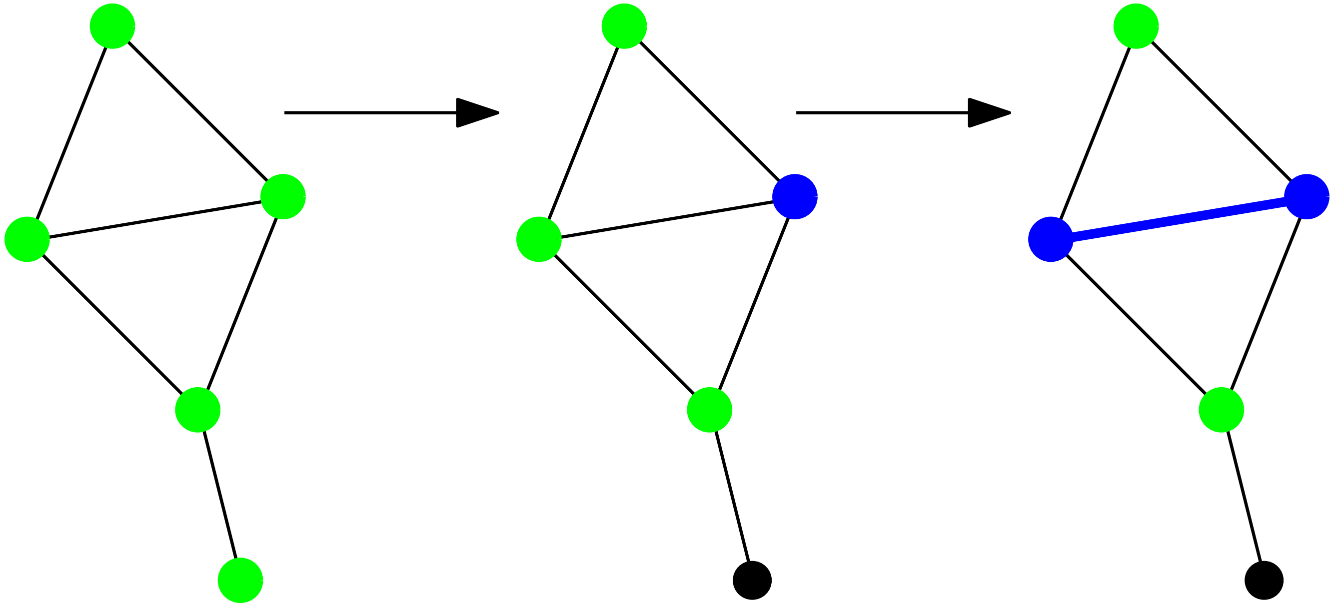
Finding one maximal clique



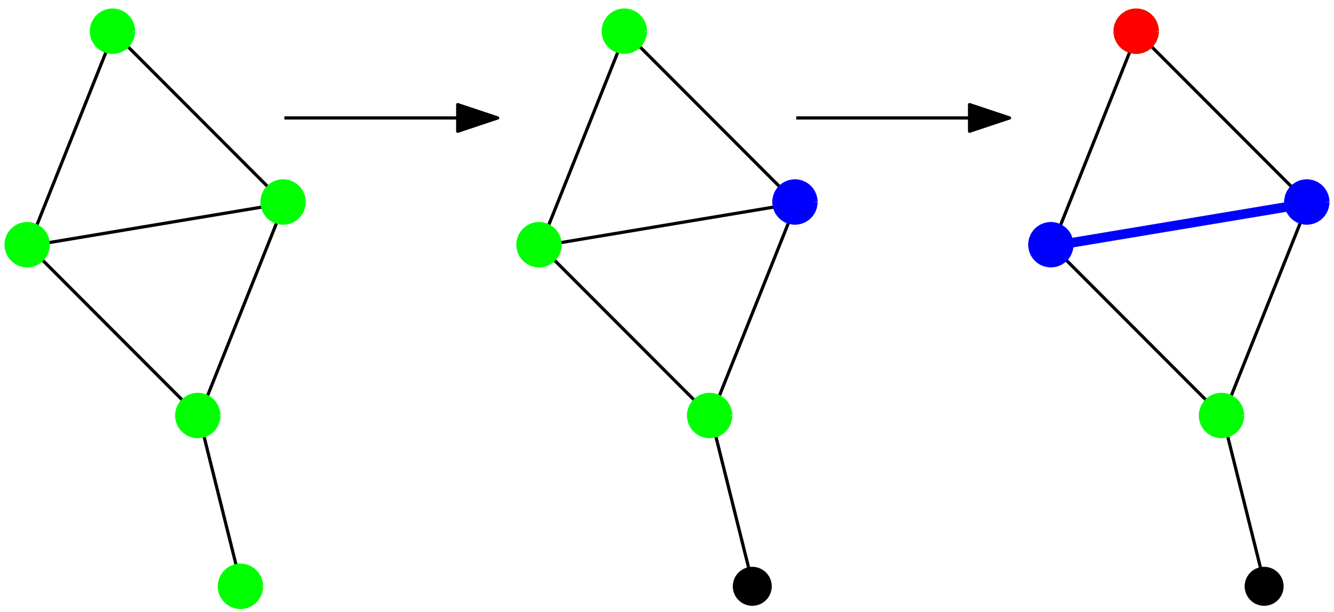
Finding one maximal clique



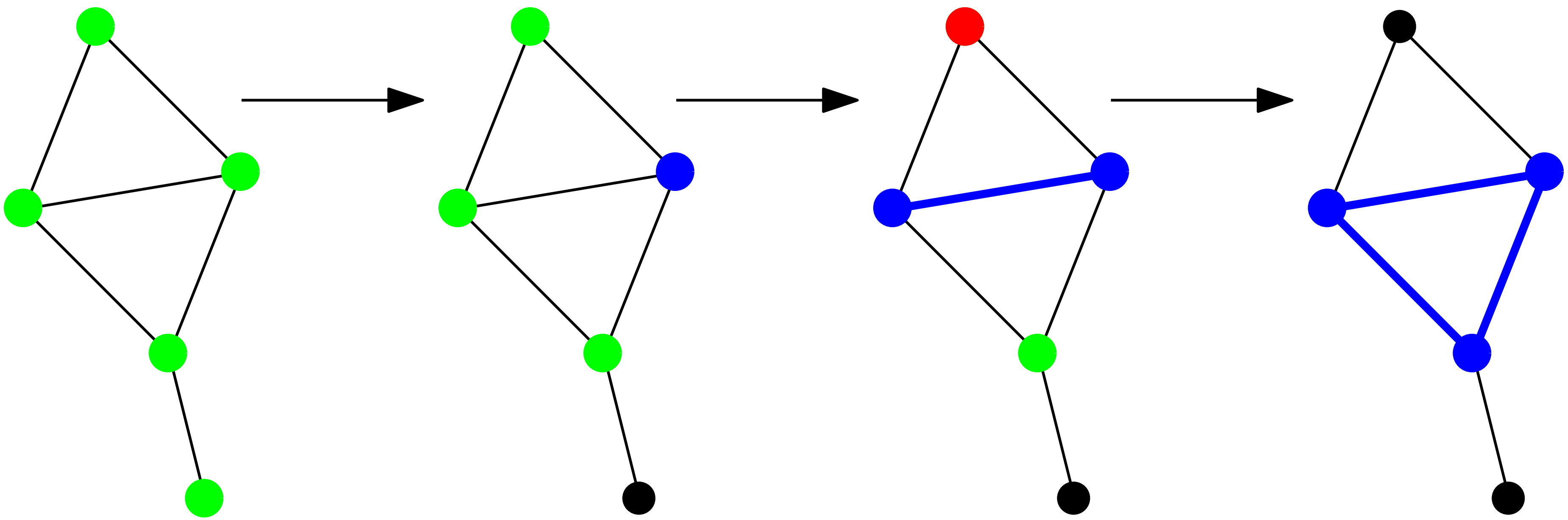
The Bron–Kerbosch Algorithm



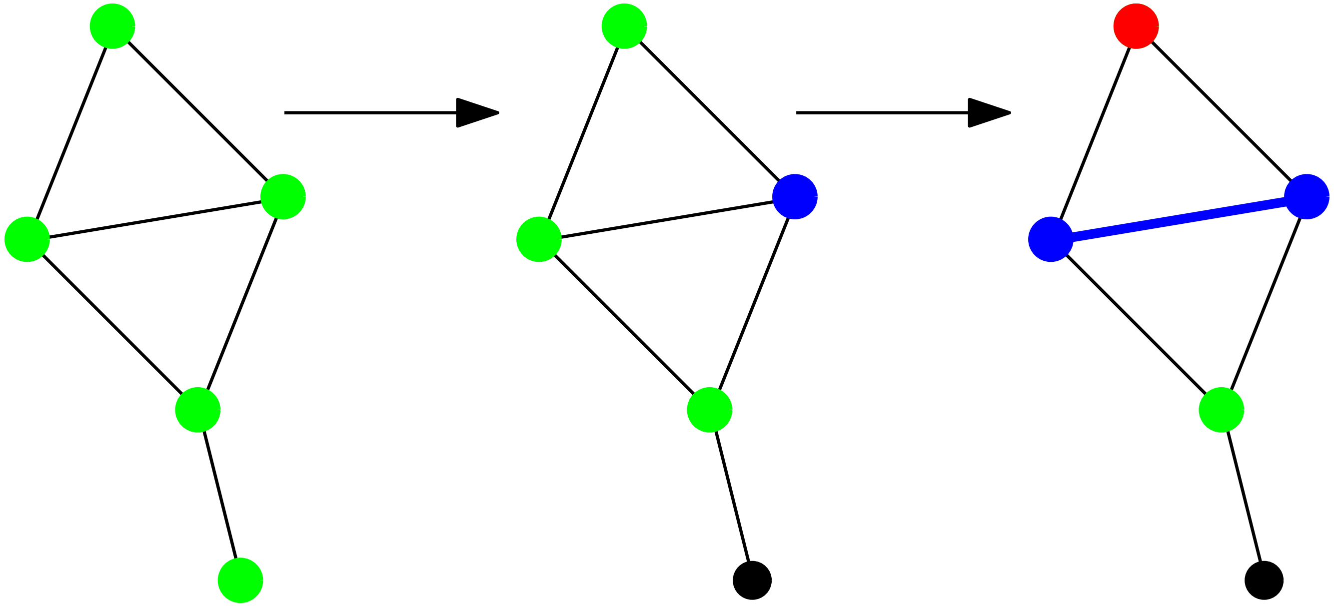
The Bron–Kerbosch Algorithm



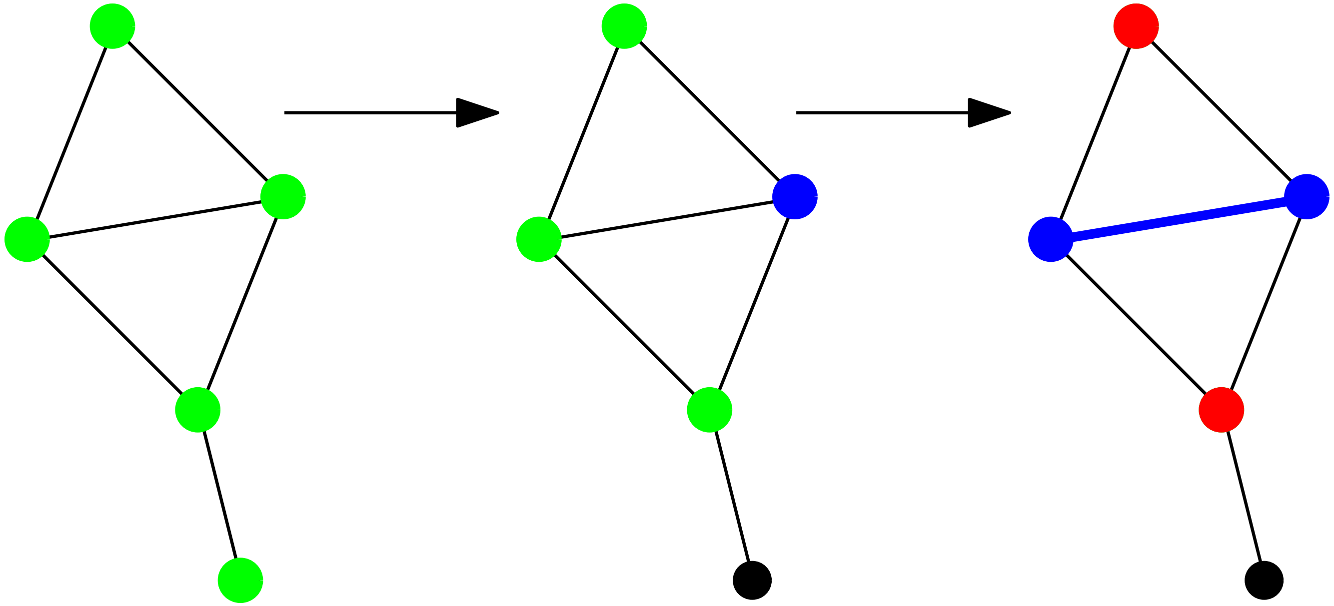
The Bron–Kerbosch Algorithm



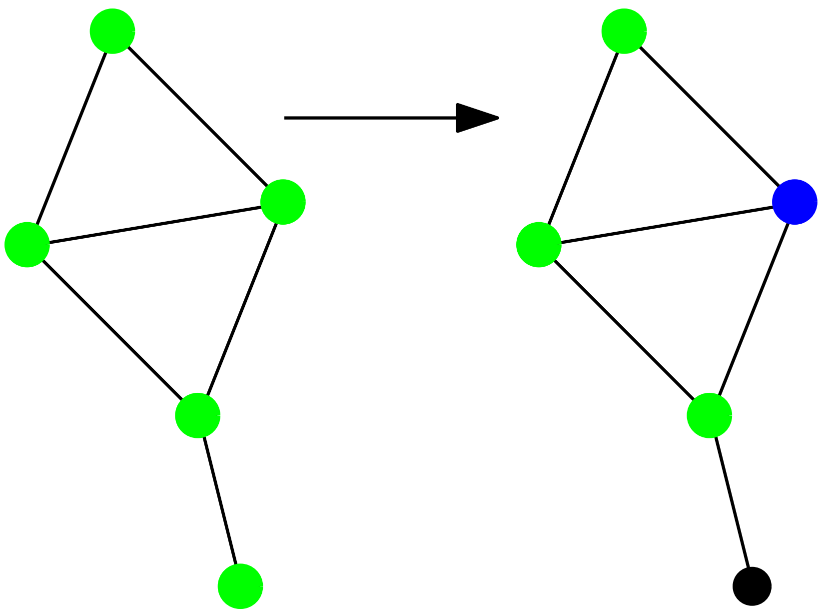
The Bron–Kerbosch Algorithm



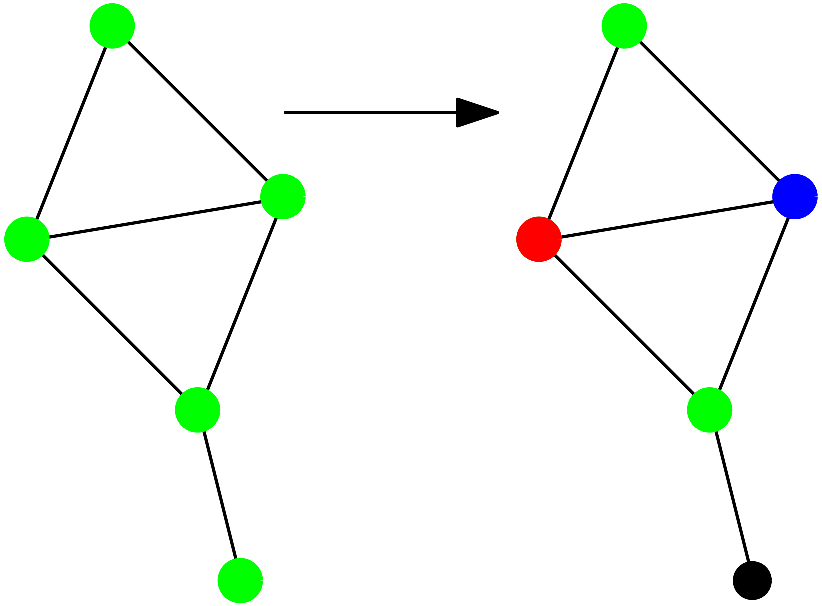
The Bron–Kerbosch Algorithm



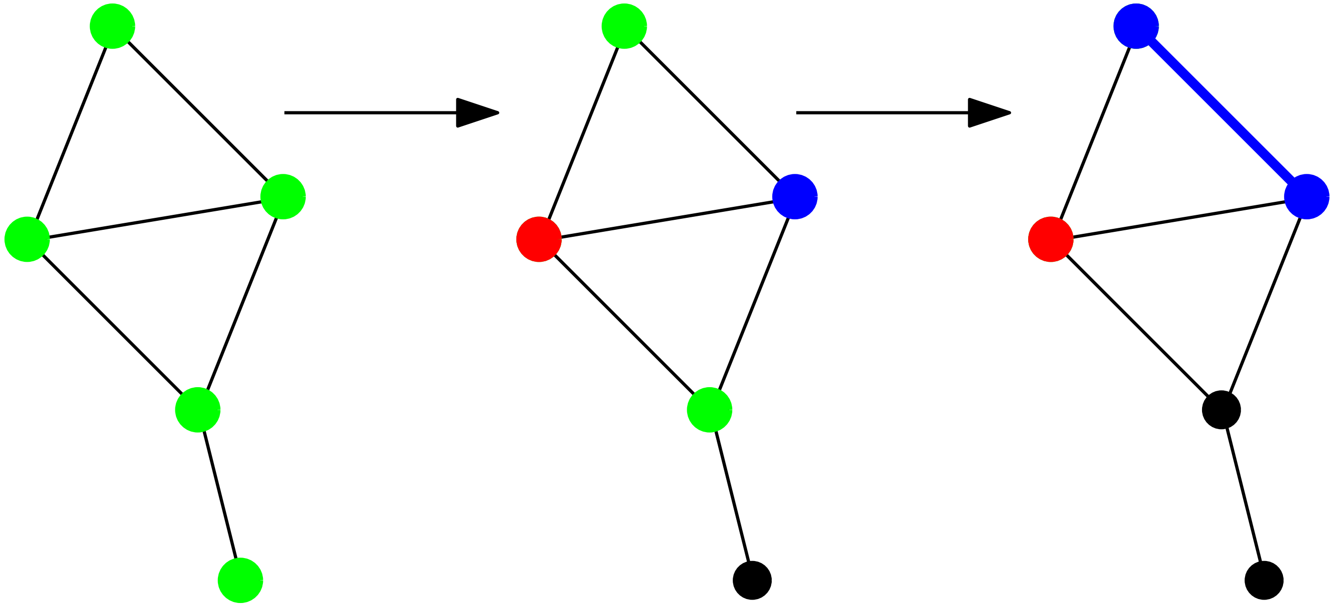
The Bron–Kerbosch Algorithm



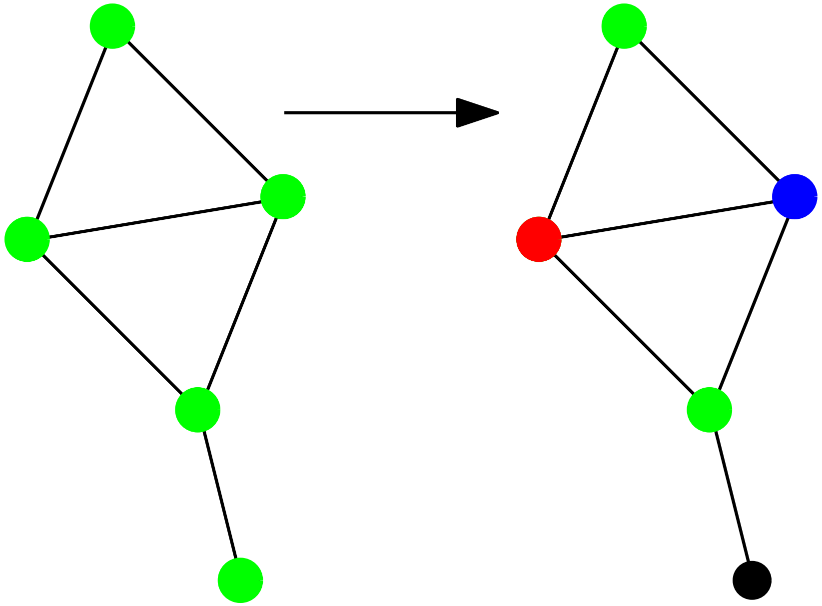
The Bron–Kerbosch Algorithm



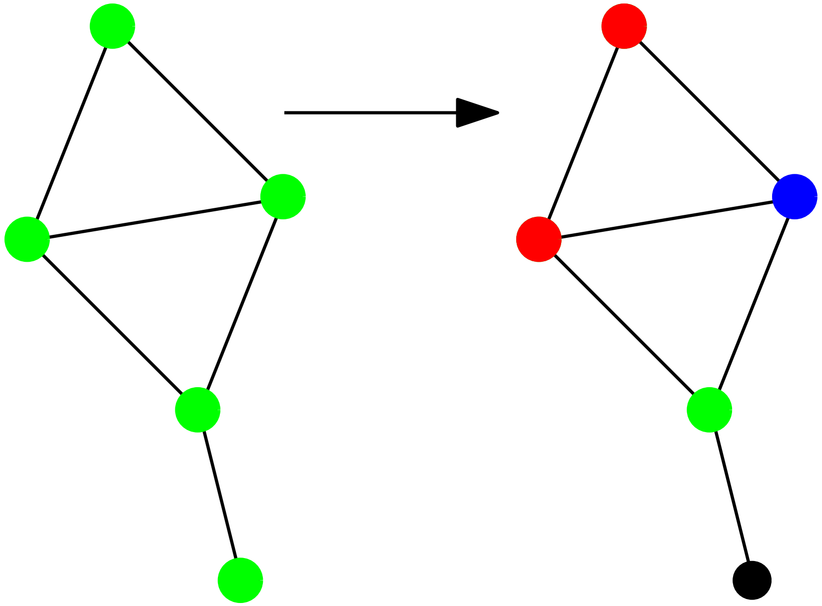
The Bron–Kerbosch Algorithm



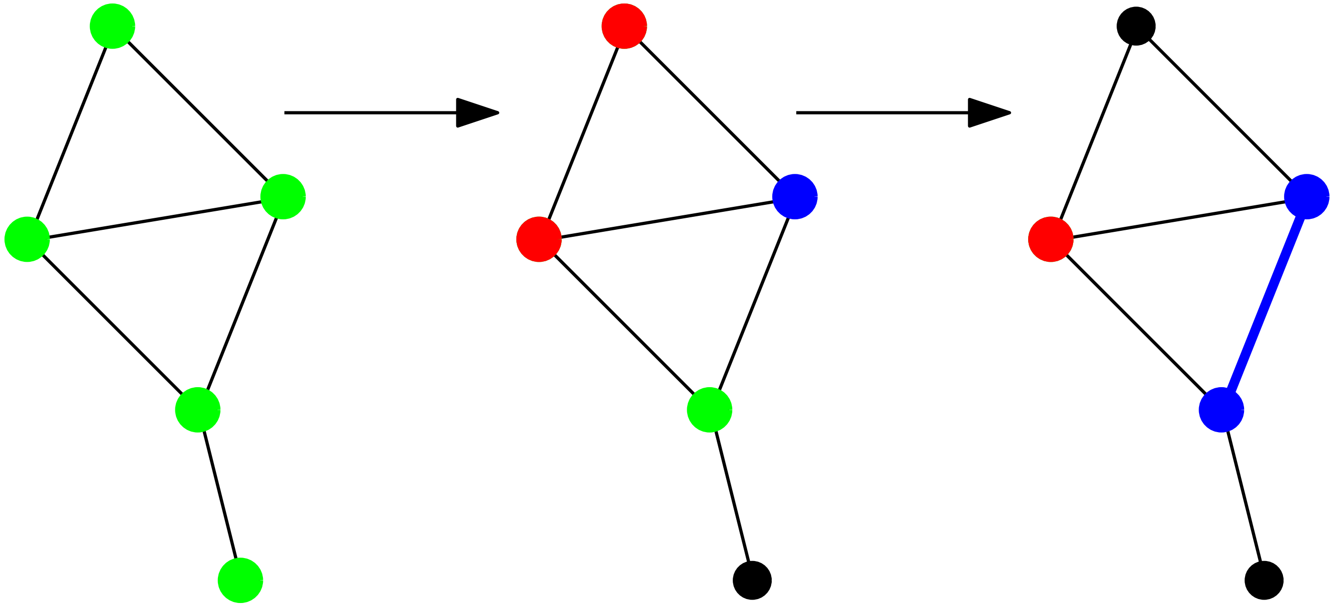
The Bron–Kerbosch Algorithm



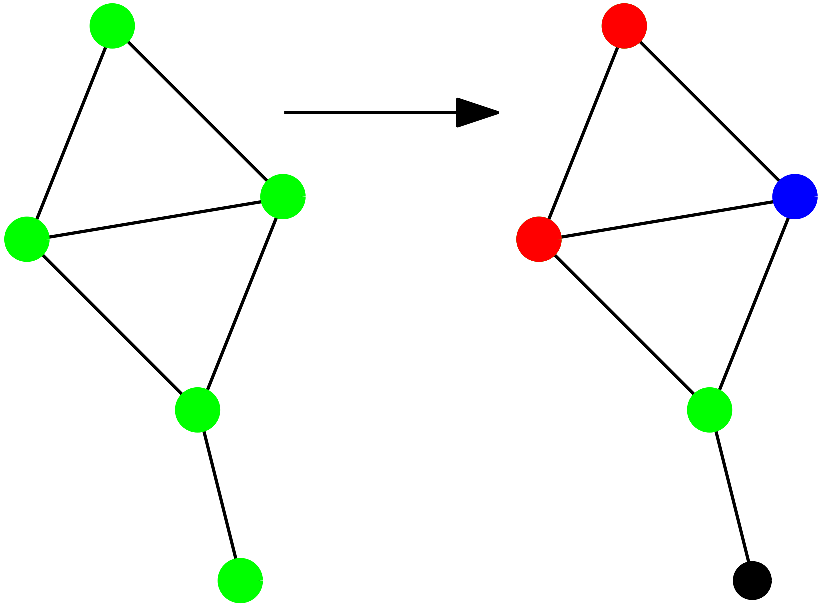
The Bron–Kerbosch Algorithm



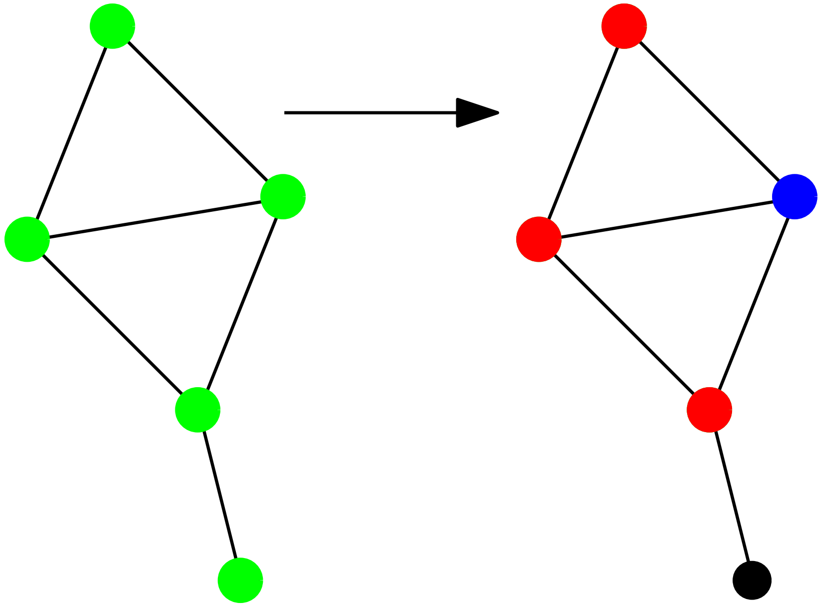
The Bron–Kerbosch Algorithm



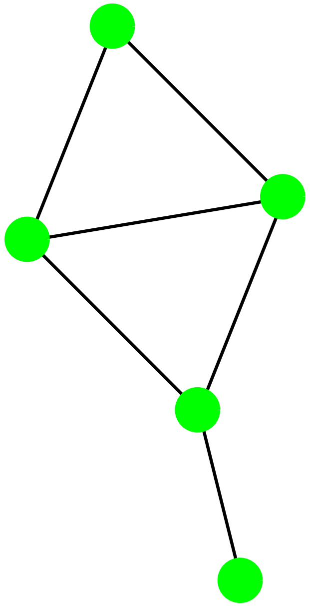
The Bron–Kerbosch Algorithm



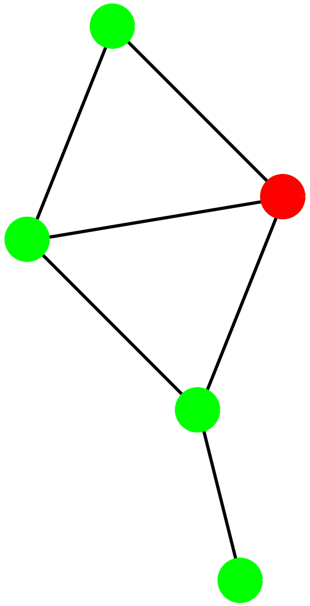
The Bron–Kerbosch Algorithm



The Bron–Kerbosch Algorithm



The Bron–Kerbosch Algorithm

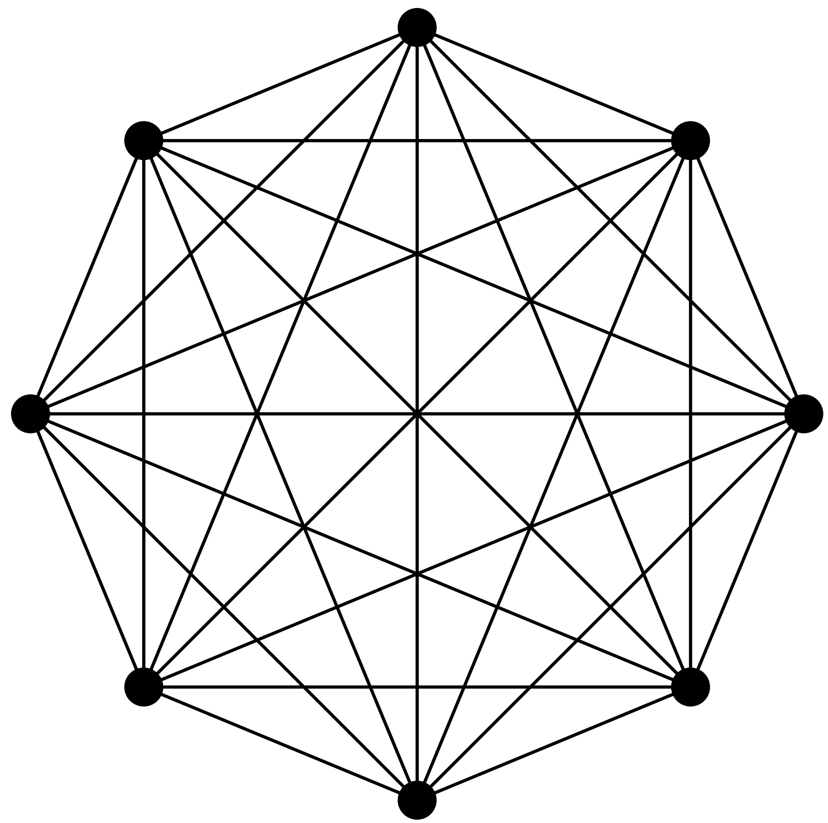


The Bron–Kerbosch Algorithm

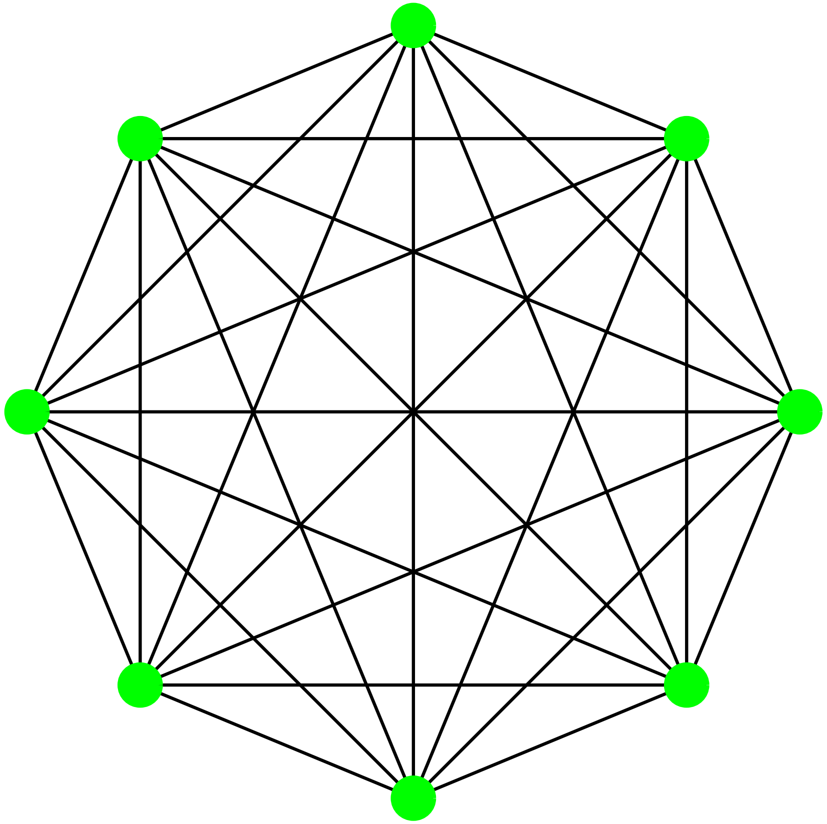
proc BronKerbosch(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
for each vertex $v \in P$ **do**
    BronKerbosch($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

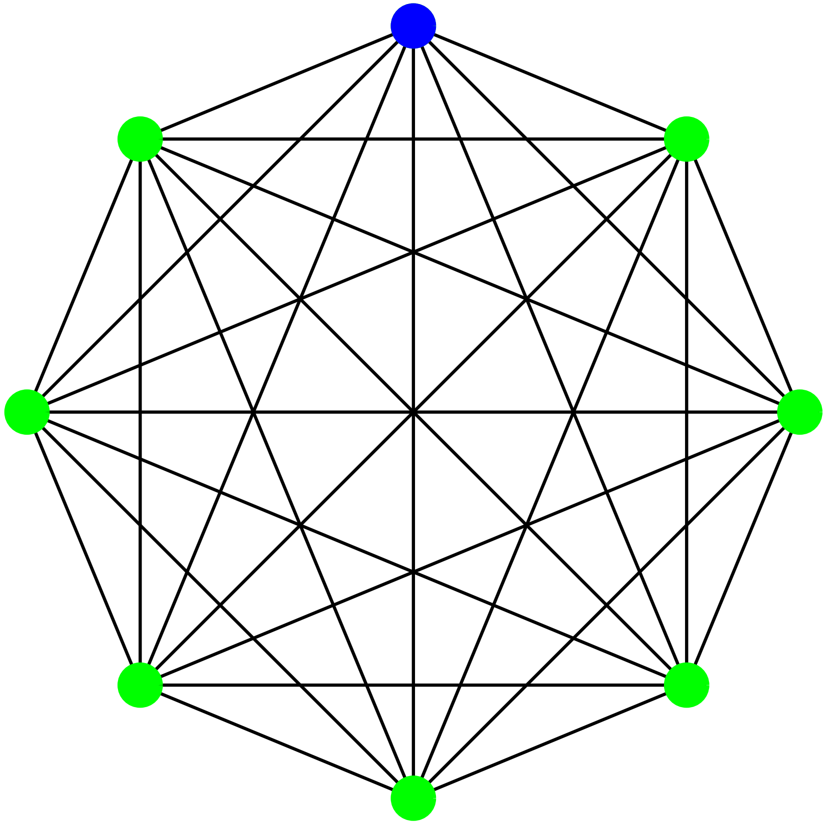
The Bron–Kerbosch Algorithm



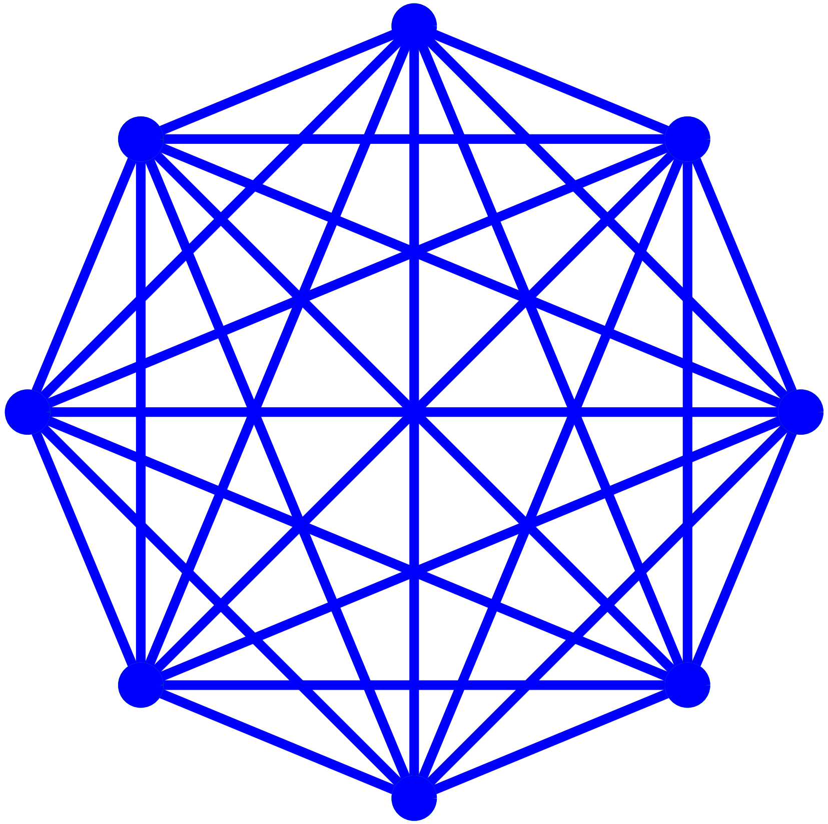
The Bron–Kerbosch Algorithm



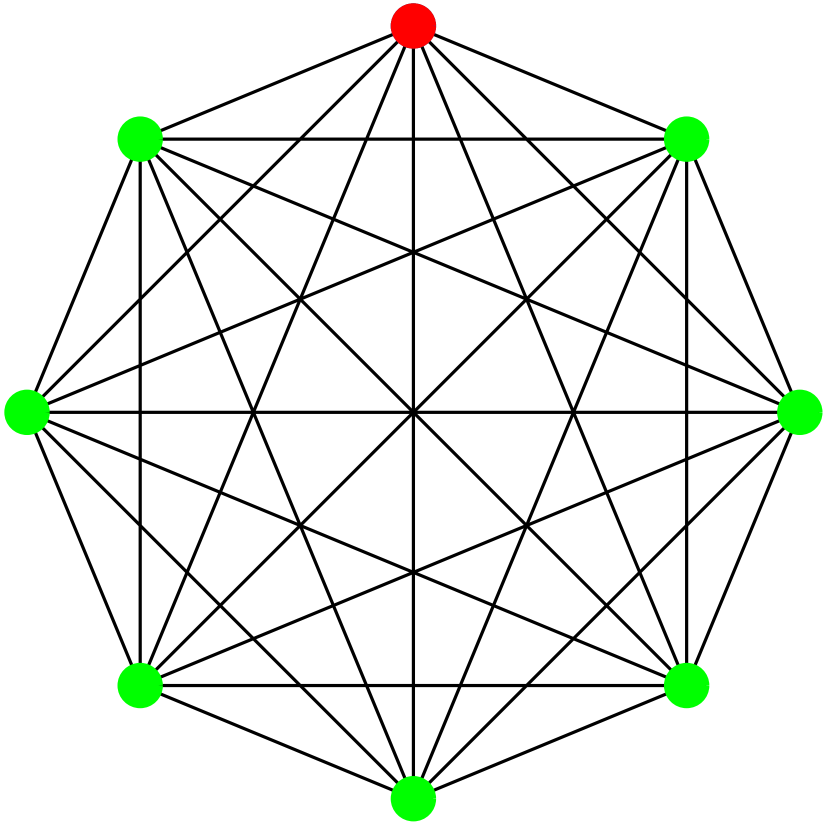
The Bron–Kerbosch Algorithm



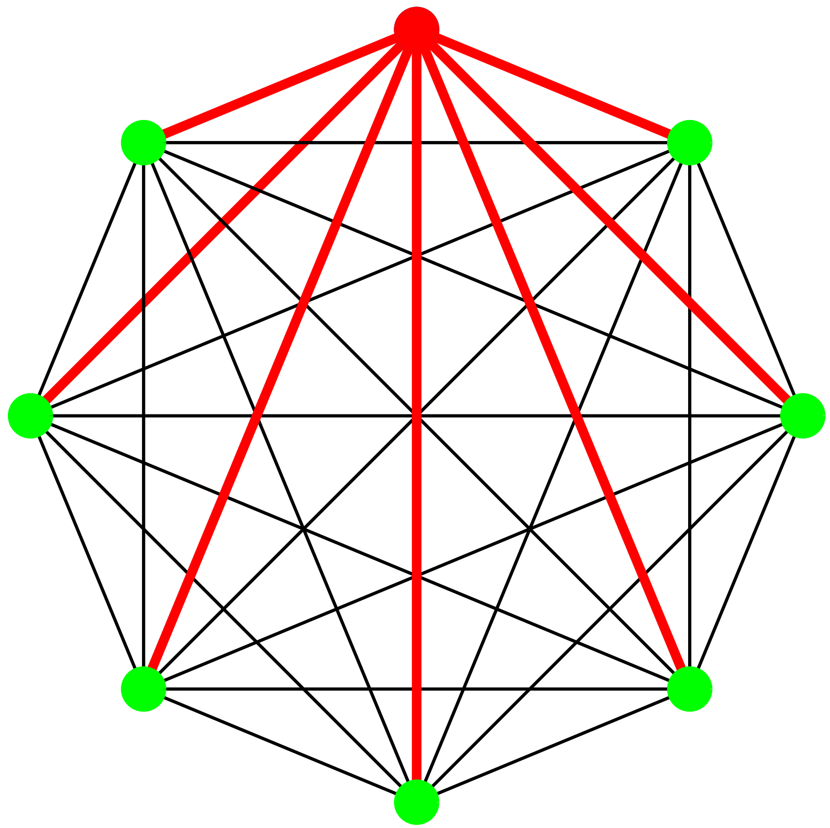
The Bron–Kerbosch Algorithm



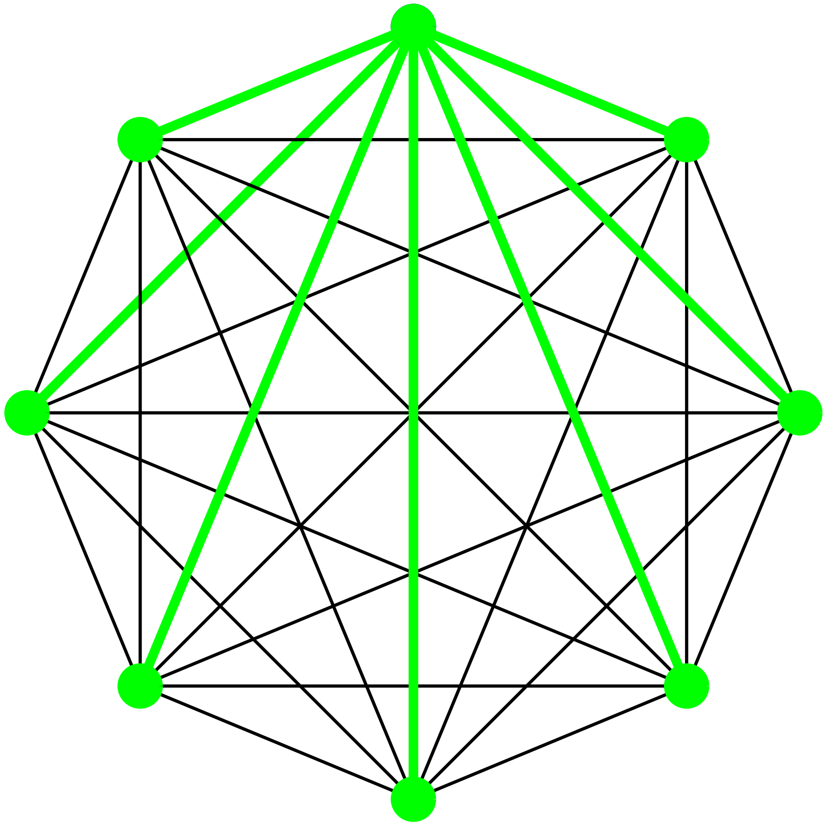
The Bron–Kerbosch Algorithm



The Bron–Kerbosch Algorithm



The Bron–Kerbosch Algorithm



The Bron–Kerbosch Algorithm with Pivoting

proc BronKerboschPivot(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
choose a pivot $u \in P \cup X$
for each vertex $v \in P \setminus \Gamma(u)$ **do**
    BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

The Bron–Kerbosch Algorithm with Pivoting

proc BronKerboschPivot(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
choose a pivot $u \in P \cup X$ to minimize $|P \setminus \Gamma(u)|$
for each vertex $v \in P \setminus \Gamma(u)$ **do**
    BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

The Bron–Kerbosch Algorithm with Pivoting

proc BronKerboschPivot(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
choose a pivot $u \in P \cup X$ to maximize $|P \cap \Gamma(u)|$
for each vertex $v \in P \setminus \Gamma(u)$ **do**
    BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

The Bron–Kerbosch Algorithm with Pivoting

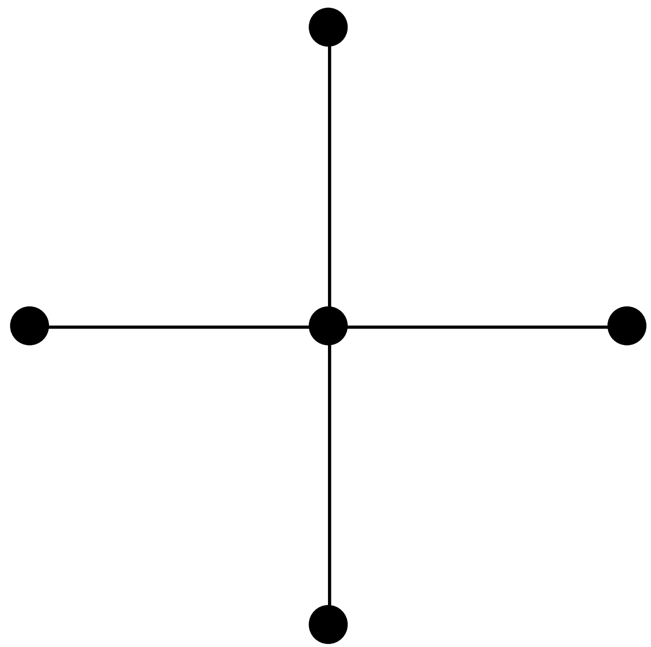
$$T(n) \leq \max_k \{kT(n-k)\} + O(n^2)$$

The Bron–Kerbosch Algorithm with Pivoting

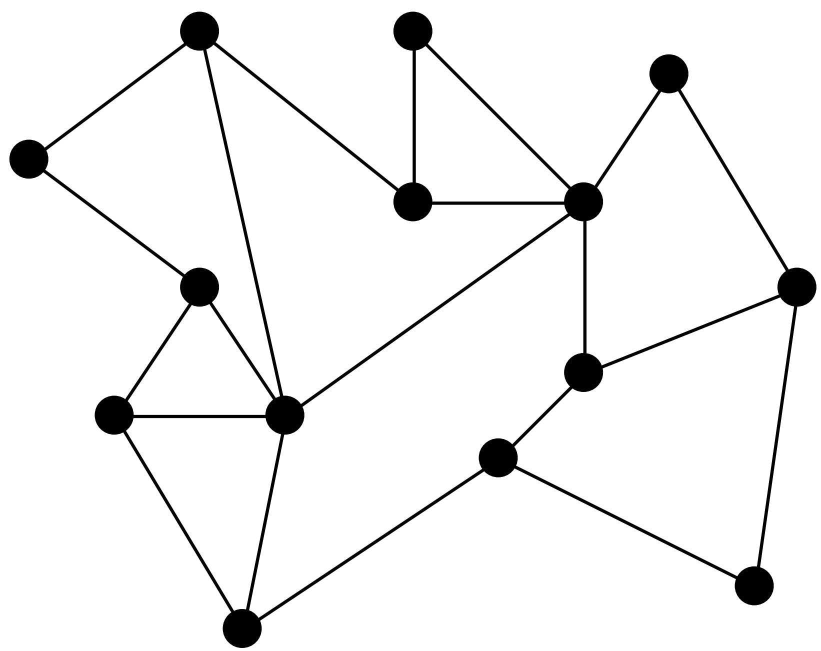
$$T(n) \leq \max_k \{kT(n-k)\} + O(n^2)$$

$$T(n) = O(3^{n/3})$$

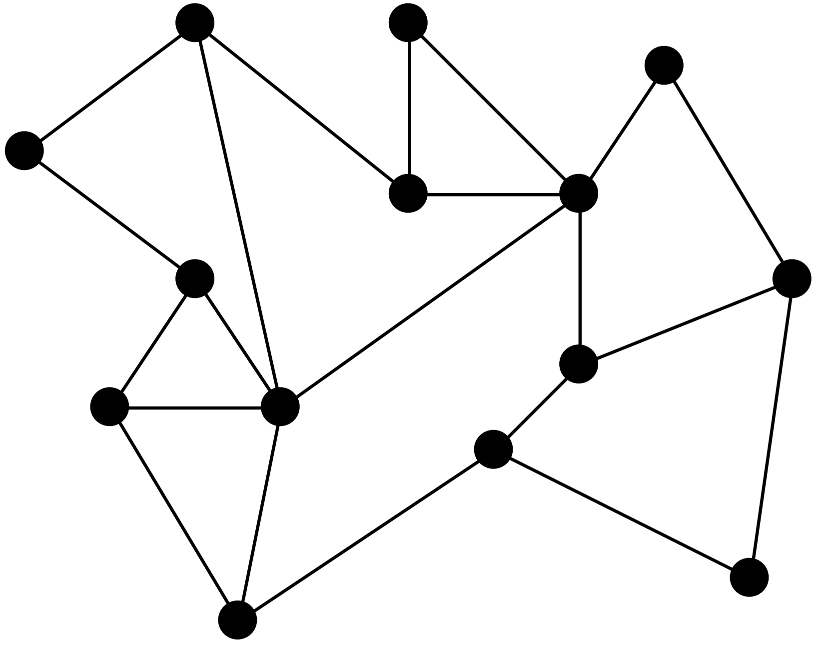
The Bron–Kerbosch Algorithm



The Bron–Kerbosch Algorithm



The Bron–Kerbosch Algorithm



All cliques in planar graphs may be listed in time $O(n)$

Chiba and Nishizeki (1985), Chrobak and Eppstein (1991)

The Bron–Kerbosch Algorithm

Want to characterize the running time with a parameter.

Let p be our parameter of choice.

An algorithm is *fixed-parameter tractable* with parameter p if it has running time

$$f(p)n^{O(1)}$$

The key is to avoid things like n^p.

Parameterize on Sparsity

Parameterize on Sparsity

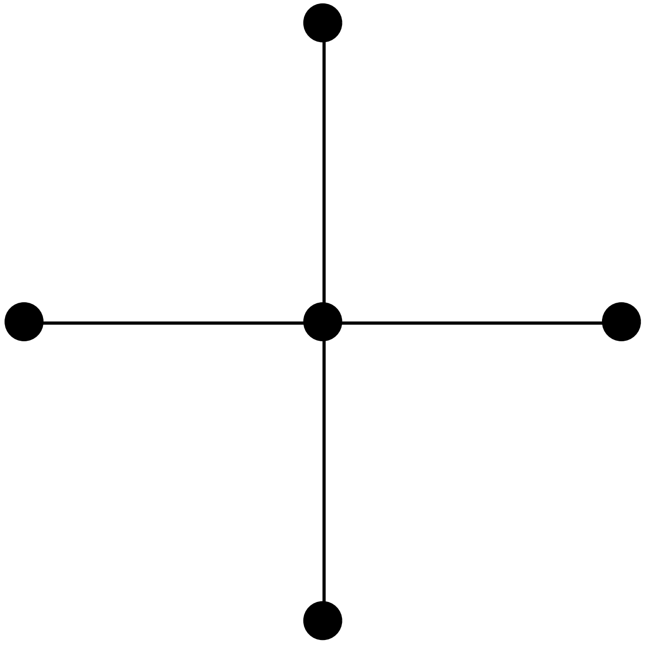
degeneracy:

Parameterize on Sparsity

degeneracy:

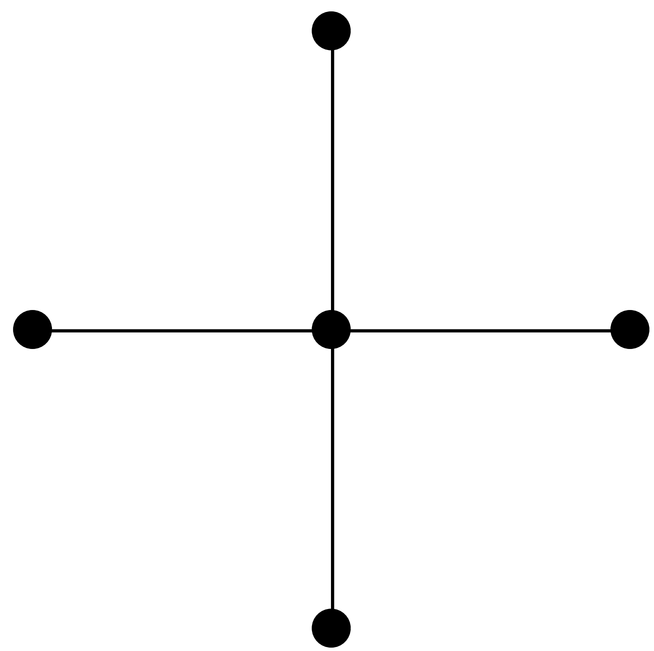
The minimum integer d such that every subgraph of G has a vertex of degree d or less.

Degeneracy



Degeneracy

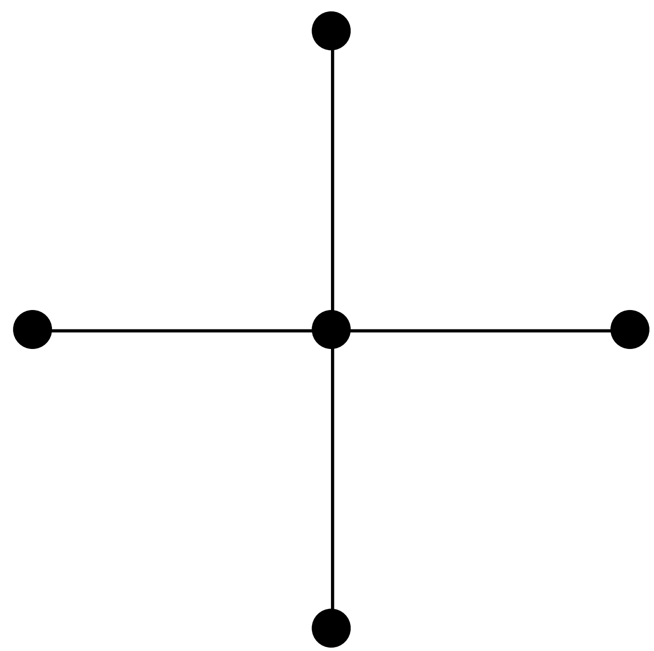
$d = 1$

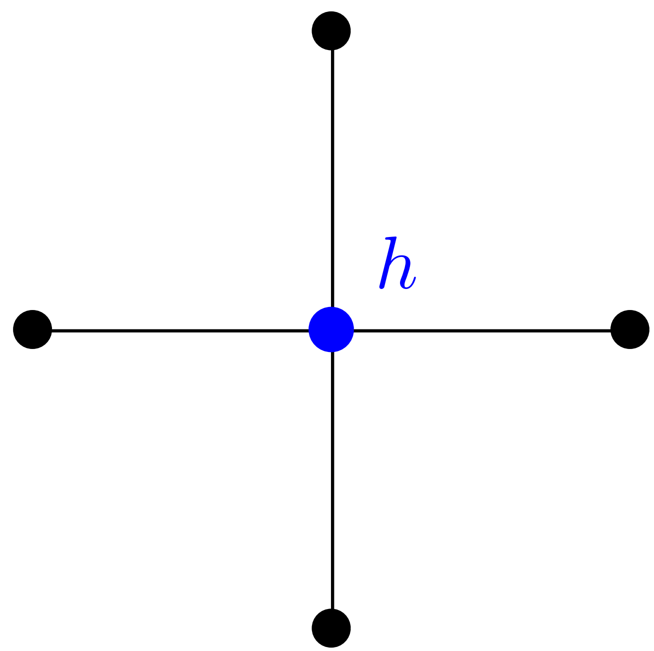


Degeneracy

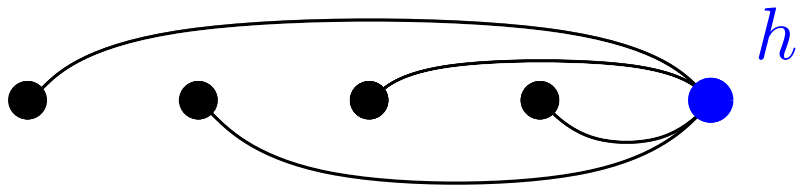
degeneracy:

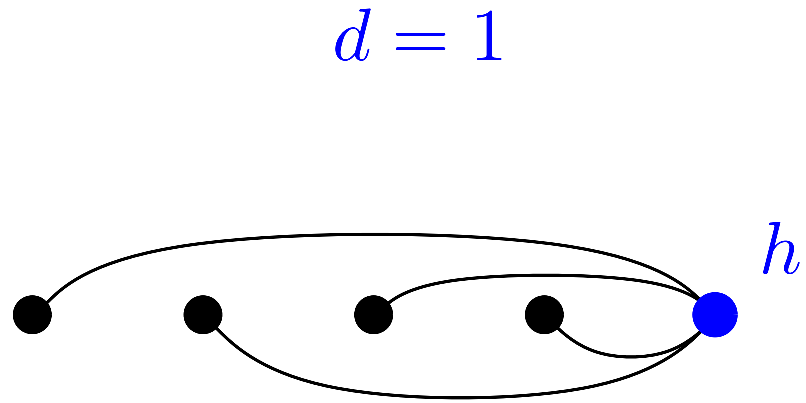
The minimum integer d such that there is an ordering of the vertices where each vertex has at most d neighbors later in the ordering.

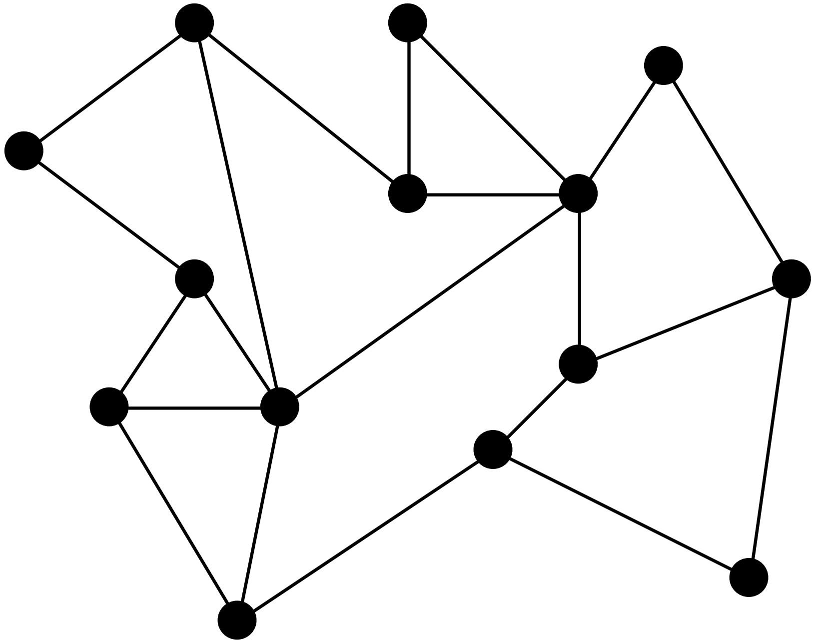




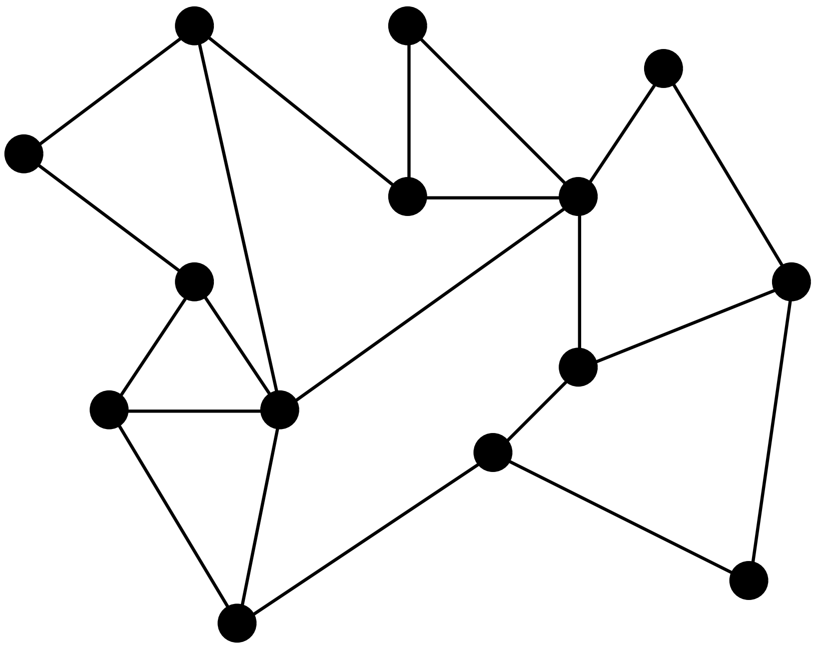
h


h

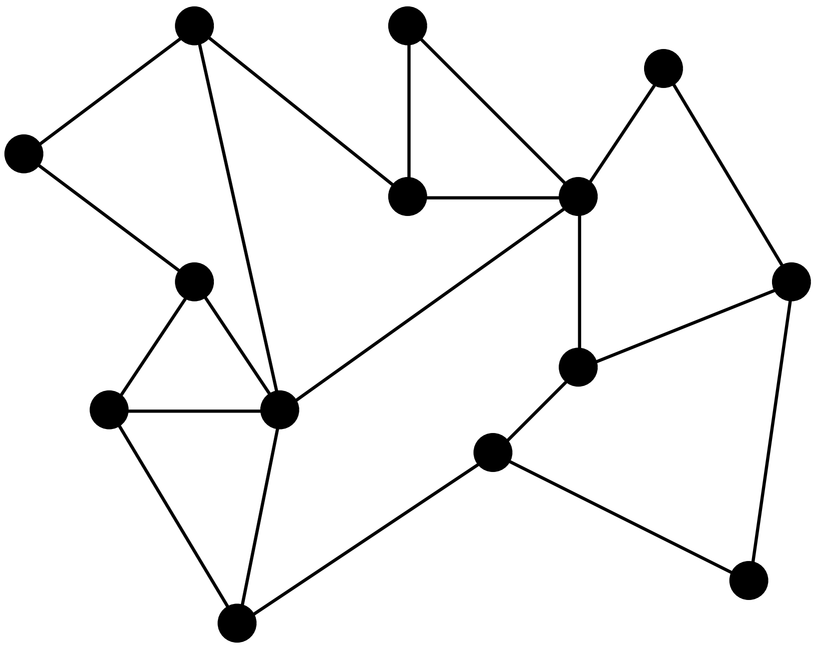

$d = 1$
h

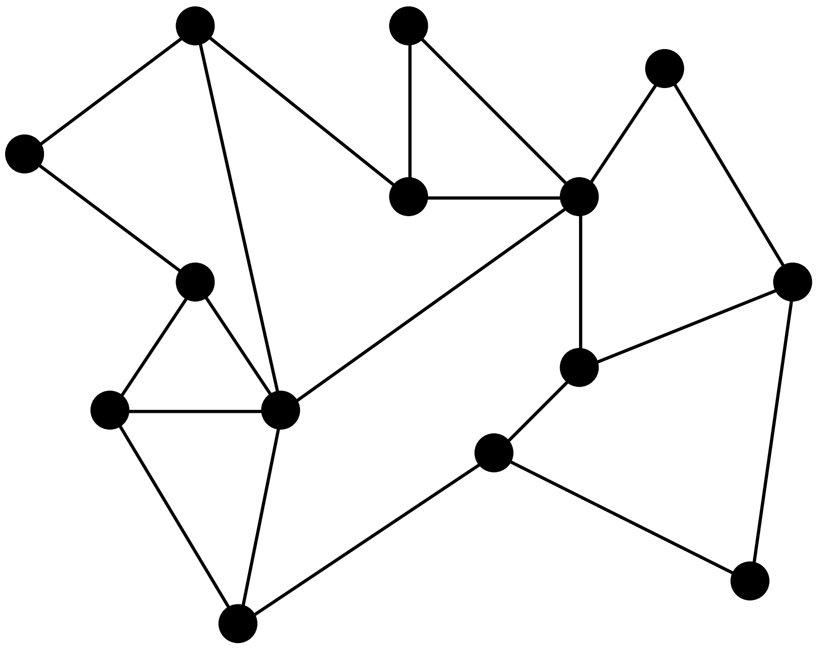


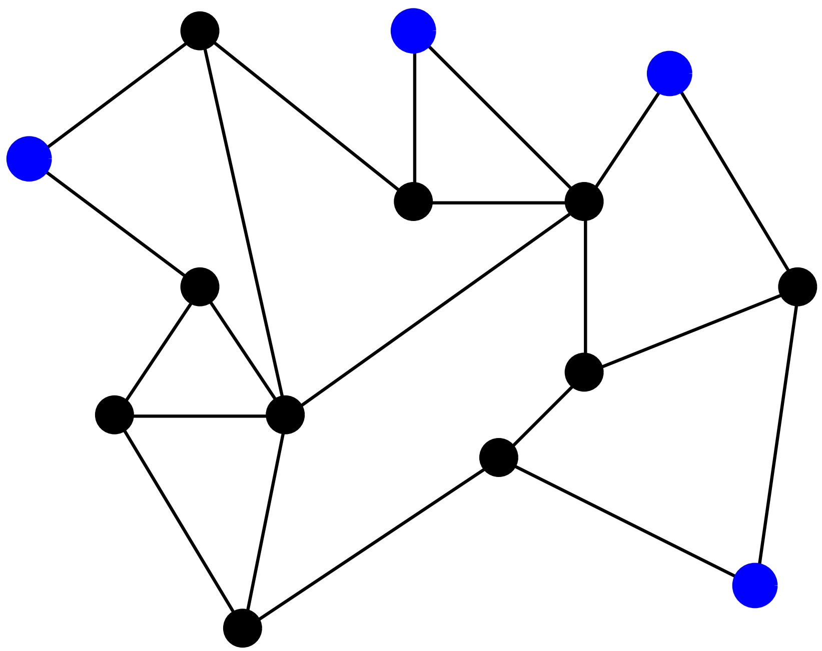
Planar graphs have degeneracy at most 5

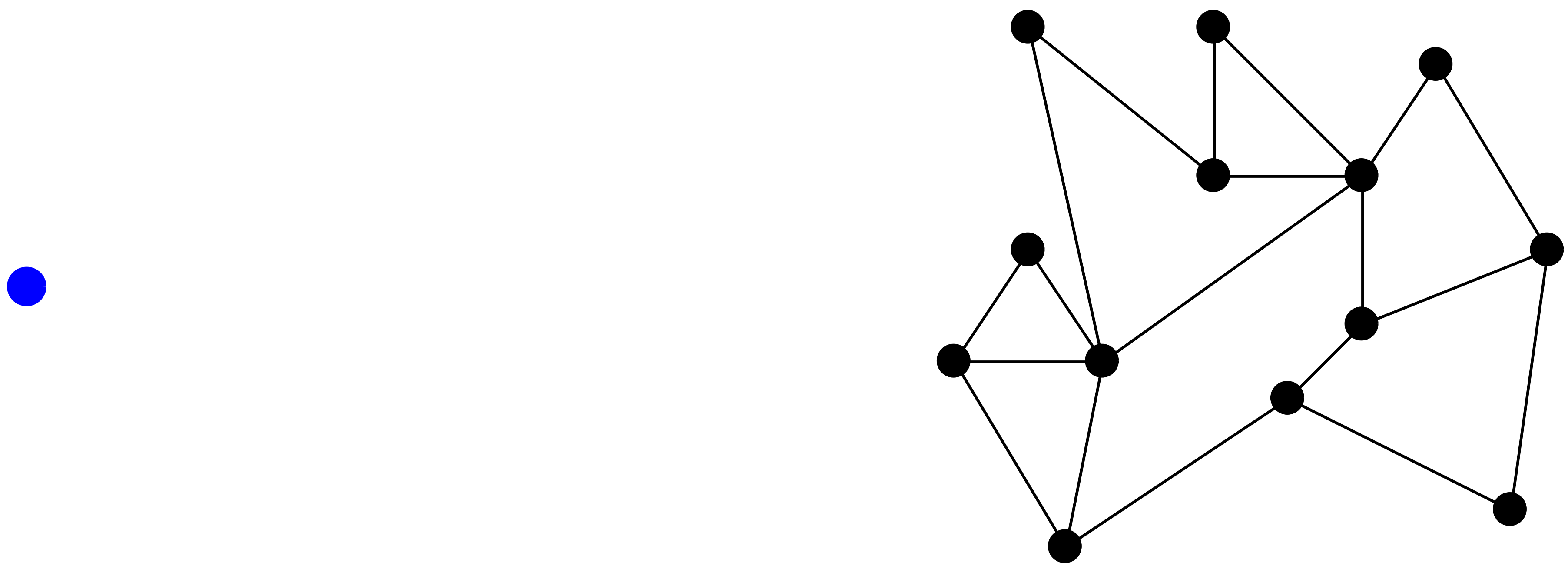


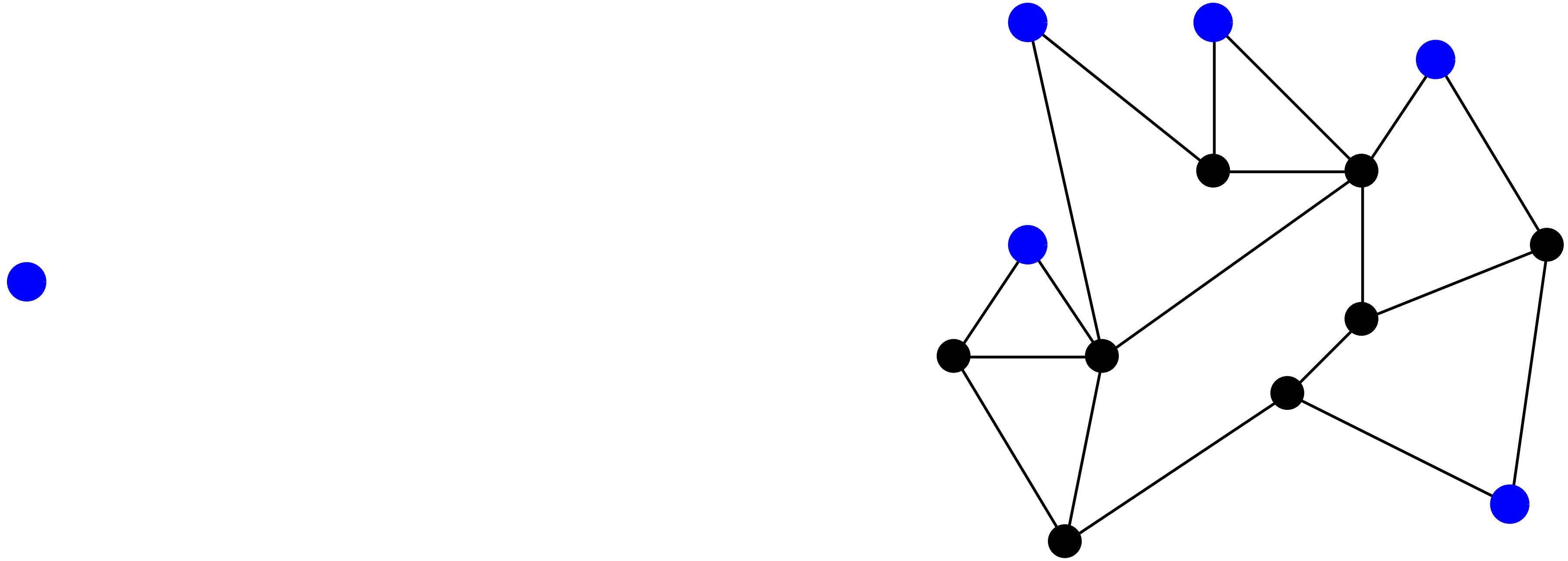
Degeneracy is easy to compute

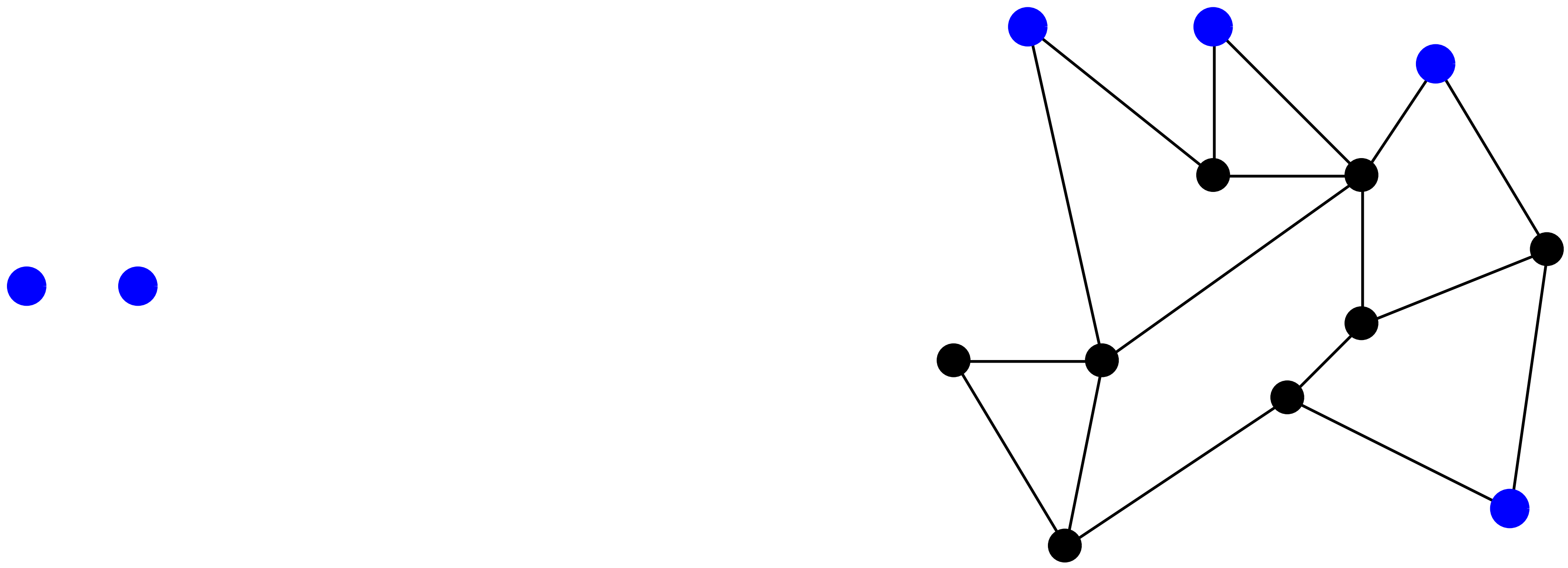


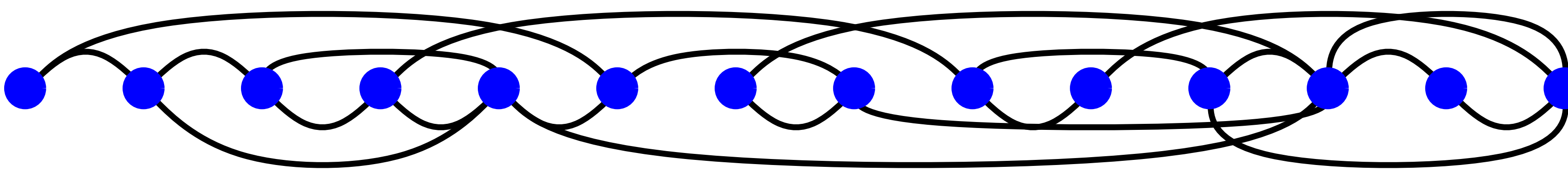












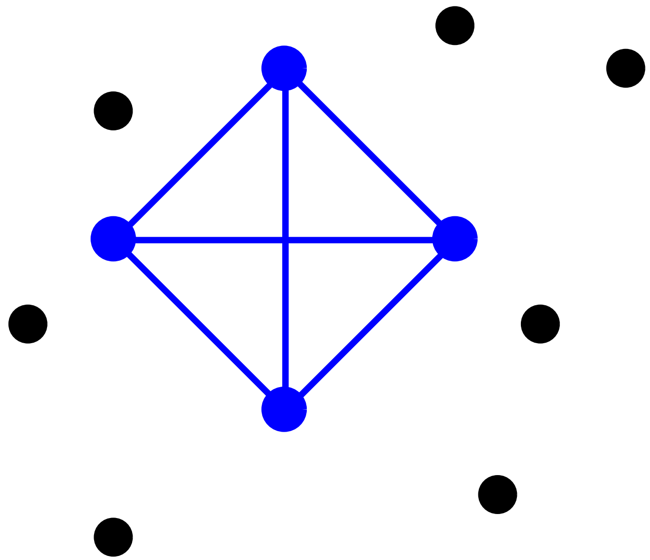
d-degenerate graphs...

d-degenerate graphs...

cannot contain cliques with more than $d + 1$ vertices

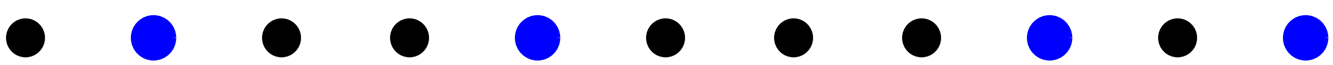
d-degenerate graphs...

cannot contain cliques with more than $d + 1$ vertices



d-degenerate graphs...

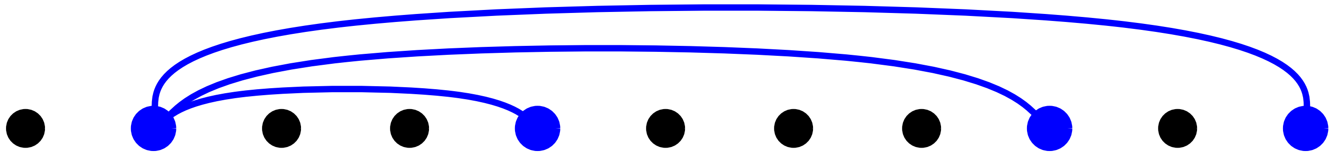
cannot contain cliques with more than $d + 1$ vertices



d-degenerate graphs...

cannot contain cliques with more than $d + 1$ vertices

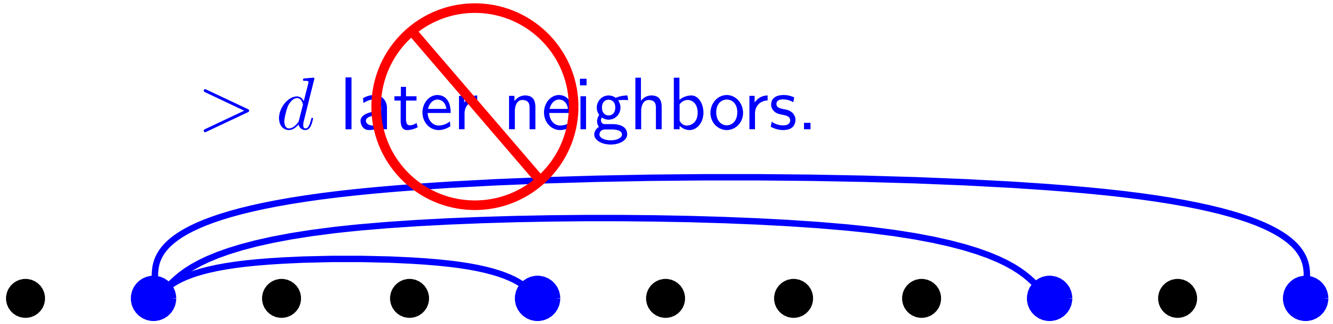
$> d$ later neighbors.



d-degenerate graphs...

cannot contain cliques with more than $d + 1$ vertices

$> d$ later neighbors.



d-degenerate graphs...

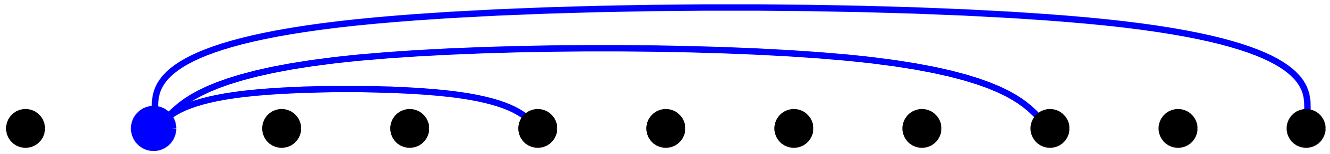
d-degenerate graphs...

have fewer than dn edges.

d-degenerate graphs...

have fewer than dn edges.

$\leq d$ later neighbors.



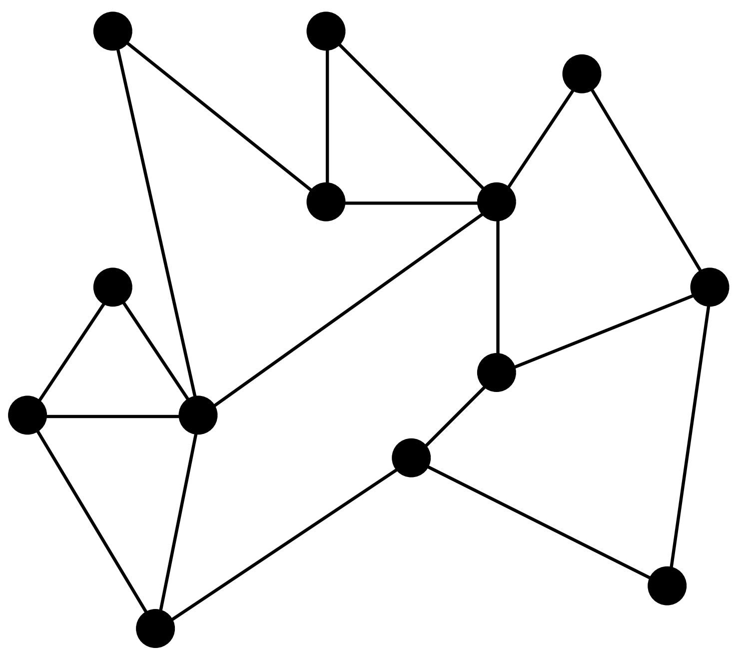
A few more facts about degeneracy...

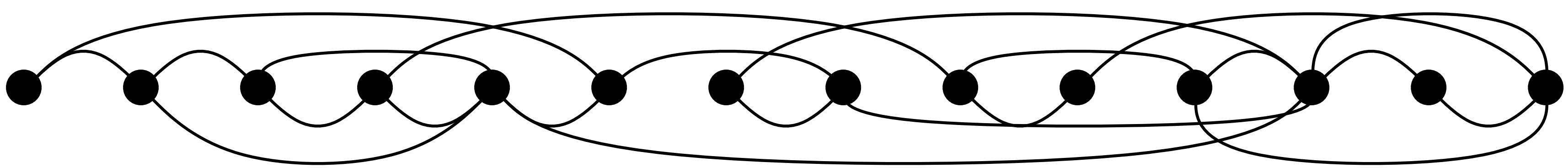
Degeneracy is within a constant factor of other popular sparsity measures.

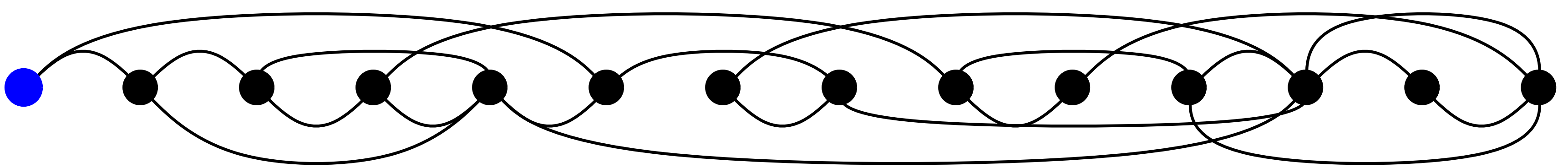
A few more facts about degeneracy...

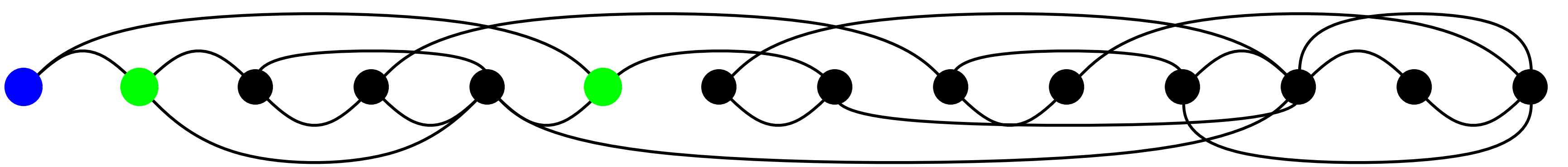
Degeneracy is within a constant factor of other popular sparsity measures.

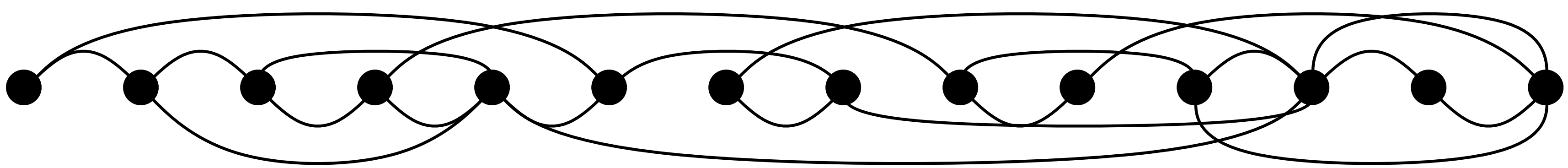
Graphs generated by the preferential attachment mechanism of Barabási and Albert have low degeneracy.

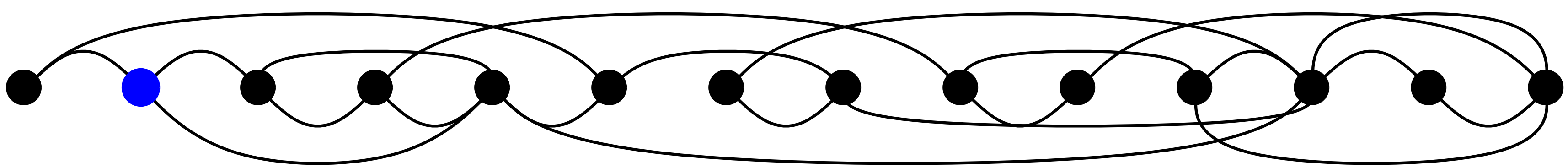


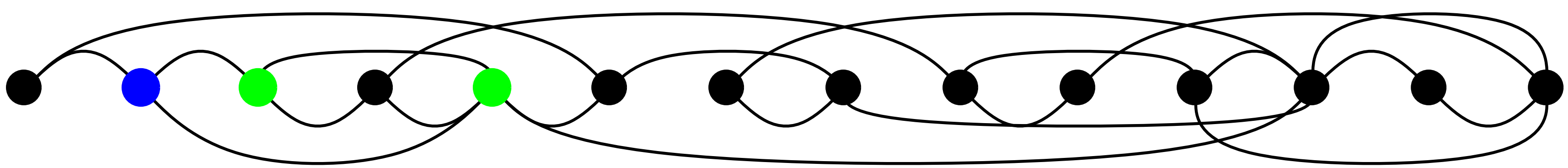


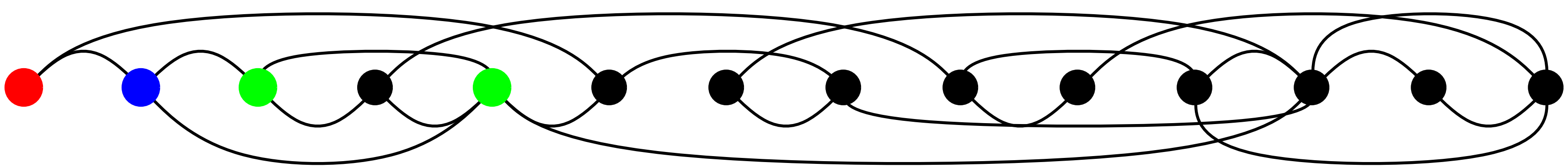












proc BronKerboschDegeneracy(V, E)

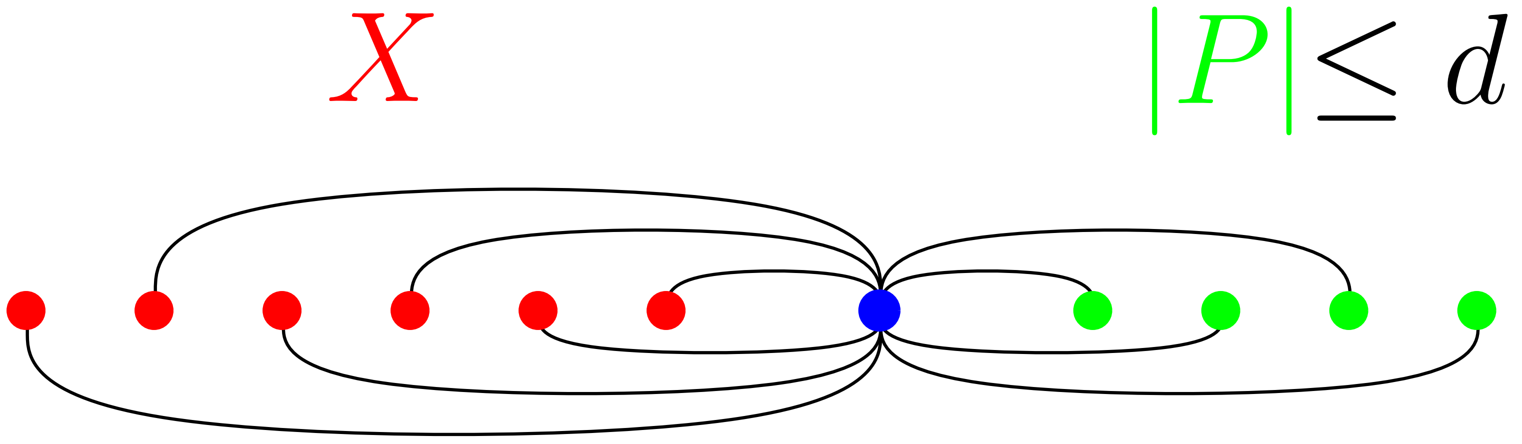
for each vertex v_i in a degeneracy ordering v_0, v_1, v_2, . . . of (V, E) **do**

    $P \leftarrow v_i$'s later neighbors

    $X \leftarrow v_i$'s earlier neighbors

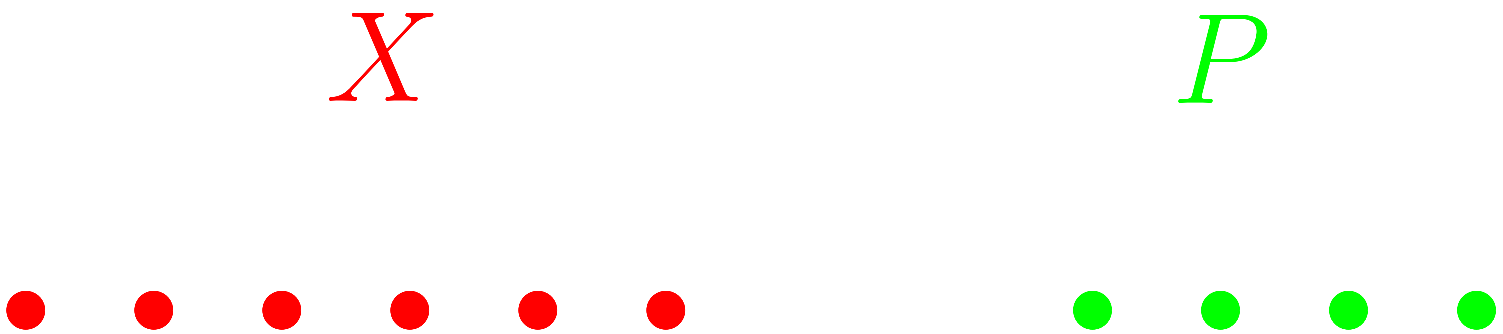
    BronKerboschPivot(P, $\{v_i\}$, X)

end for


X
$|P| \leq d$

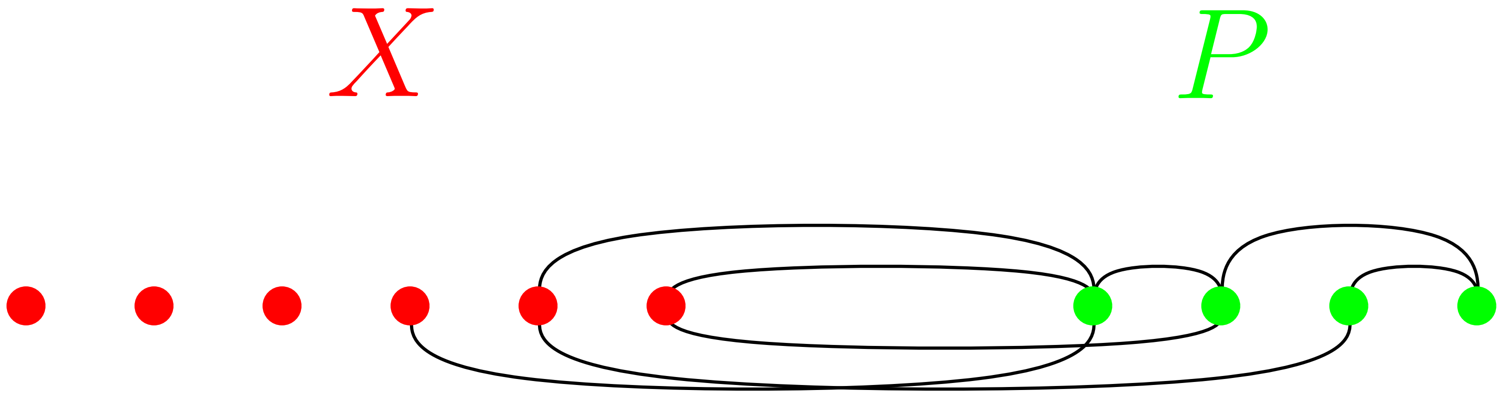
Computing the pivot

Pick $u \in X \cup P$ that maximizes $|P \cap \Gamma(u)|$.



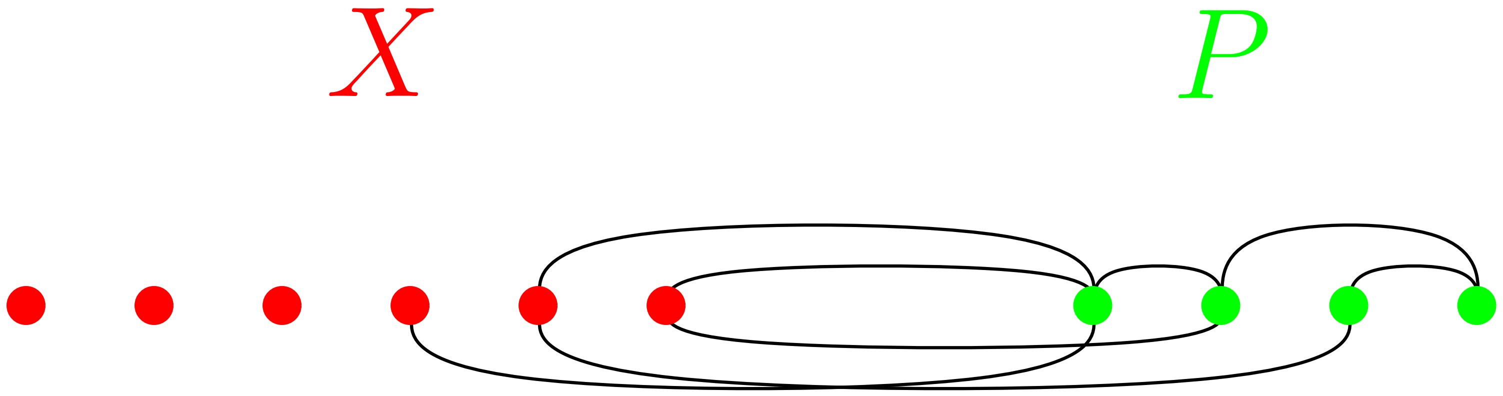
Computing the pivot

Pick $u \in X \cup P$ that maximizes $|P \cap \Gamma(u)|$.



Computing the pivot

Pick $u \in X \cup P$ that maximizes $|P \cap \Gamma(u)|$.



$$O(|P|(|X| + |P|))$$

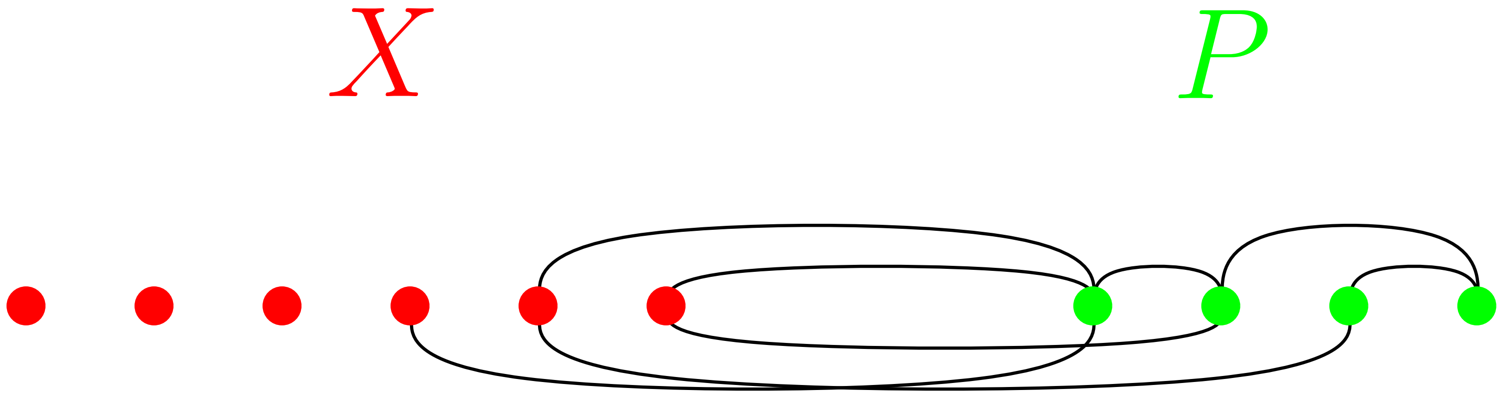
proc BronKerboschPivot(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
choose a pivot $u \in P \cup X$ to maximize $|P \cap \Gamma(u)|$
for each vertex $v \in P \setminus \Gamma(u)$ **do**
    BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

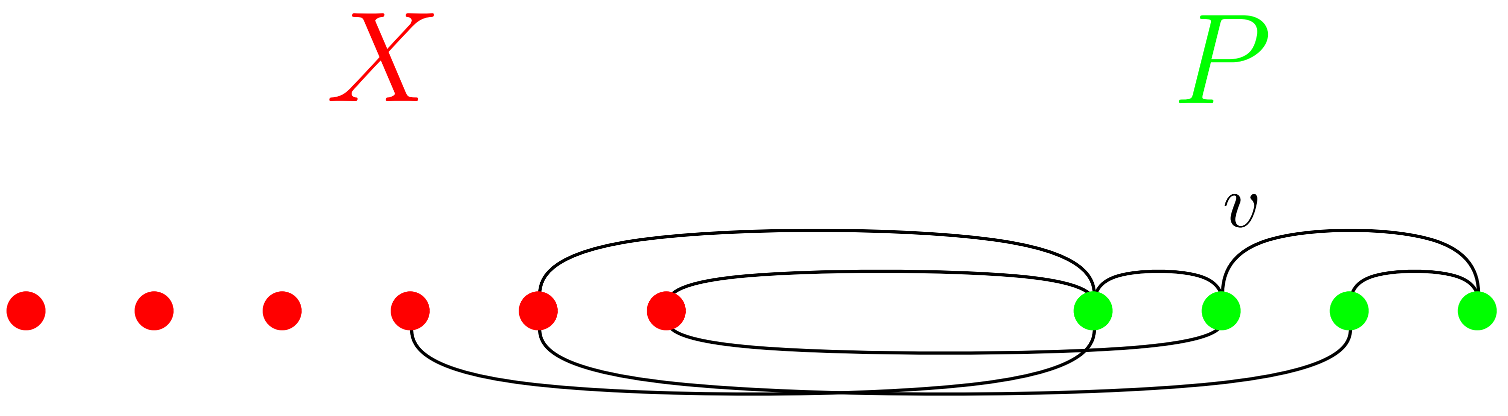
proc BronKerboschPivot(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
choose a pivot $u \in P \cup X$ to maximize $|P \cap \Gamma(u)|$
for each vertex $v \in P \setminus \Gamma(u)$ **do**
    BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

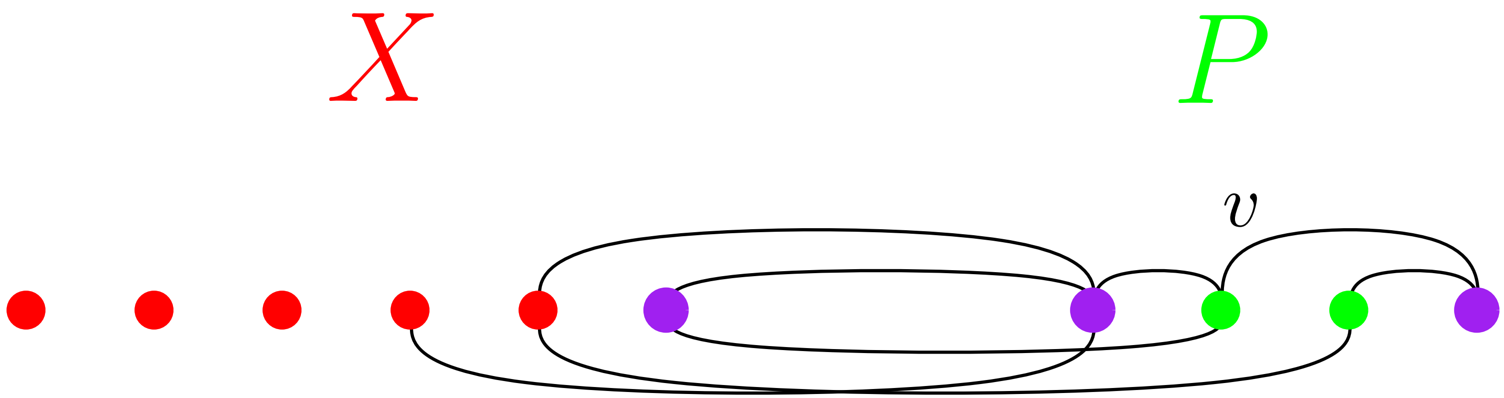
Find subgraph induced by v's neighbors.



Find subgraph induced by v's neighbors.

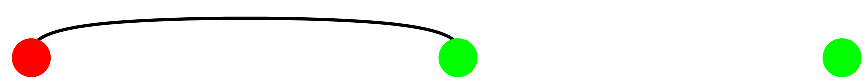


Find subgraph induced by v's neighbors.



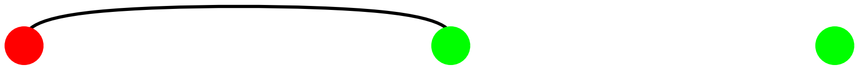
Find subgraph induced by v's neighbors.

$X \cap \Gamma(v)$ $P \cap \Gamma(v)$



Find subgraph induced by v's neighbors.

$X \cap \Gamma(v)$ $P \cap \Gamma(v)$



$O(|P|(|X| + |P|))$

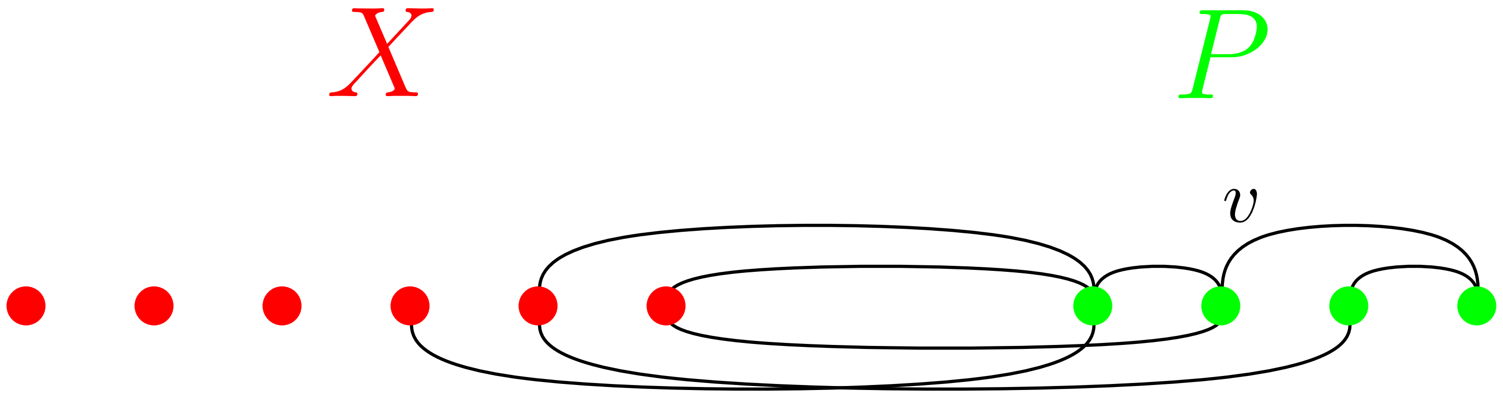
proc BronKerboschPivot(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
choose a pivot $u \in P \cup X$ to maximize $|P \cap \Gamma(u)|$
for each vertex $v \in P \setminus \Gamma(u)$ **do**
    BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

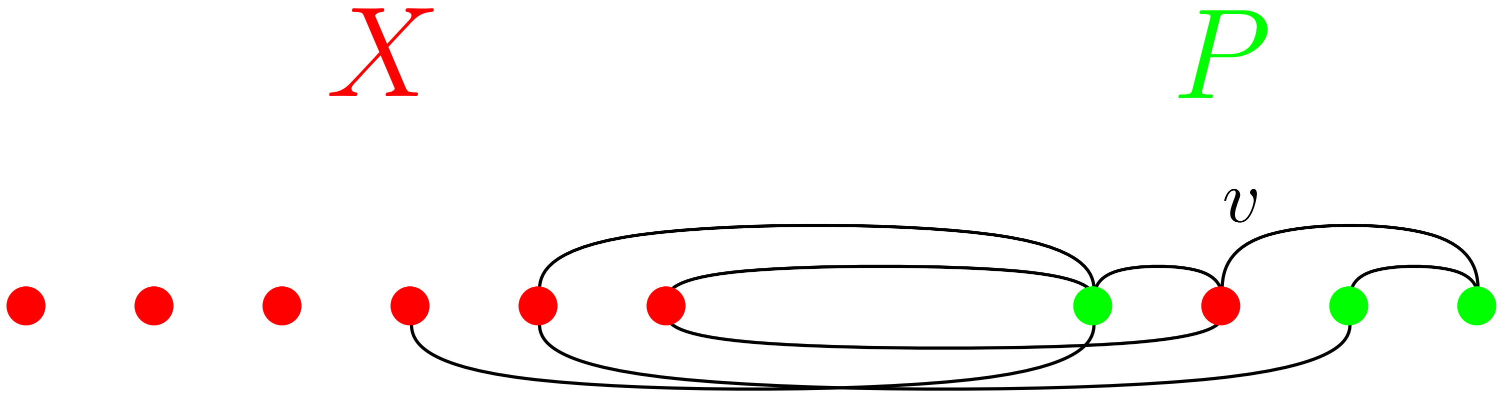
proc BronKerboschPivot(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
choose a pivot $u \in P \cup X$ to maximize $|P \cap \Gamma(u)|$
for each vertex $v \in P \setminus \Gamma(u)$ **do**
    BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

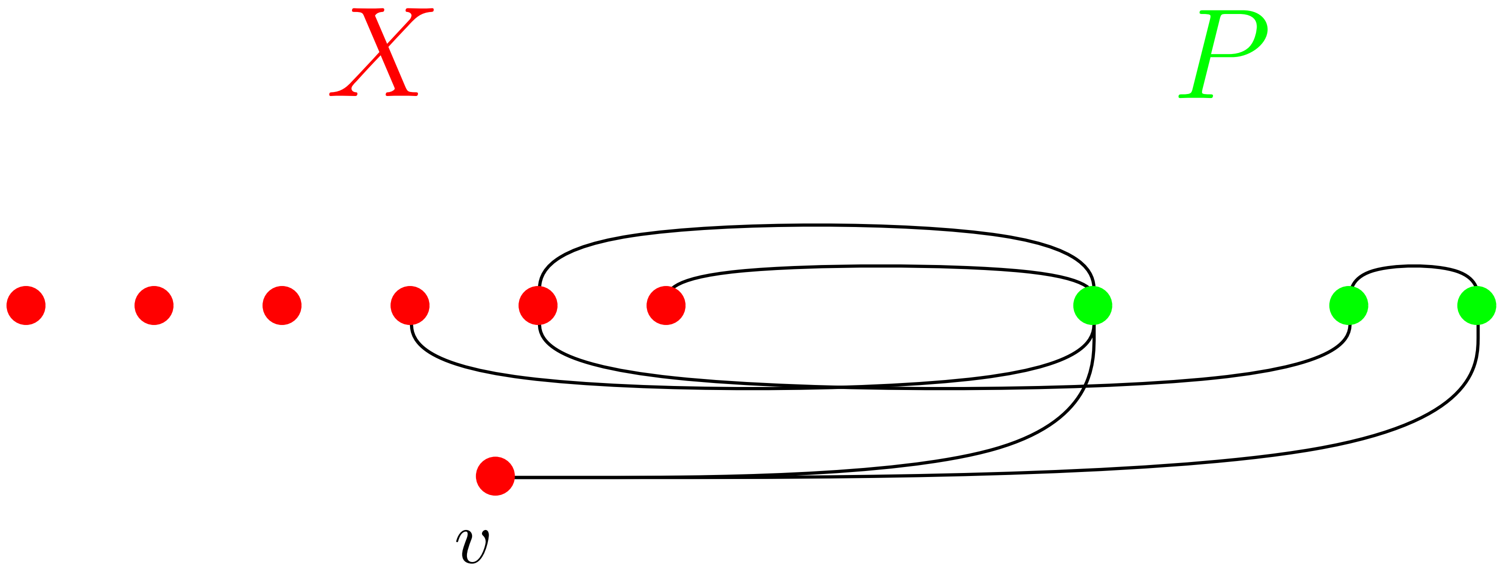
Remove v from P and add it to X.



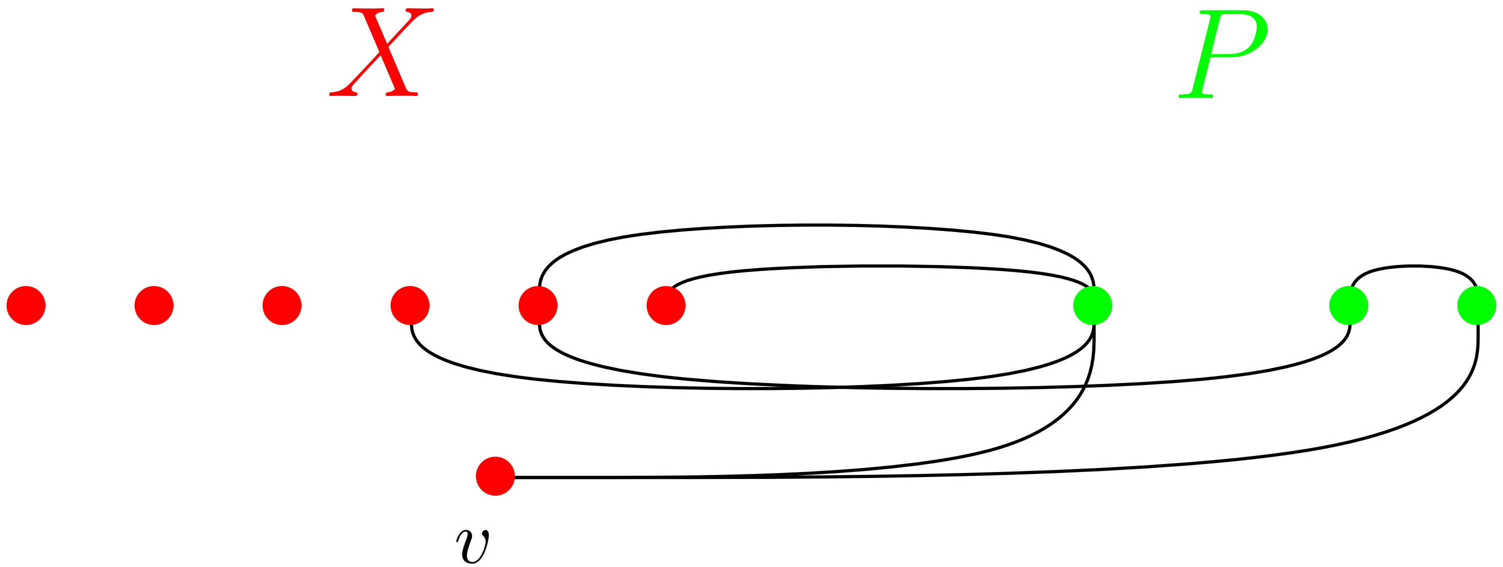
Remove v from P and add it to X.



Remove v from P and add it to X.



Remove v from P and add it to X.



$$O(|P|(|X| + |P|))$$

proc BronKerboschPivot(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
choose a pivot $u \in P \cup X$ to maximize $|P \cap \Gamma(u)|$
for each vertex $v \in P \setminus \Gamma(u)$ **do**
    BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

proc BronKerboschPivot(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
choose a pivot $u \in P \cup X$ to maximize $|P \cap \Gamma(u)|$
for each vertex $v \in P \setminus \Gamma(u)$ **do**
    BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

$$O(|P|^2(|X| + |P|))$$

$$
\begin{aligned}
T(n) &\leq \max_k \{kT(n-k)\} + O(n^2) \\
&= O(3^{n/3})
\end{aligned}
$$

$$
\begin{aligned}
T(n) &\leq \max_k \{kT(n-k)\} + O(n^2) \\
&= O(3^{n/3})
\end{aligned}
$$

$$
D(p, x) \leq \max_k \{kD(p-k, x)\} + O(p^2(p+x))
$$

$$
\begin{aligned}
T(n) &\leq \max_k \{kT(n-k)\} + O(n^2) \\
&= O(3^{n/3})
\end{aligned}
$$

$$
\begin{aligned}
D(p,x) &\leq \max_k \{kD(p-k,x)\} + O(p^2(p+x)) \\
&\leq (d+x)\left[\max_k \left\{\tfrac{kD(p-k,x)}{d+x}\right\} + O(p^2)\right]
\end{aligned}
$$

$$
\begin{aligned}
T(n) &\le \max_k \{kT(n-k)\} + O(n^2) \\
&= O(3^{n/3})
\end{aligned}
$$

$$
\begin{aligned}
D(p,x) &\le \max_k \{kD(p-k,x)\} + O(p^2(p+x)) \\
&\le (d+x)\left[\max_k \left\{\tfrac{kD(p-k,x)}{d+x}\right\} + O(p^2)\right] \\
&\le (d+x)\left[\max_k \{kT(p-k)\} + O(p^2)\right]
\end{aligned}
$$

$$
\begin{aligned}
T(n) &\leq \max_k\{kT(n-k)\} + O(n^2) \\
&= O(3^{n/3})
\end{aligned}
$$

$$
\begin{aligned}
D(p,x) &\leq \max_k\{kD(p-k,x)\} + O(p^2(p+x)) \\
&\leq (d+x)\left[\max_k \left\{\tfrac{kD(p-k,x)}{d+x}\right\} + O(p^2)\right] \\
&\leq (d+x)\left[\max_k\{kT(p-k)\} + O(p^2)\right] \\
&= O((d+x)3^{p/3})
\end{aligned}
$$

$$
\begin{aligned}
T(n) &\leq \max_k \{kT(n-k)\} + O(n^2) \\
&= O(3^{n/3})
\end{aligned}
$$

$$
\begin{aligned}
D(p,x) &\leq \max_k \{kD(p-k,x)\} + O(p^2(p+x)) \\
&\leq (d+x)\left[\max_k \left\{\tfrac{kD(p-k,x)}{d+x}\right\} + O(p^2)\right] \\
&\leq (d+x)\left[\max_k \{kT(p-k)\} + O(p^2)\right] \\
&= O((d+x)3^{p/3}) \\
&= O((d+x)3^{d/3})
\end{aligned}
$$

$$\sum_{v \in V} O(d + |X_v|)3^{d/3})$$

$$\sum_{v \in V} O(d + |X_v|)3^{d/3})$$

$$= O((dn + m)3^{d/3})$$

$$\sum_{v \in V} O(d + |X_v|)3^{d/3})$$

$$= O((dn + m)3^{d/3})$$

$$= O(dn3^{d/3})$$

$$\sum_{v \in V} O(d + |X_v|)3^{d/3})$$

$$= O((dn + m)3^{d/3})$$

$$= O(dn3^{d/3})$$

$$= O(f(d)n) \quad \text{where } f(d) = d3^{d/3}$$

Our running time: $O(dn3^{d/3})$

Our running time: $O(dn3^{d/3})$

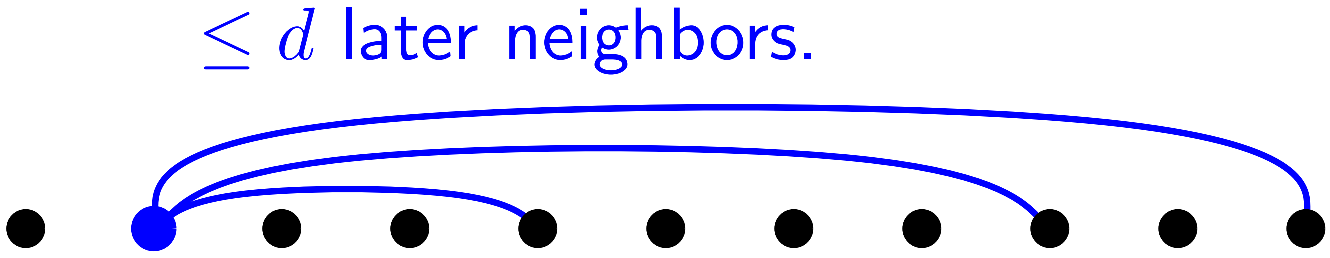
Worst-case output size: $O(d(n-d)3^{d/3})$

Our running time: $O(dn3^{d/3})$

Worst-case output size: $O(d(n-d)3^{d/3})$

When $n - d = \Omega(n)$, our algorithm is worst-case optimal.

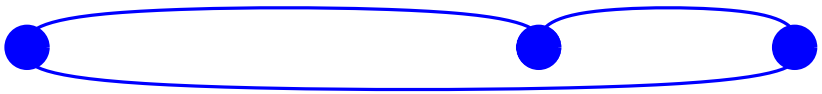
An upper bound



An upper bound

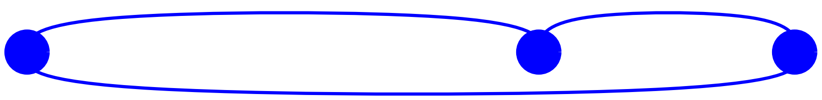
An upper bound

$\leq d$ later neighbors.



An upper bound

$\leq d$ later neighbors.



at most $O(3^{d/3})$ maximal cliques

An upper bound

$\leq d$ later neighbors.

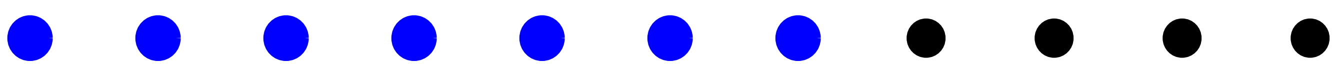
at most $O(3^{d/3})$ maximal cliques

An upper bound

An upper bound

$(n-d-3)3^{d/3}$ $\quad$ $3^{\frac{d+3}{3}}$

$n-d-3$ vertices $\quad$ $d+3$ vertices

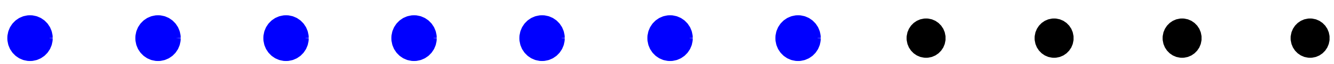


An upper bound

at most $(n-d)3^{d/3}$ maximal cliques

$(n-d-3)3^{d/3}$ $3^{\frac{d+3}{3}}$

$n-d-3$ vertices $d+3$ vertices

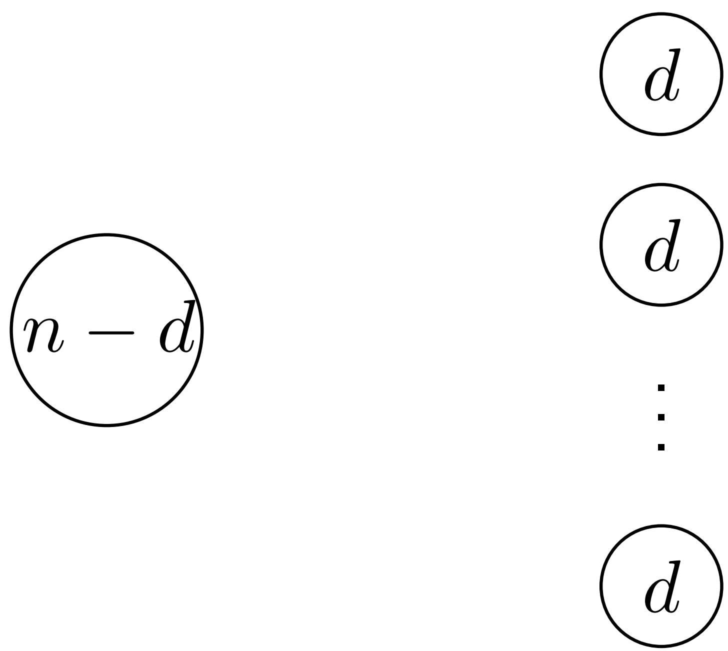


A lower bound

$K_{n-d,3,3,3,\ldots}$

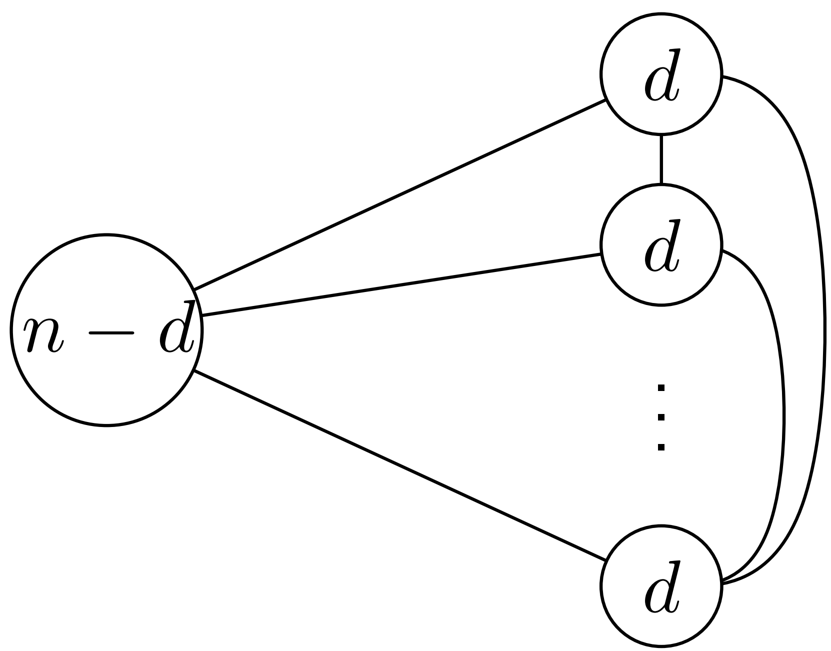
A lower bound

$K_{n-d,3,3,3,\ldots}$



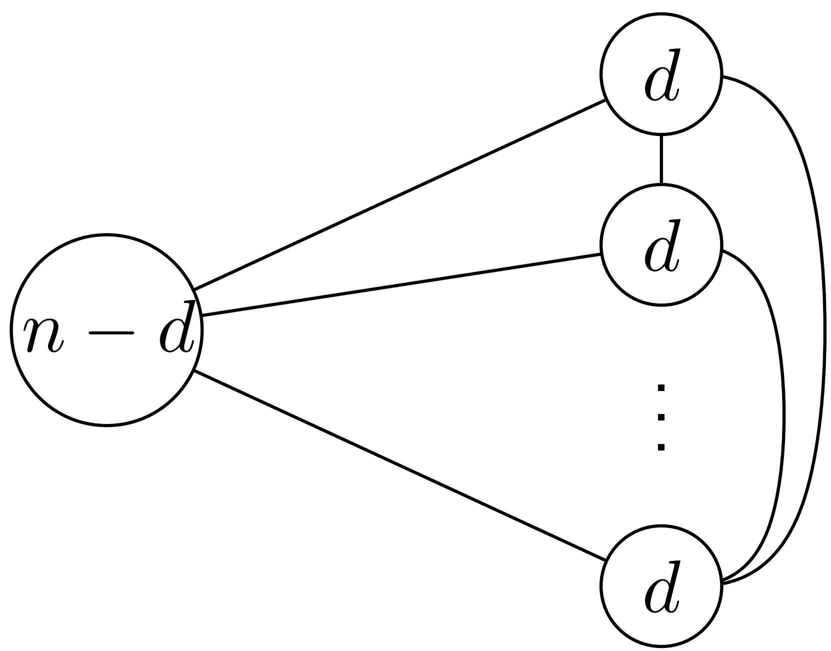
A lower bound

$$K_{n-d,3,3,3,\ldots}$$



A lower bound

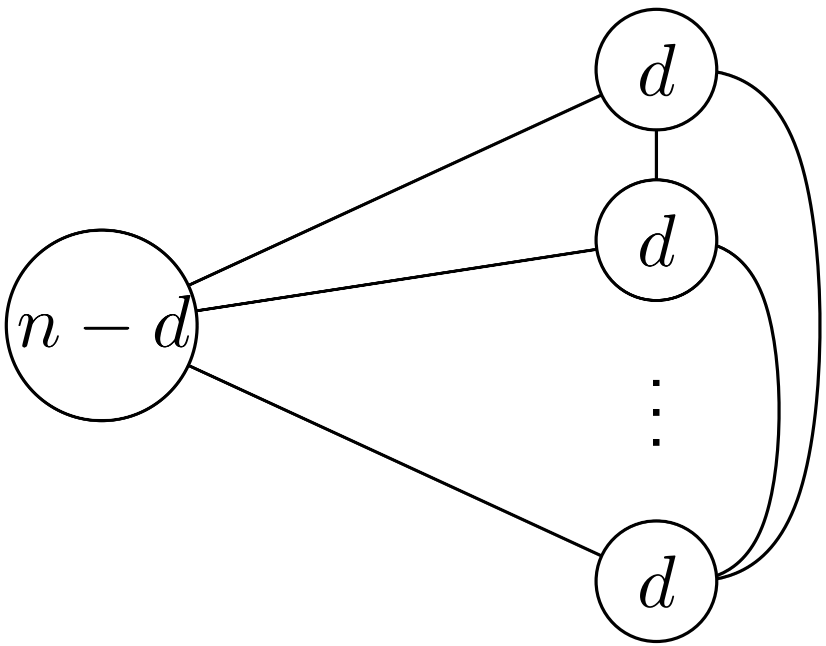
$K_{n-d,3,3,3,\ldots}$



$(n-d)3^{d/3}$
maximal cliques

A lower bound

$$K_{n-d,3,3,3,\ldots}$$



$(n-d)3^{d/3}$ maximal cliques

has degeneracy d when $(n-d) \geq 3$

at most $(n-d)3^{d/3}$ maximal cliques

each clique is of size at most $d+1$

$O(d(n-d)3^{d/3})$ worst-case output size.

The Bron–Kerbosch Algorithm

proc BronKerbosch(P, R, X)

if $P \cup X = \emptyset$ **then**
    report R as a maximal clique
end if
for each vertex $v \in P$ **do**
    BronKerbosch($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for

The Bron–Kerbosch Algorithm

proc BronKerbosch(P, R, X)

if $P \cup X = \emptyset$ **then**

  report R as a maximal clique

end if

for each vertex $v \in P$ (in degeneracy order) **do**

  BronKerbosch($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)

  $P \leftarrow P \setminus \{v\}$

  $X \leftarrow X \cup \{v\}$

end for

The Bron–Kerbosch Algorithm

proc BronKerbosch(P, R, X)

if $P \cup X = \emptyset$ **then**

  report R as a maximal clique

end if

for each vertex $v \in P$ (in degeneracy order) **do**

  BronKerboschPivot($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)

  $P \leftarrow P \setminus \{v\}$

  $X \leftarrow X \cup \{v\}$

end for

Experiments

Linux Workstation: 3.00Ghz Pentium 4, 1GB RAM

Experimental results for UCI data sets

graph	d	BK	BK-pivot	BK-hybrid	BK-degen	Uno
karate	4	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec	**$<$1sec***
dolphins	4	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec
power	5	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec
polbooks	6	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec
adjnoun	6	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec
football	8	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec
lesmis	9	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec
celegens	9	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec	**seg. fault***
netscience	19	2.8sec	$<$ 1sec	$<$ 1sec	$<$ 1sec	$<$ 1sec
internet	25	19.4sec	10.3sec	**$<$ 1sec**	**$<$ 1 sec**	**$<$1sec***
condmat	29	$>$ 3min	65sec	**1.6sec**	**2.61sec**	$<$ 1sec
polblogs	36	$>$ 3min	2sec	1.5sec	1.2sec	1.8sec
astro	56	$>$ 3min	12.3sec	**1.4sec**	**3.14sec**	$<$ 1sec

Experimental results for BIOGRID data sets (PPI Networks)

graph	d	BK	BK-pivot	BK-hybrid	BK-degen	Uno
mouse	6	< 1sec	< 1sec	< 1sec	< 1sec	< 1sec
worm	10	< 1sec	< 1sec	< 1sec	< 1sec	< 1sec
plant	12	< 1sec	< 1sec	< 1sec	< 1sec	< 1sec
fruitfly	12	< 1sec	2.2sec	< 1sec	< 1sec	< 1sec
human	12	1.4sec	3.3sec	< 1sec	< 1sec	< 1sec
fission-yeast	34	2.8sec	1.1sec	< 1sec	< 1sec	< 1sec
yeast	64	> 3min	81sec	44.3sec	**20.5sec**	**121.1sec***

Experimental results for Pajek data sets

graph	d	BK	BK-pivot	BK-hybrid	BK-degen	Uno
foldoc	12	4.2sec	9sec	$<$ 1sec	1sec	$<$ 1sec
patents	24	$>$ 5min	$>$ 5min	**4.3sec**	**5.3sec**	2.2sec
eatRS	34	19.8sec	53sec	12.3sec	9.12sec	14.9sec
hep-th	37	$>$ 5min	69.6sec	**22.6sec**	**17.2sec**	**41.5sec***
days-all	73	$>$ 5min	379.1sec	206.5sec	**51.4sec**	10min 25sec
ND-www	155	$>$ 5min	$>$ 5min	**27.8sec**	**41.11sec**	**9.7sec***

Thank you!