Near-optimal Fixed-parameter Tractability of the Bron–Kerbosch Algorithm for Maximal Cliques

Darren Strash
Department of Computer Science
UC Irvine

Joint work with David Eppstein and Maarten Löffler

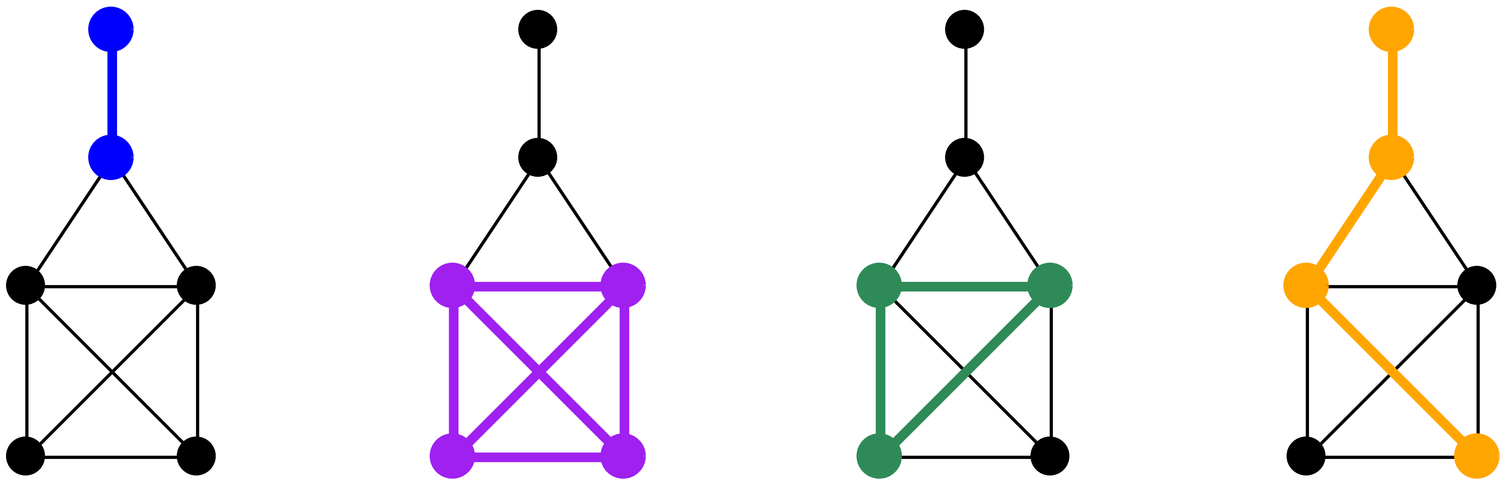
What is a Maximal Clique?

A clique that cannot be made bigger by adding more vertices

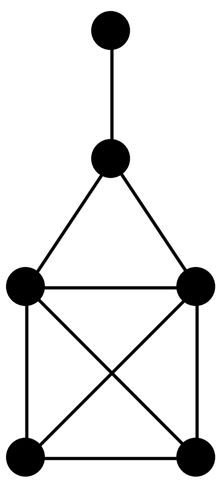
What is a Maximal Clique?

A clique that cannot be made bigger by adding more vertices

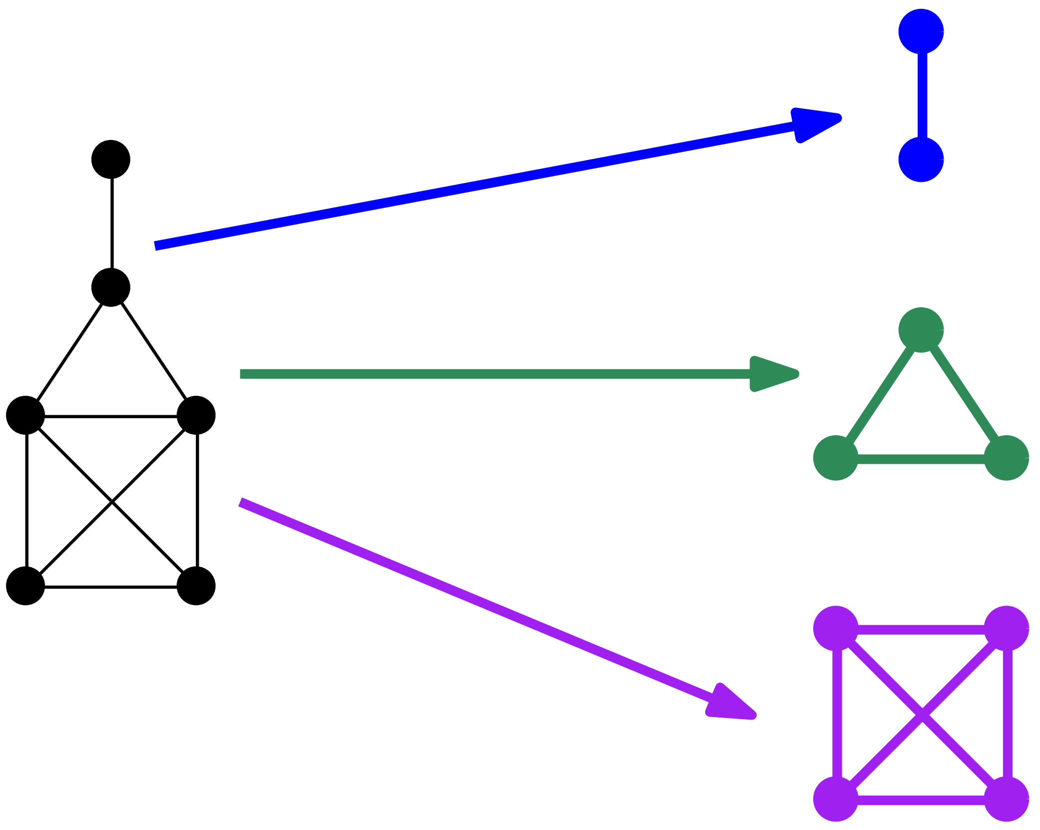
Maximal | Maximal, Maximum | Not Maximal | Not Clique



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

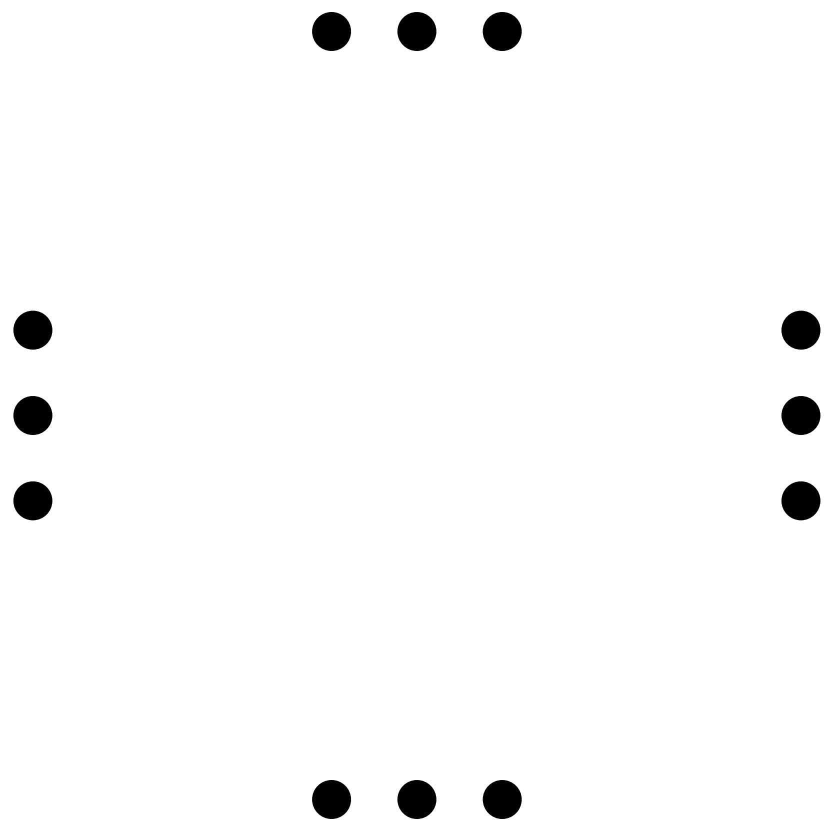
Detect structural motifs from similarities between proteins

Determine the docking regions between biomolecules

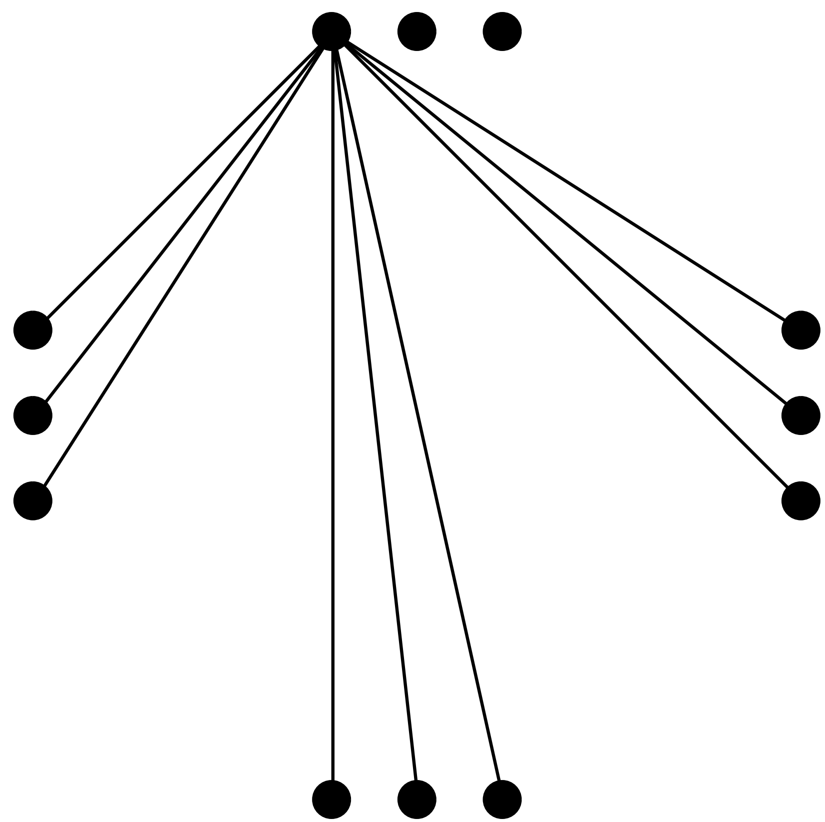
Document clustering for information retrieval

There may be many maximal cliques.

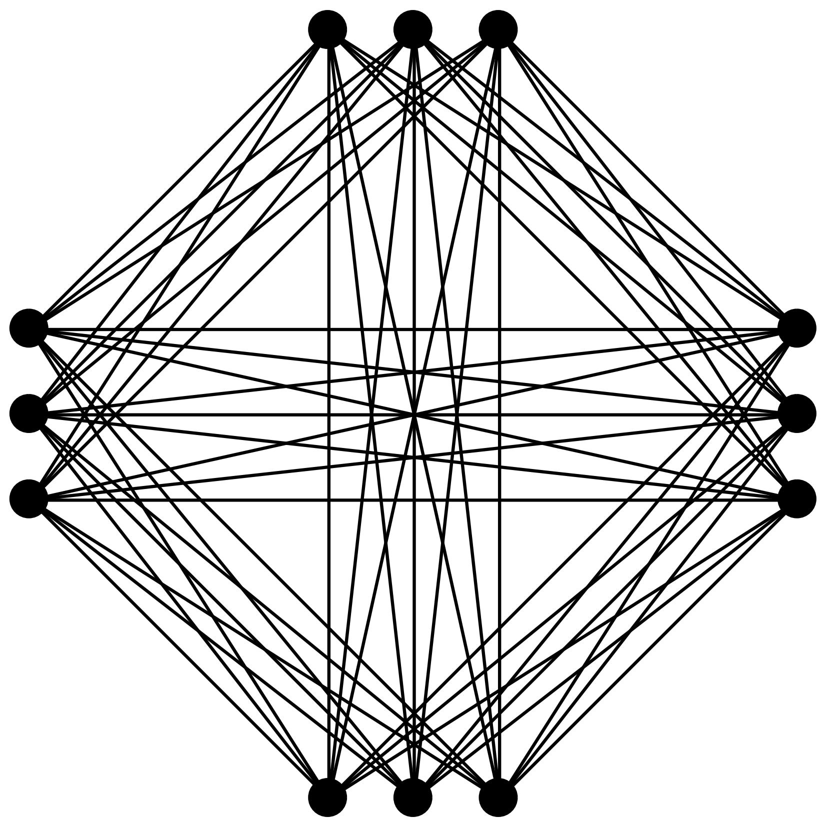
There may be many maximal cliques.



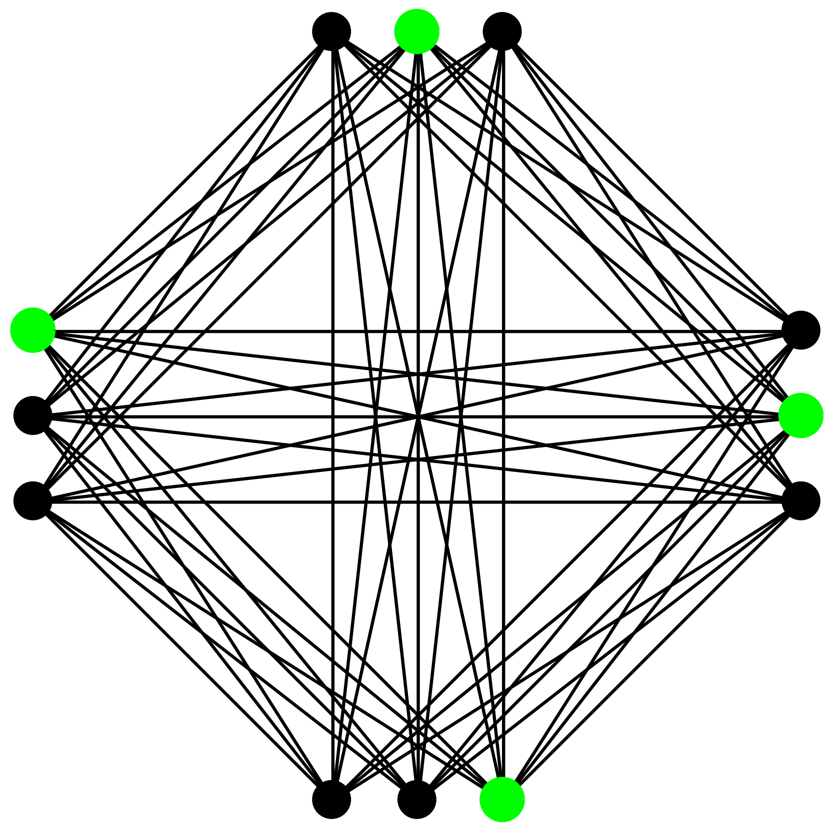
There may be many maximal cliques.



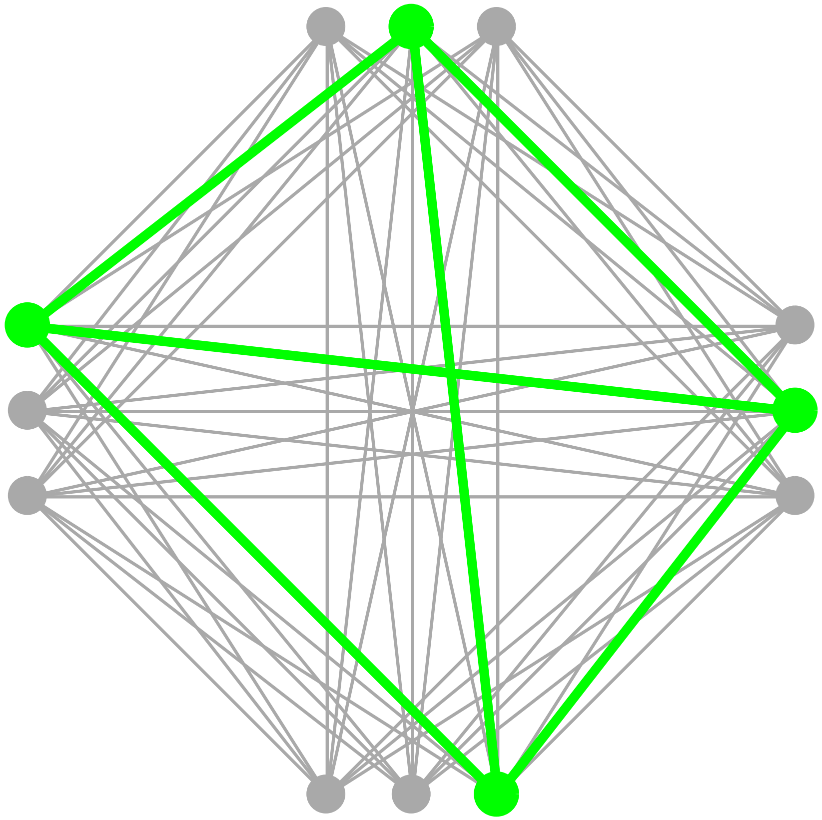
There may be many maximal cliques.



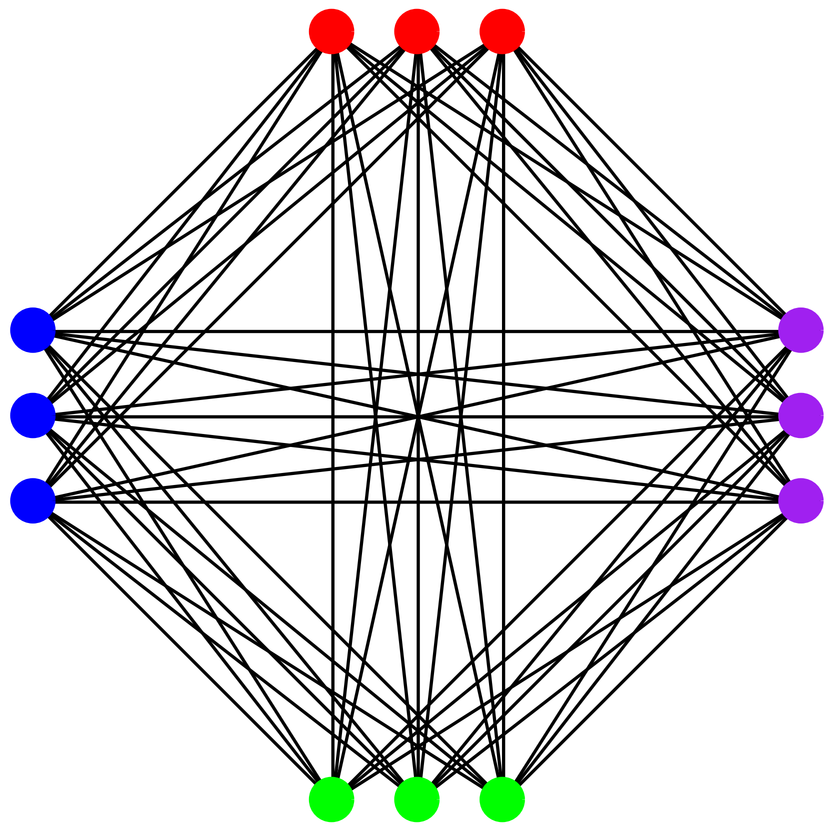
There may be many maximal cliques.



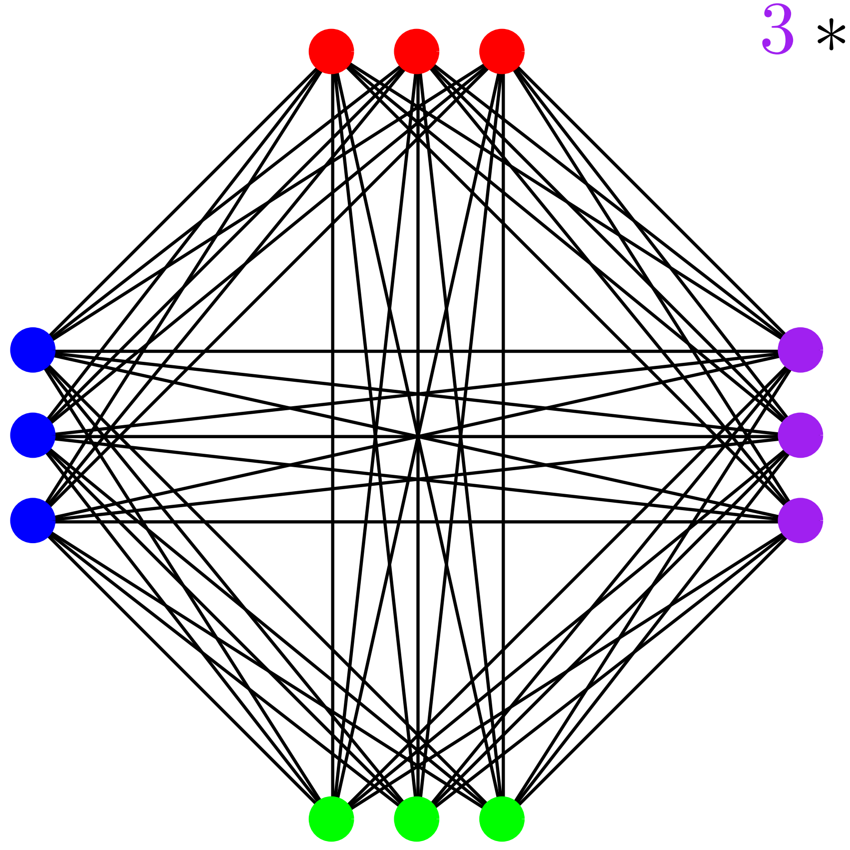
There may be many maximal cliques.



There may be many maximal cliques.

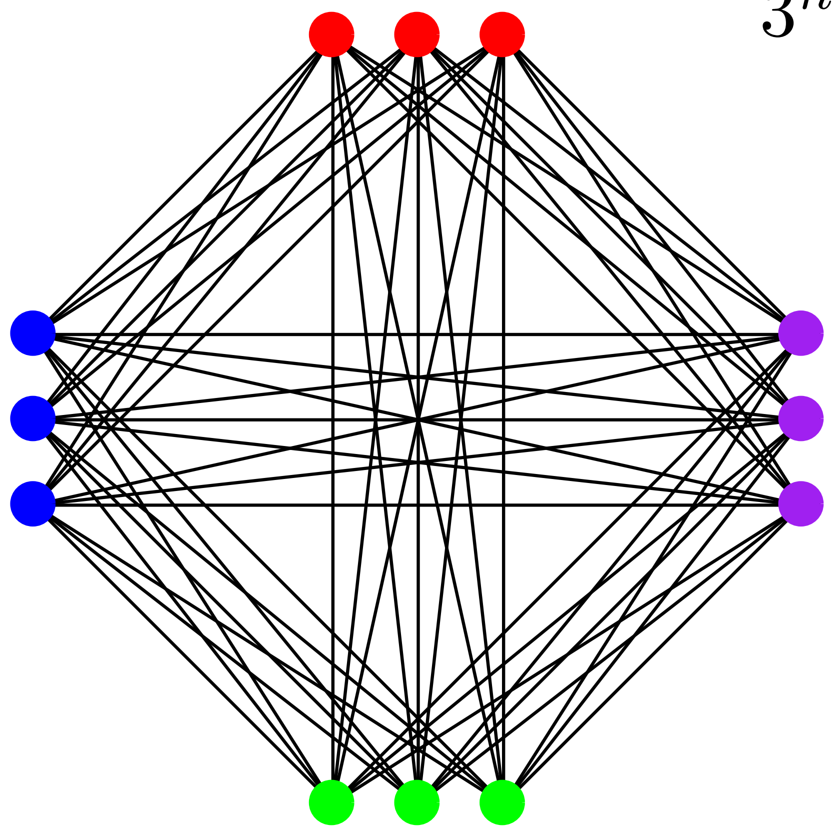


There may be many maximal cliques.



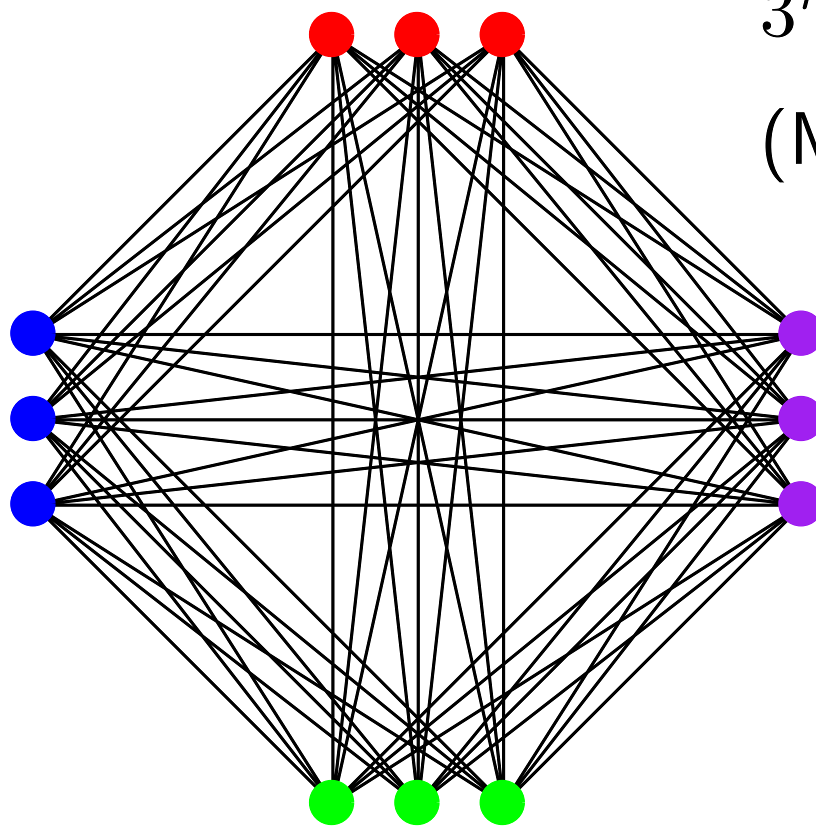
$3 * 3 * 3 * 3$

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**

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

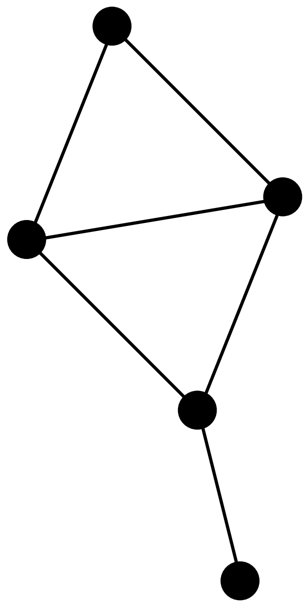
There are many heuristics, which make it faster

Its variations work well in practice.

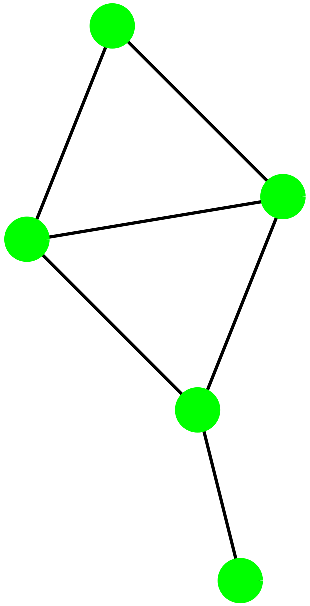
Confirmed through computational experiments
Johnston (1976), Koch (2001), Baum (2003)

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

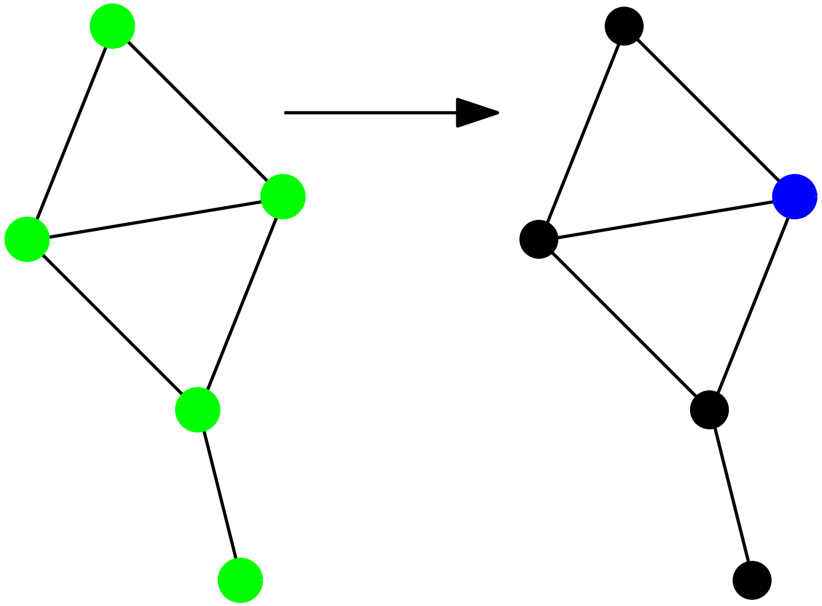
Finding one maximal clique



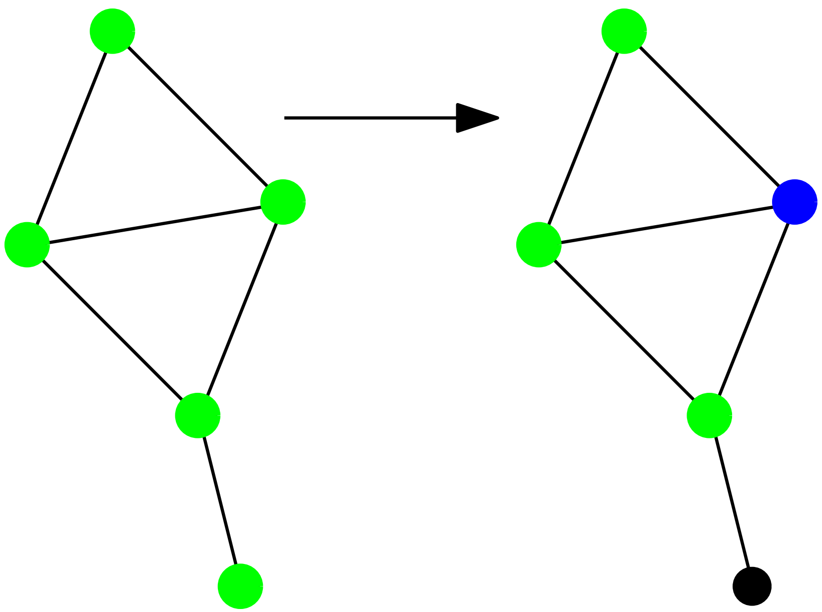
Finding one maximal clique



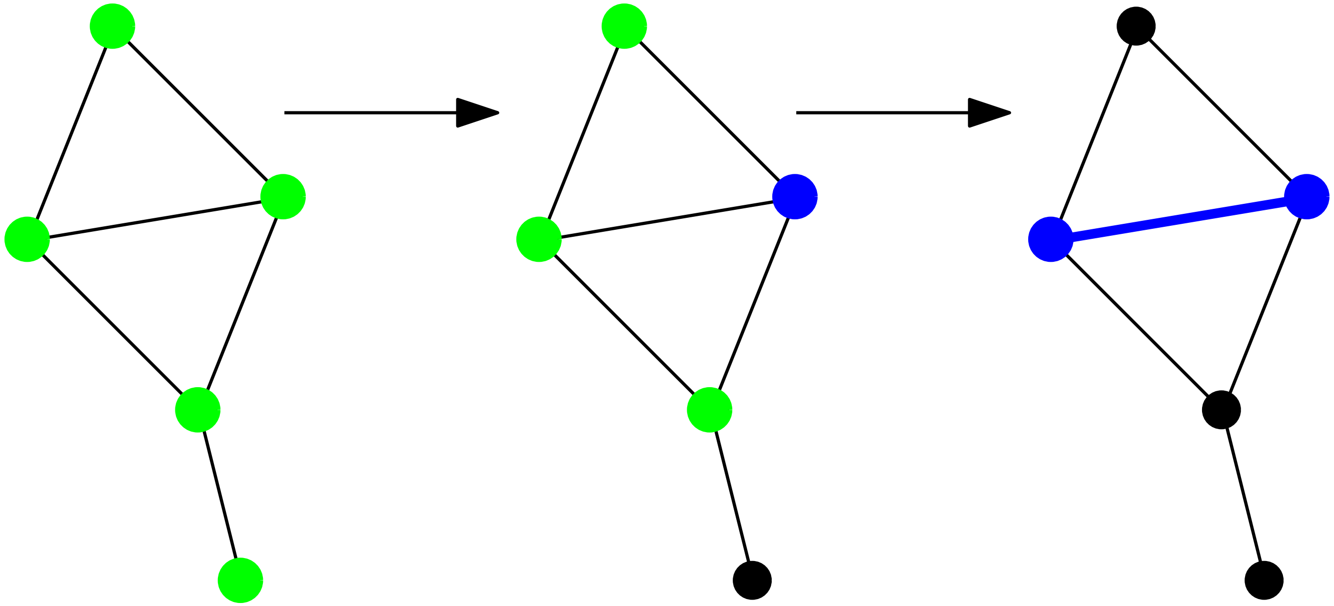
Finding one maximal clique



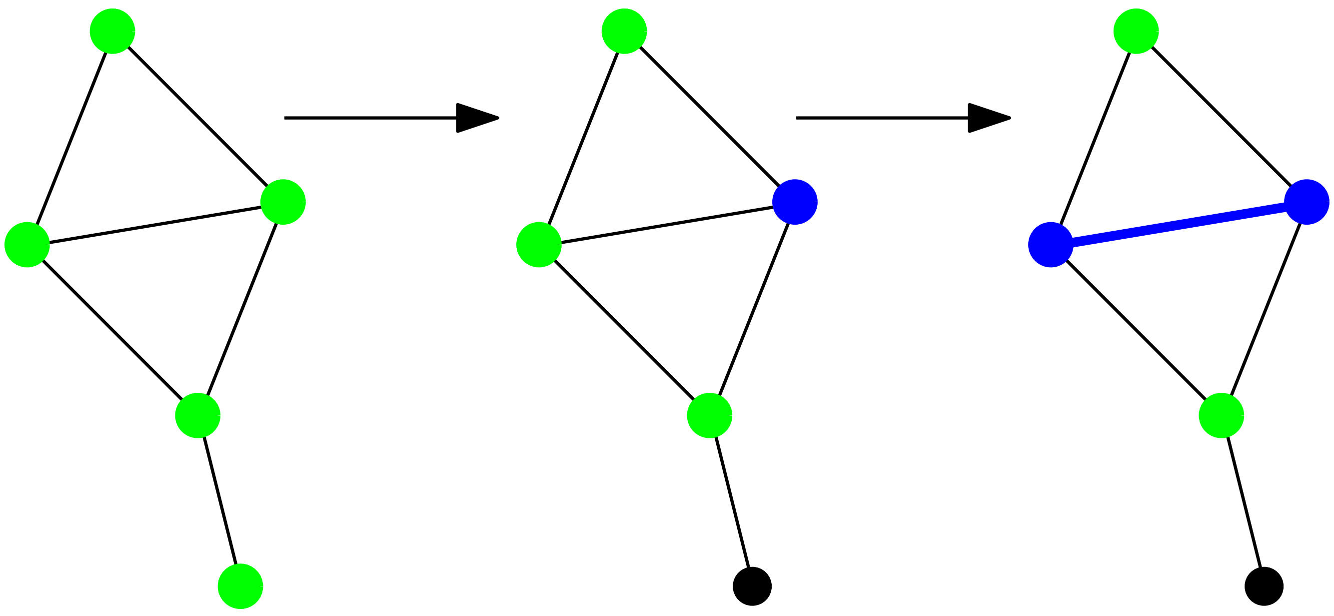
Finding one maximal clique



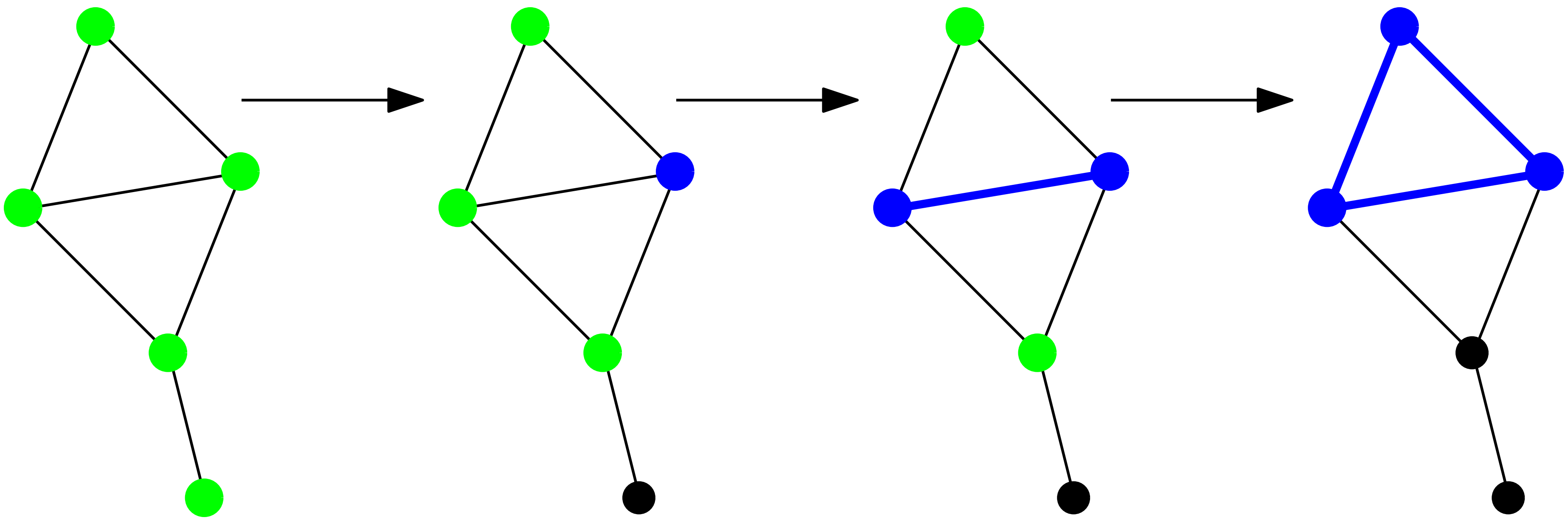
Finding one maximal clique



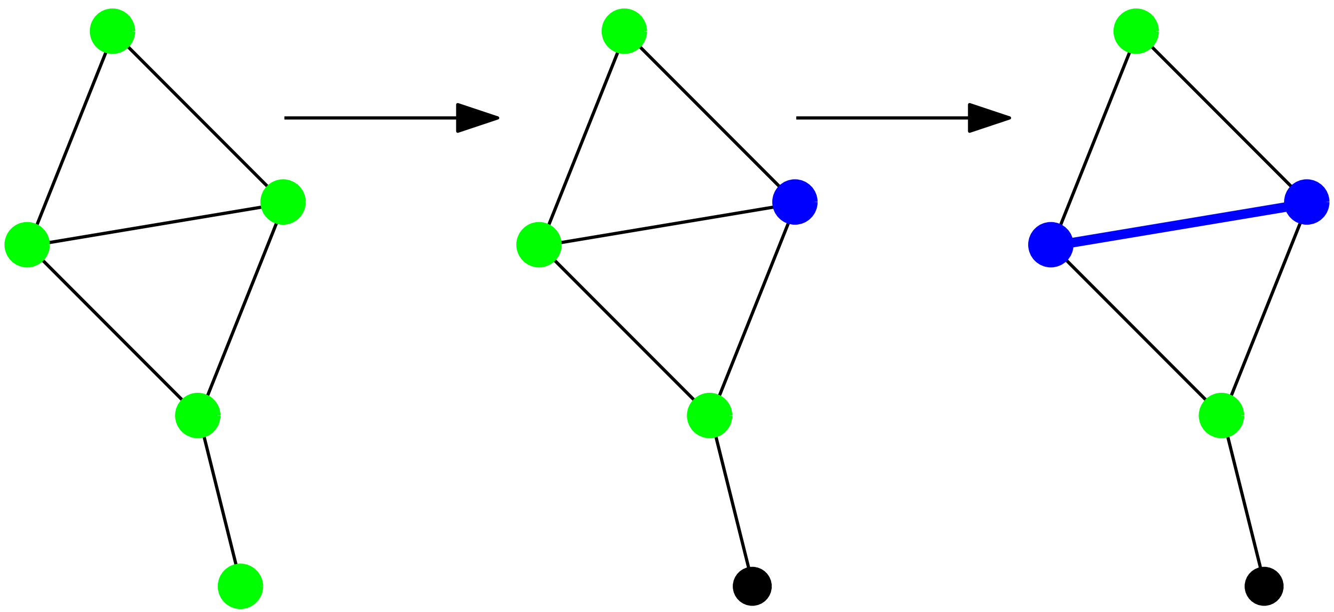
Finding one maximal clique



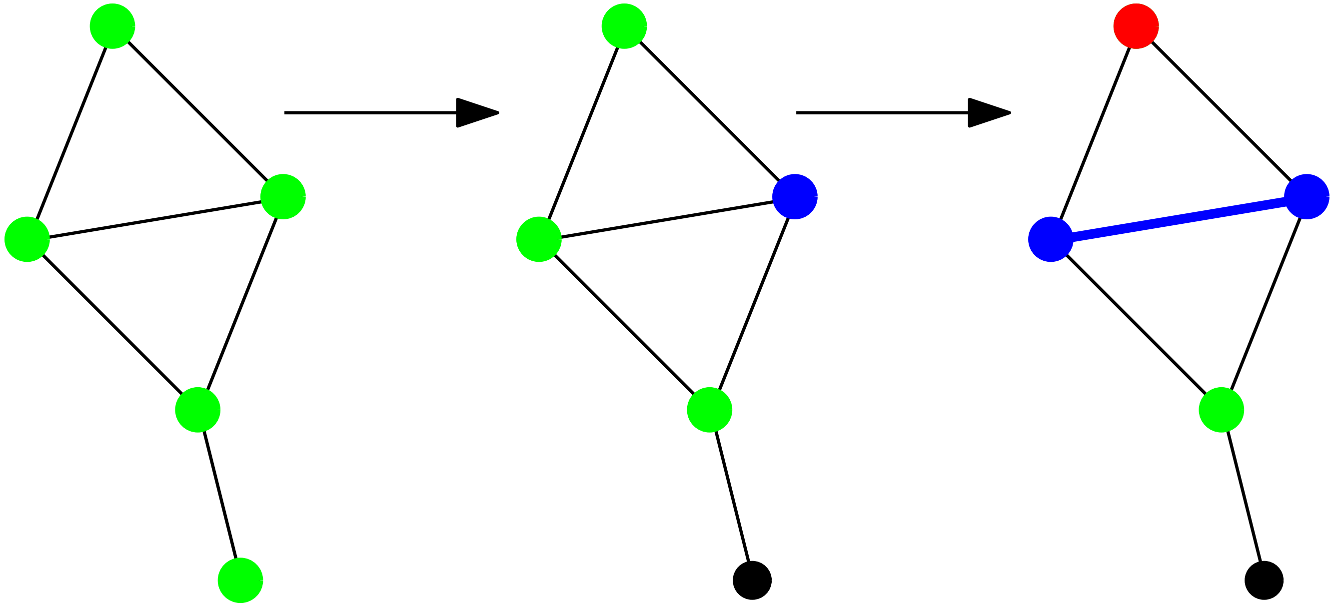
Finding one maximal clique



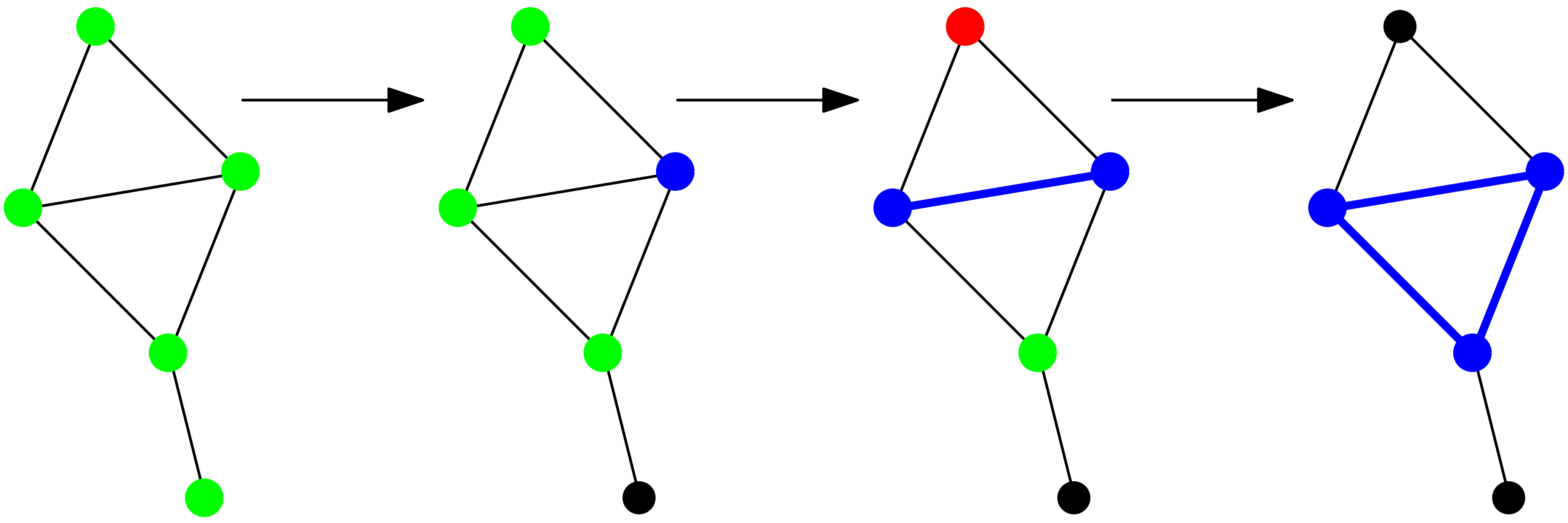
The Bron–Kerbosch Algorithm



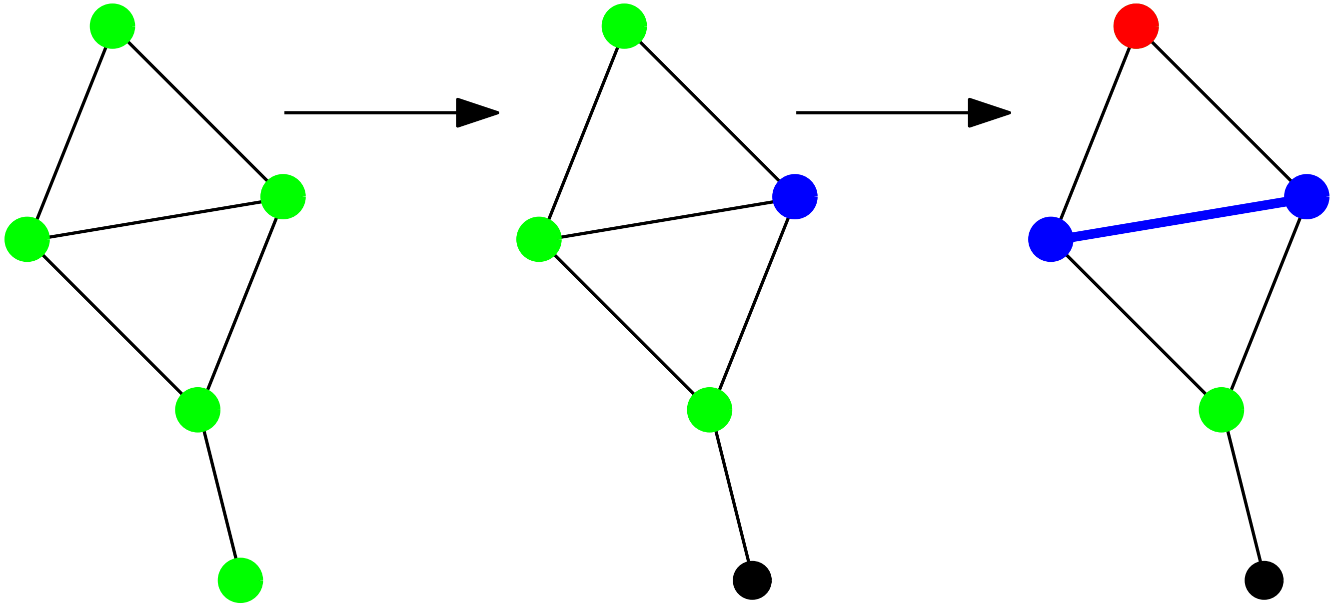
The Bron–Kerbosch Algorithm



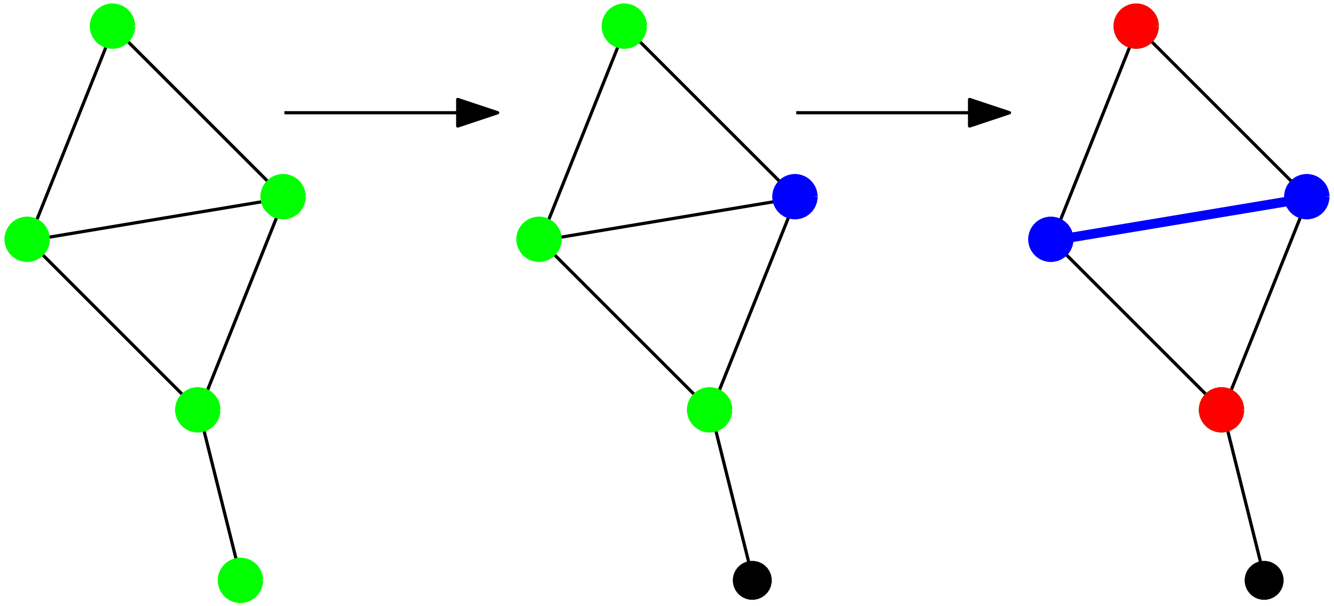
The Bron–Kerbosch Algorithm



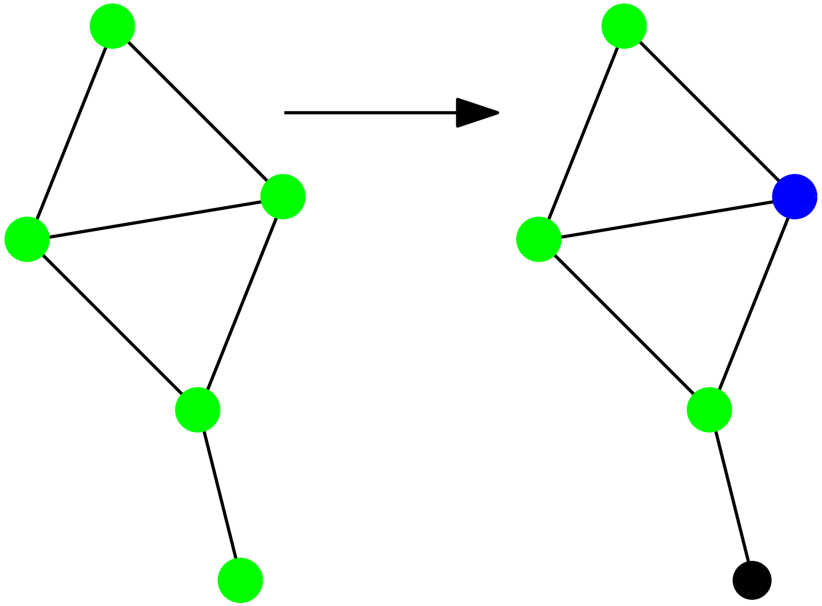
The Bron–Kerbosch Algorithm



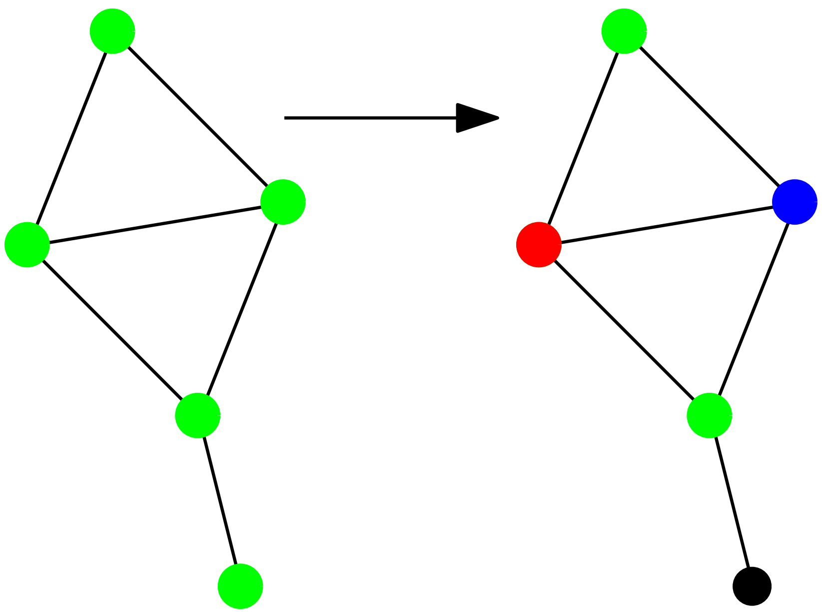
The Bron–Kerbosch Algorithm



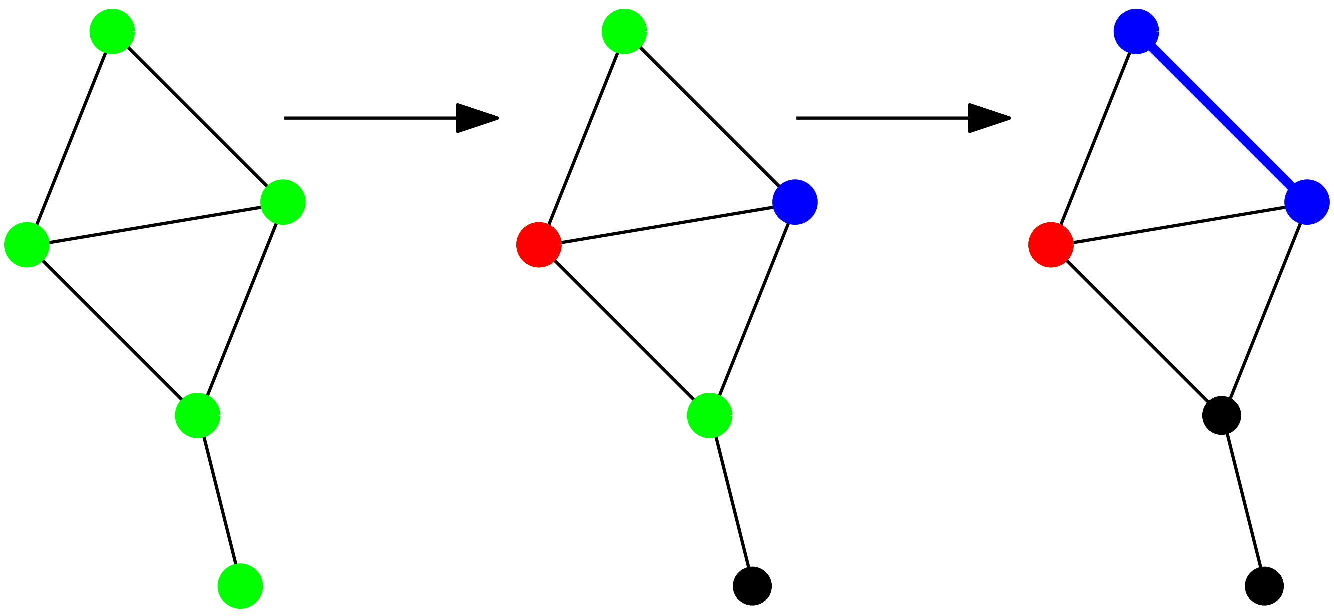
The Bron–Kerbosch Algorithm



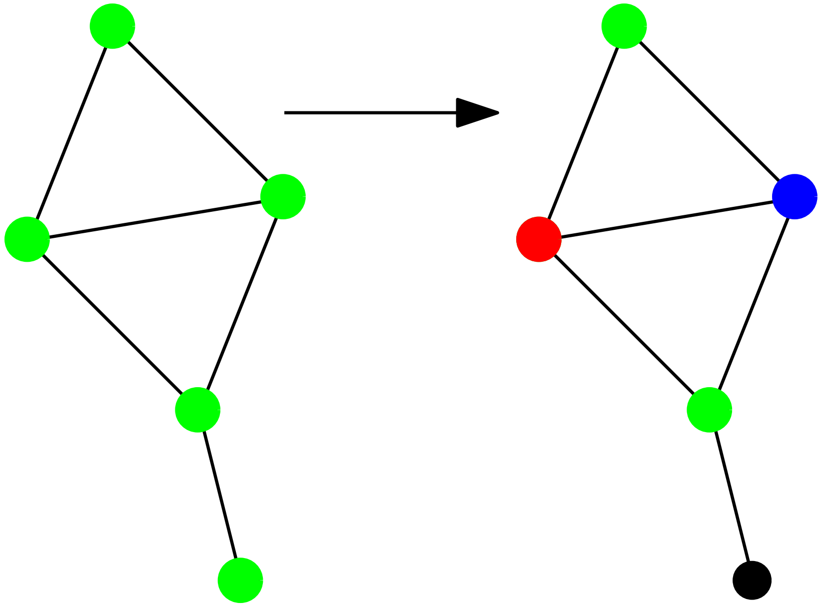
The Bron–Kerbosch Algorithm



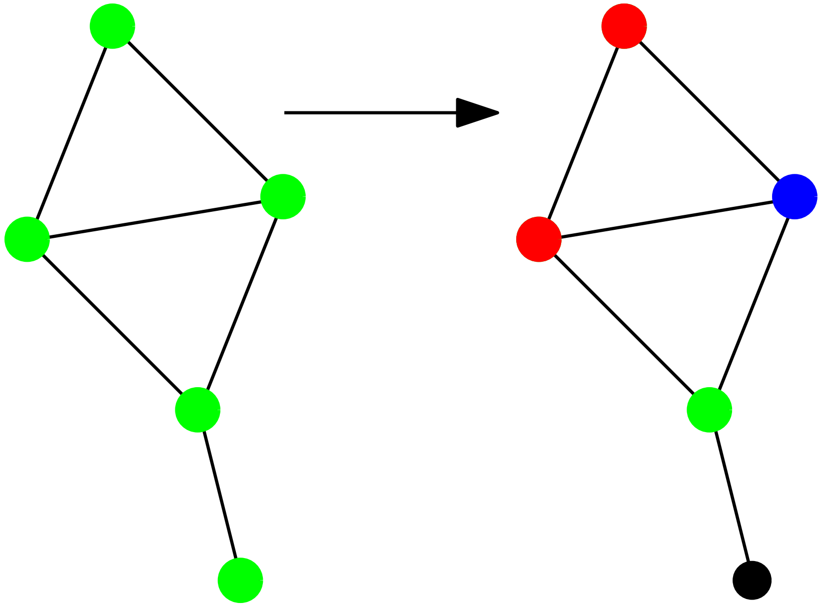
The Bron–Kerbosch Algorithm



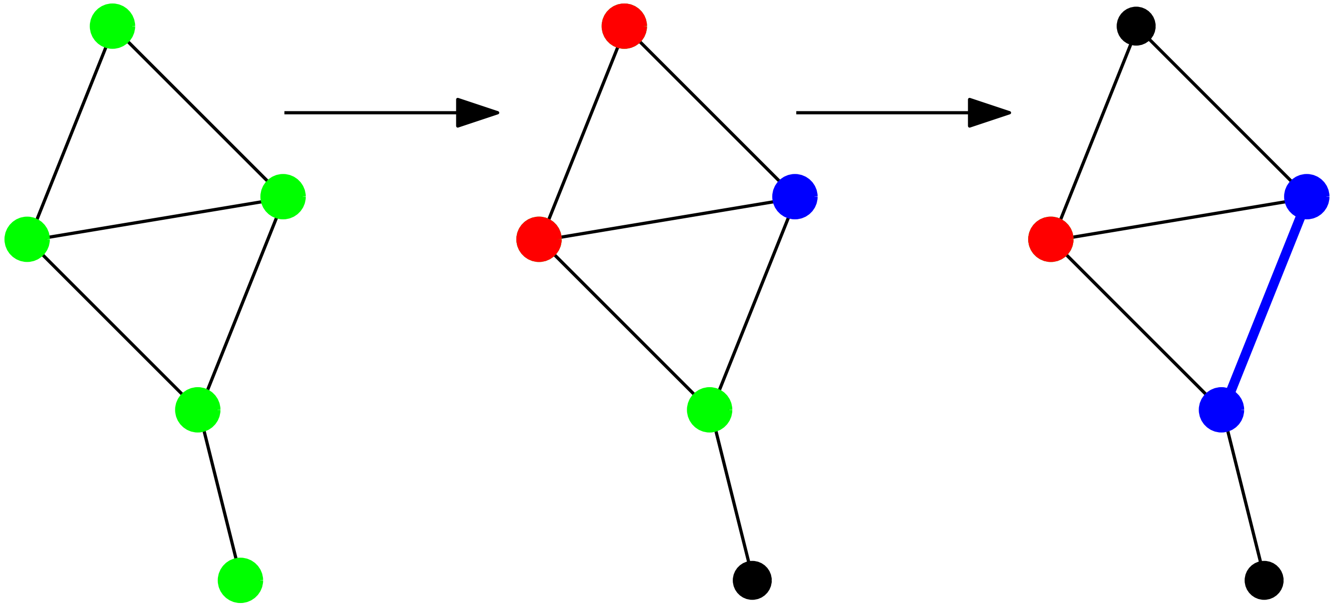
The Bron–Kerbosch Algorithm



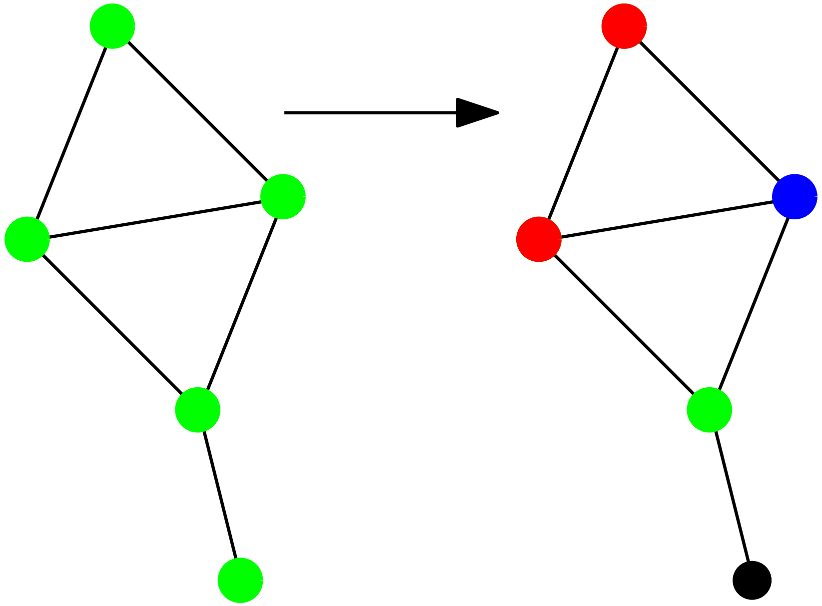
The Bron–Kerbosch Algorithm



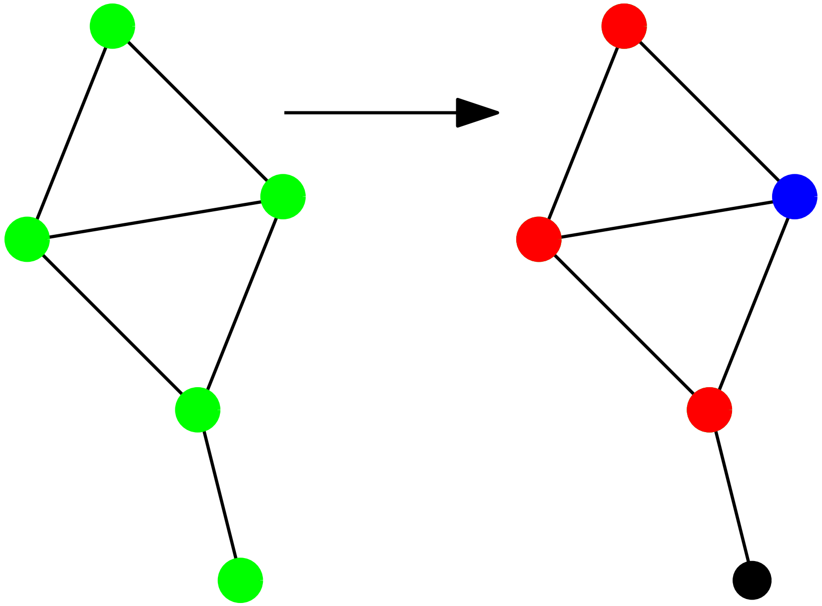
The Bron–Kerbosch Algorithm



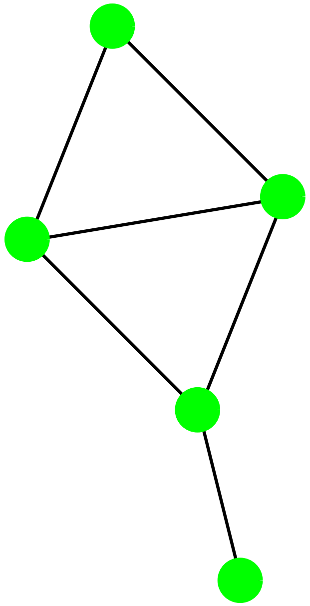
The Bron–Kerbosch Algorithm



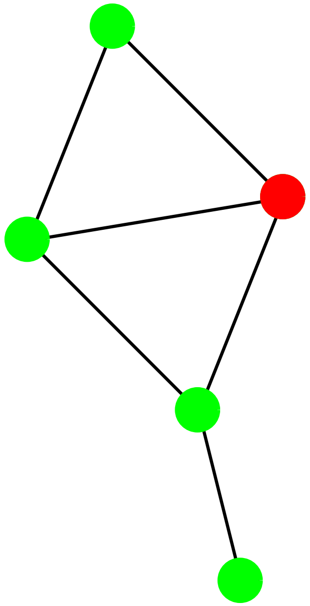
The Bron–Kerbosch Algorithm



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**

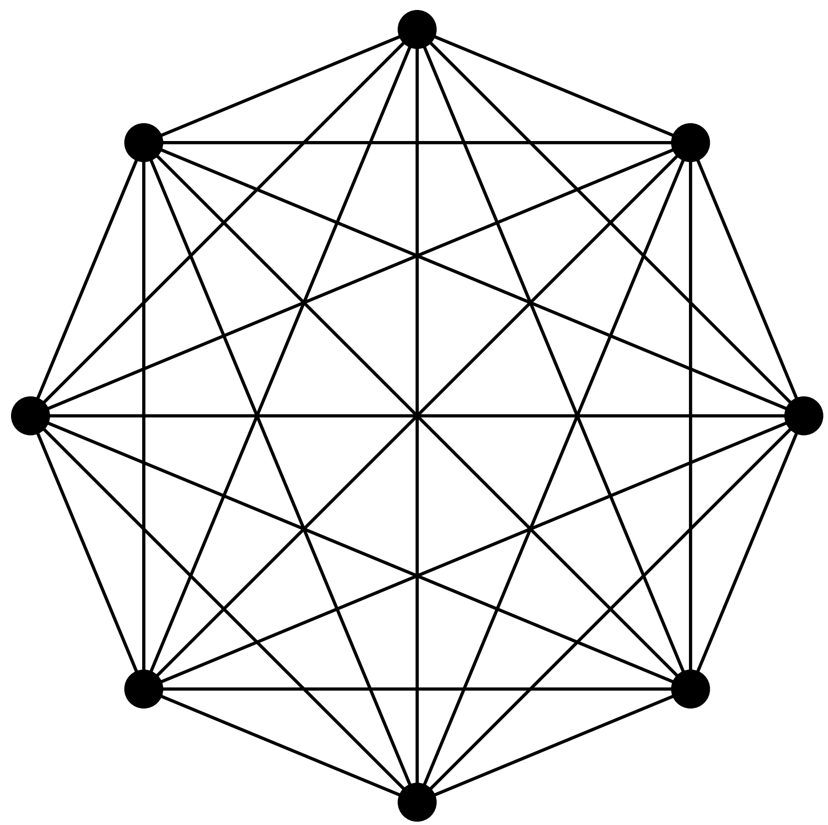
    BronKerbosch($P \cap \Gamma(v)$, $R \cup \{v\}$, $X \cap \Gamma(v)$)

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

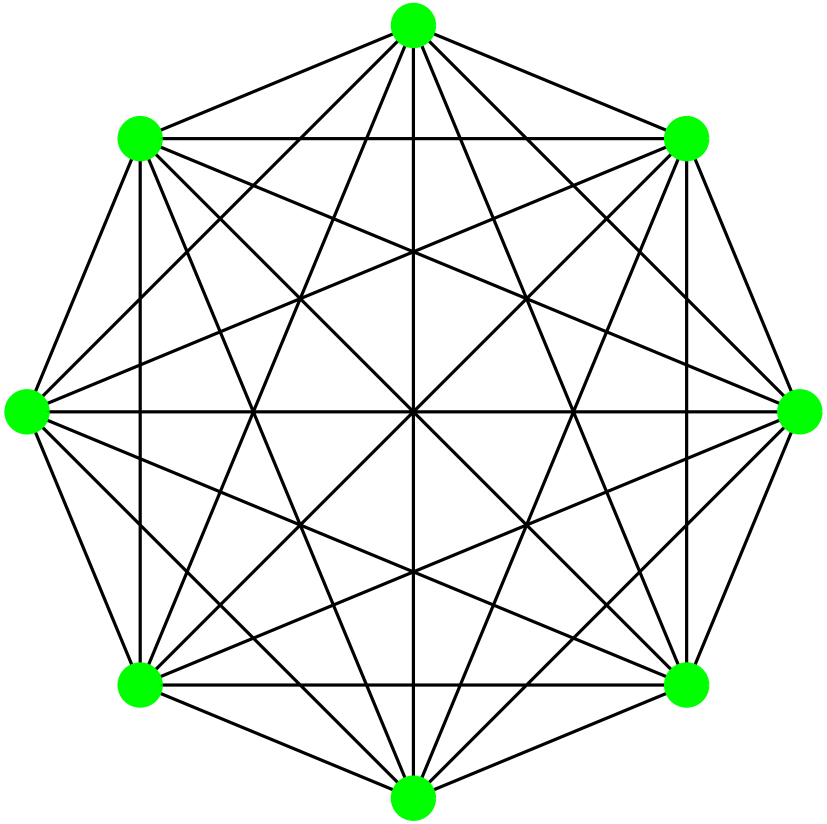
    $X \leftarrow X \cup \{v\}$

end for

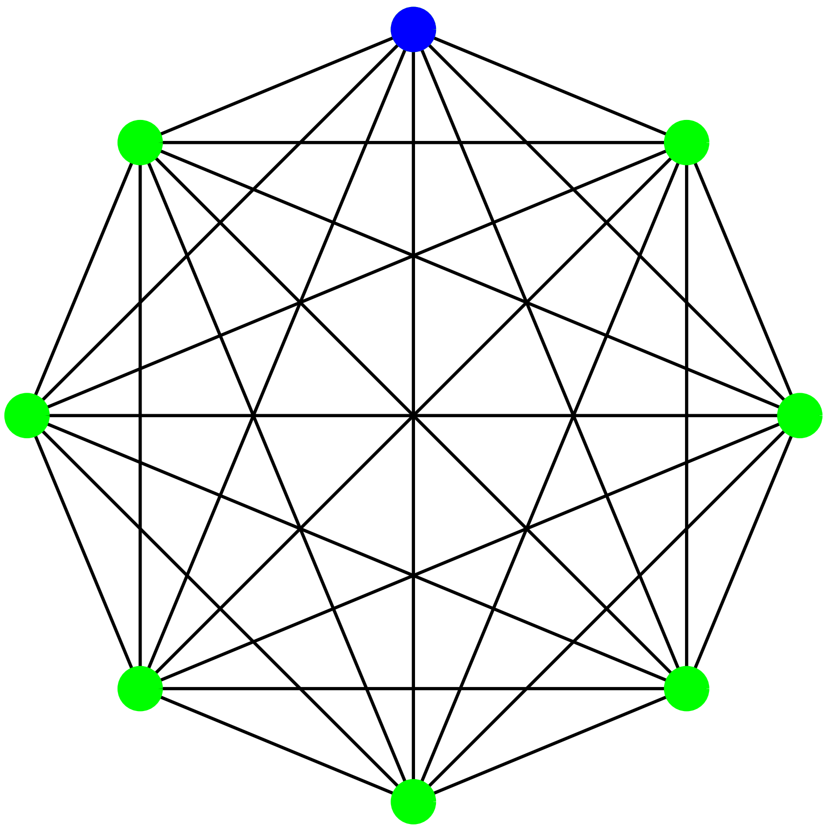
The Bron–Kerbosch Algorithm



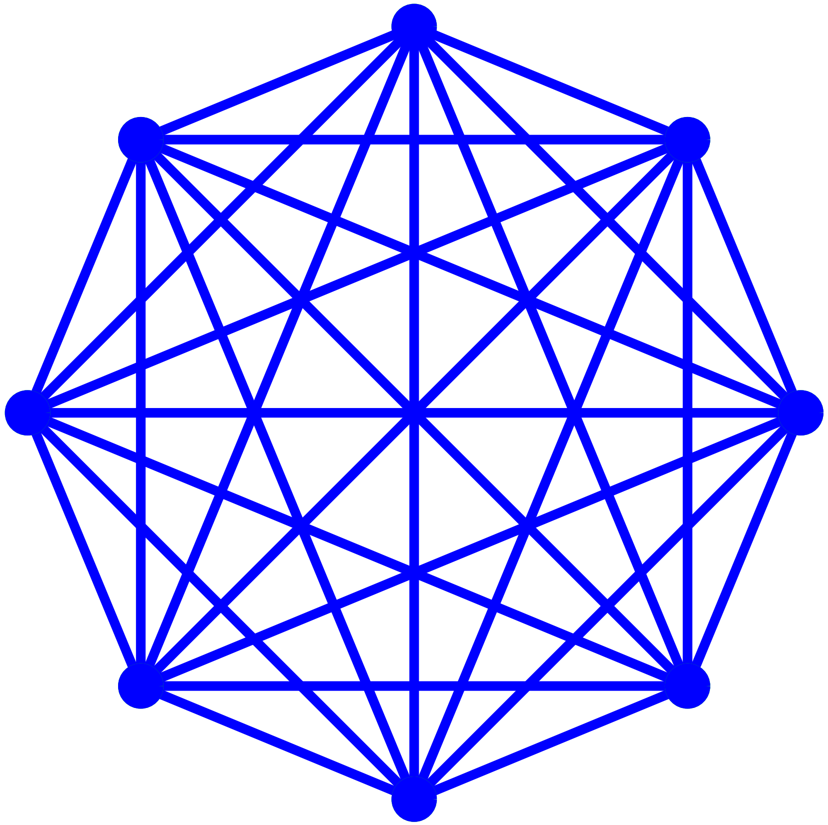
The Bron–Kerbosch Algorithm



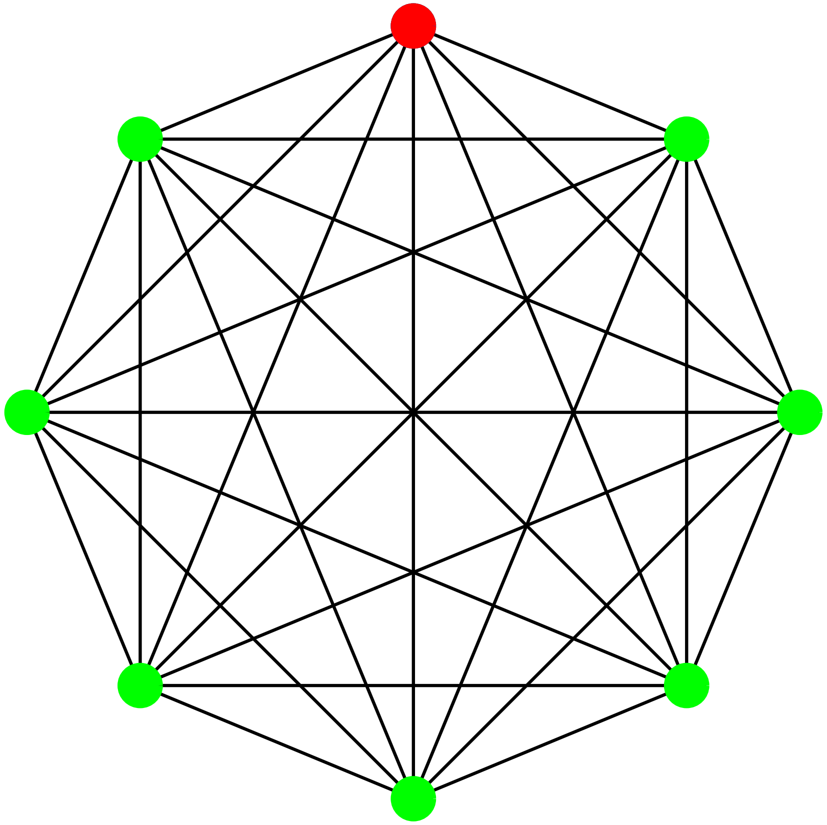
The Bron–Kerbosch Algorithm



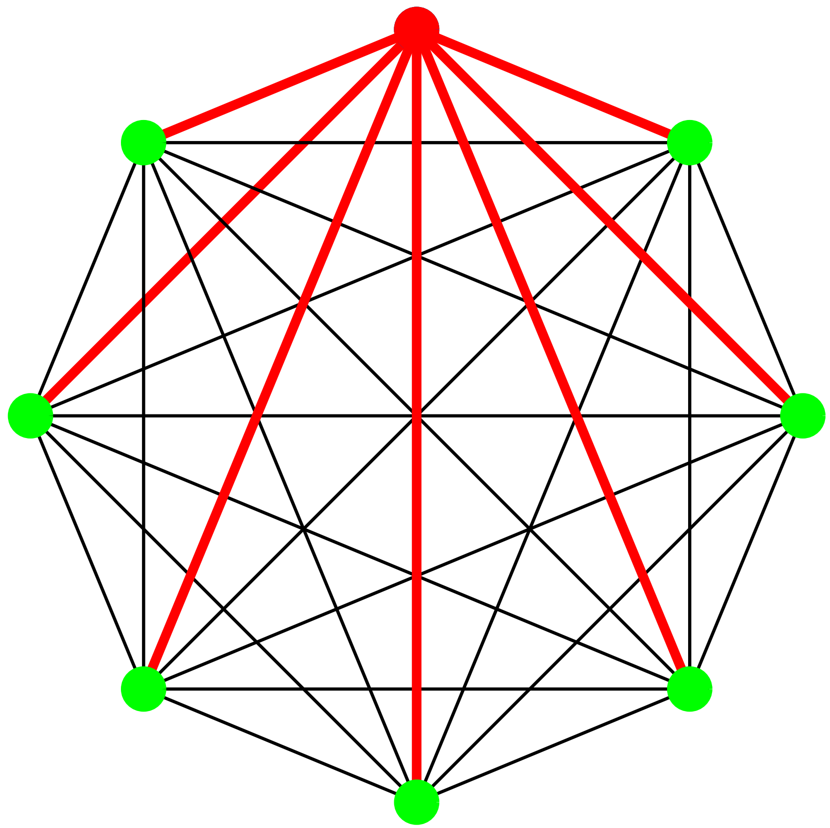
The Bron–Kerbosch Algorithm



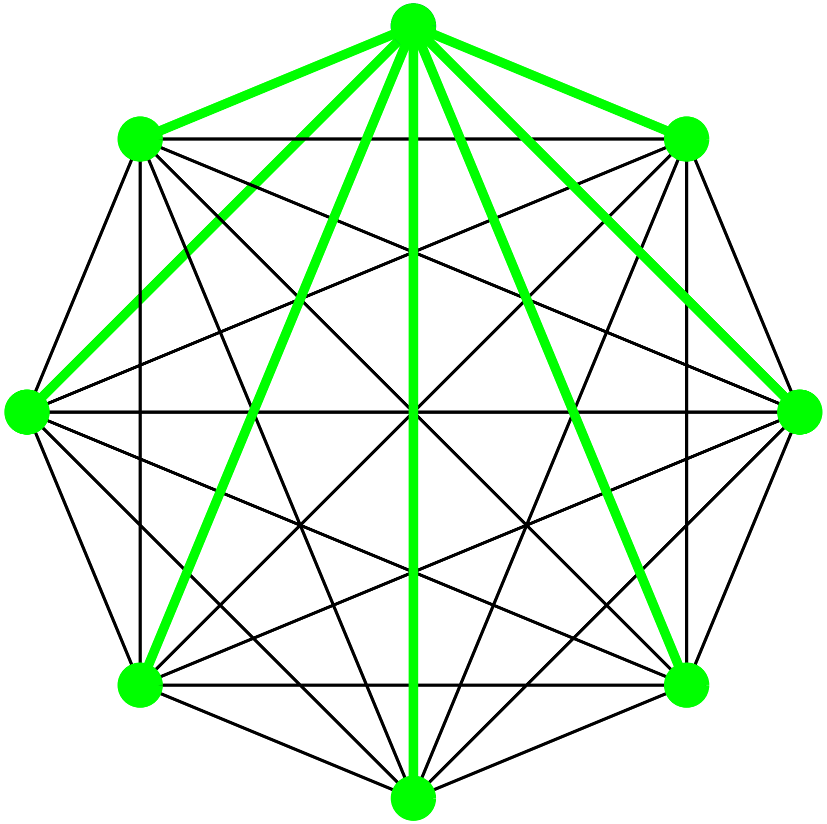
The Bron–Kerbosch Algorithm



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 ∪ X = ∅ then
    report R as a maximal clique
end if
choose a pivot u ∈ P ∪ X to minimize |P \ Γ(u)|
for each vertex v ∈ P \ Γ(u) do
    BronKerboschPivot(P ∩ Γ(v), R ∪ {v}, X ∩ Γ(v))
    P ← P \ {v}
    X ← X ∪ {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

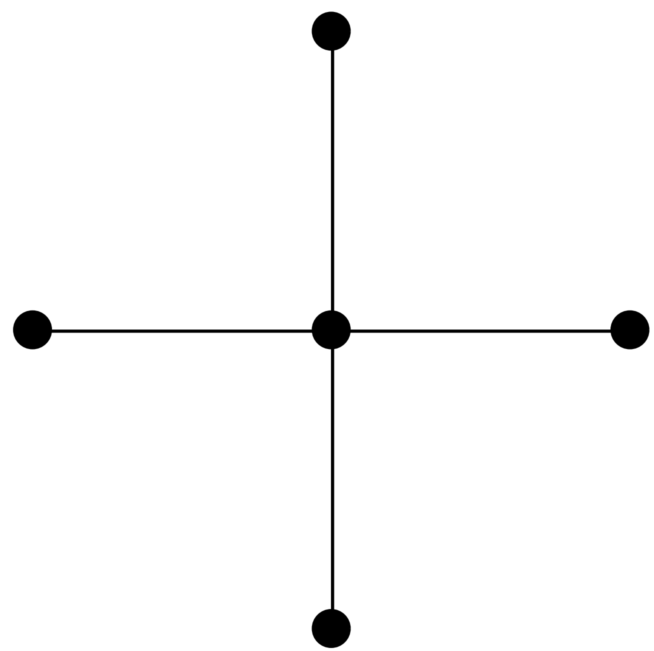
$$T(n) \leq \max_{k}\{kT(n-k)\} + O(n^2)$$

The Bron–Kerbosch Algorithm with Pivoting

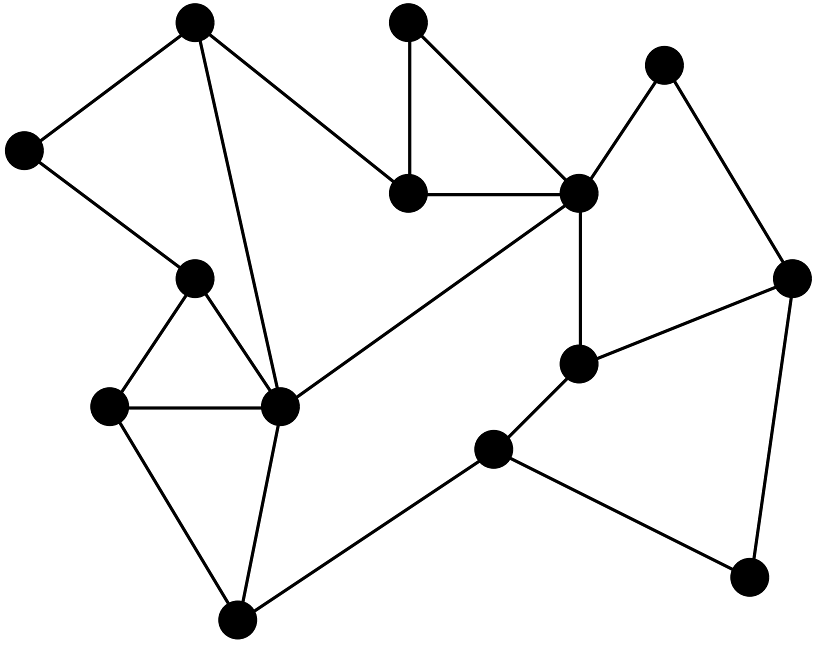
$$T(n) \leq \max_k \{kT(n-k)\} + O(n^2)$$

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

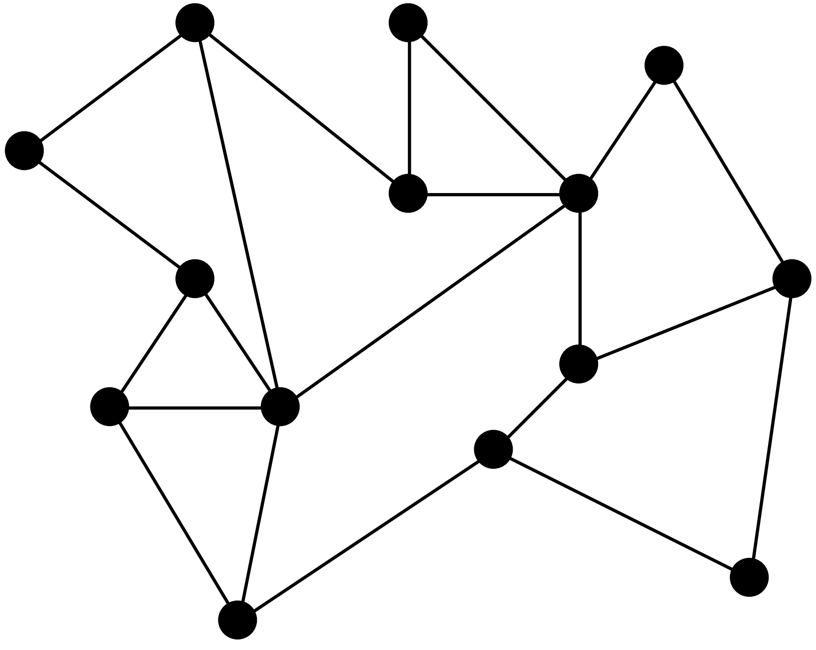
The Bron–Kerbosch Algorithm



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

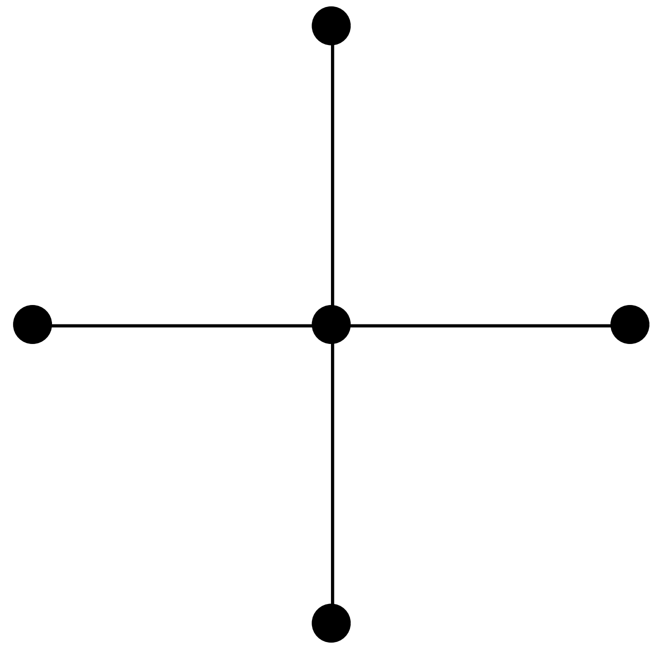
degeneracy:

Parameterize on Sparsity

degeneracy:

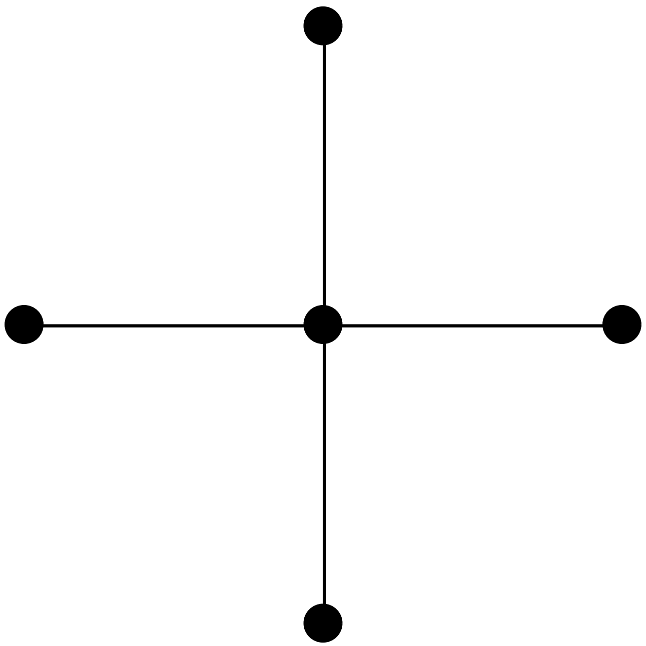
The minimum integer d such that every subgraph of G has a vertex of degree d or less.

Degeneracy



Degeneracy

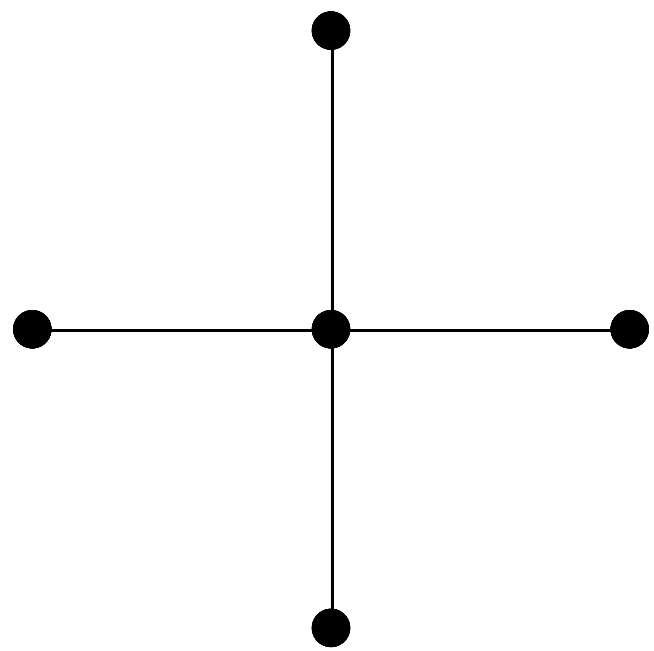
$d = 1$

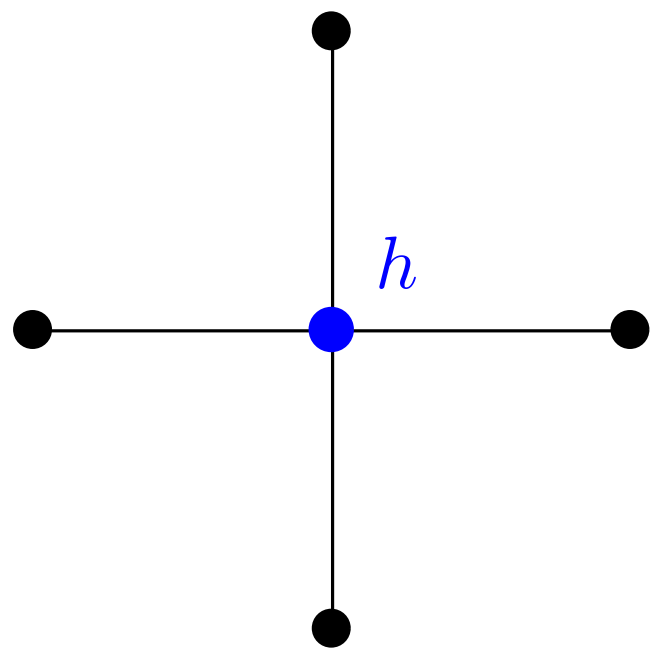


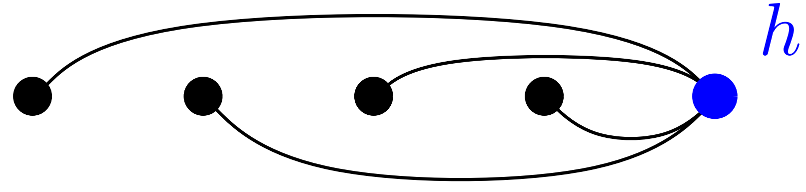
Degeneracy

degeneracy:

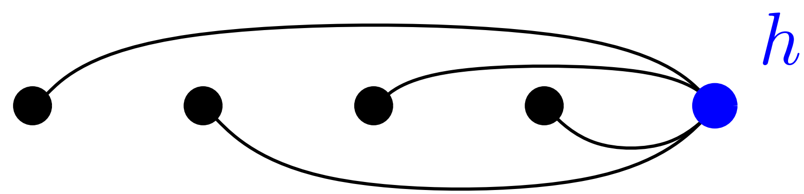
The minimum integer d such that there is an ordering of the vertices where each vertex has at most d neighbors later in the ordering.

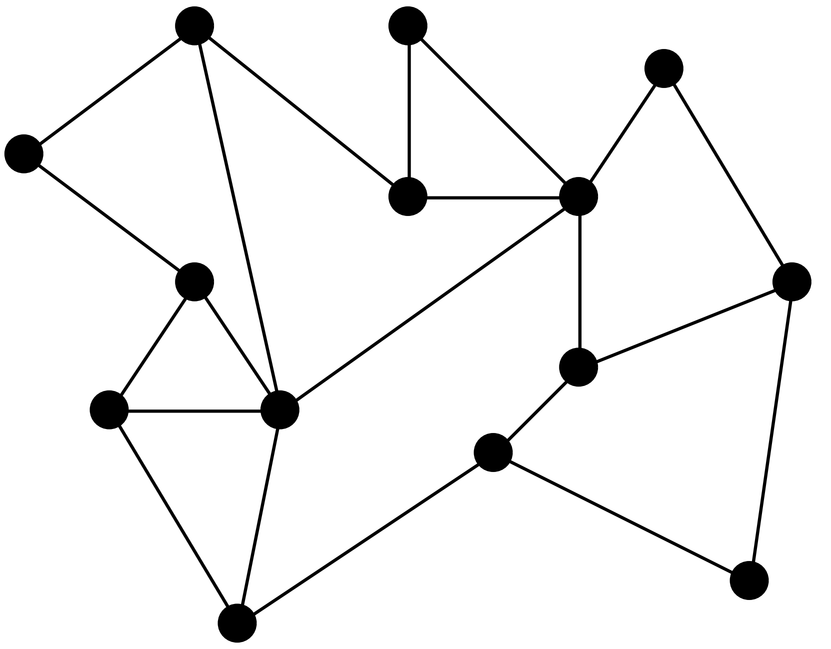



h

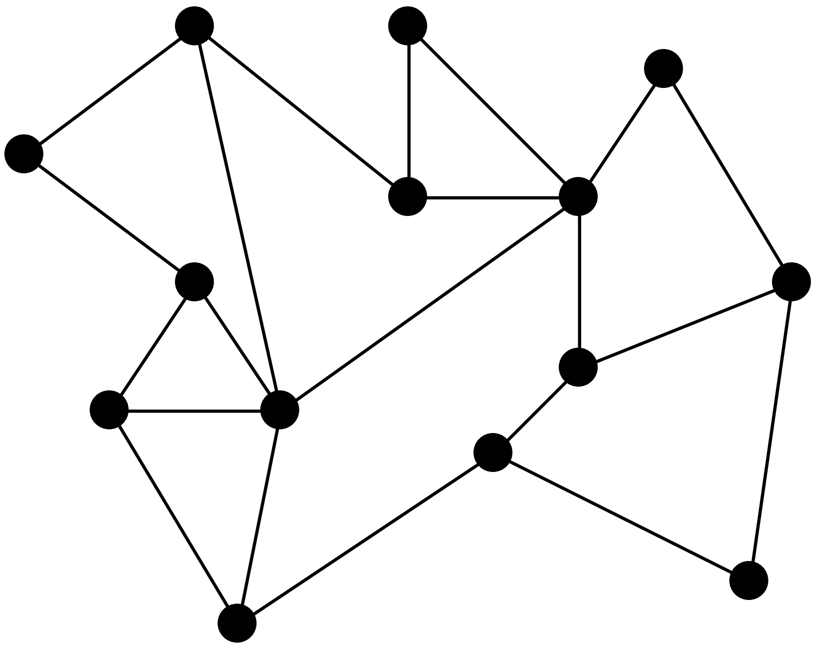

h

$d = 1$

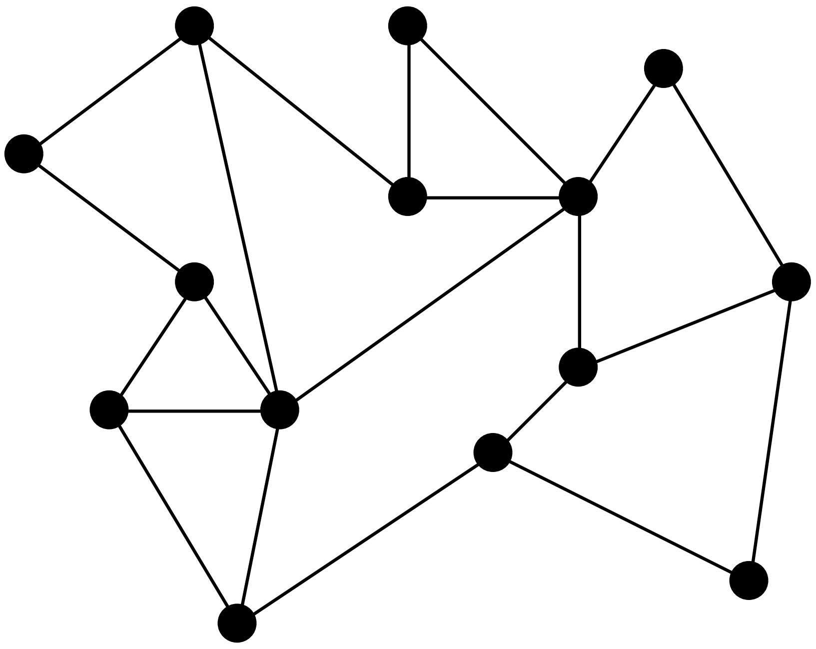


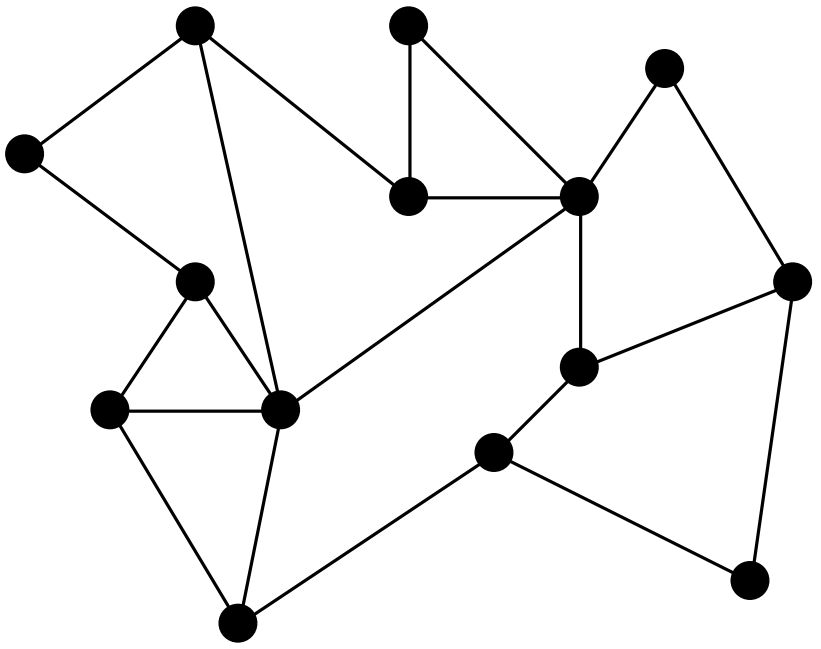


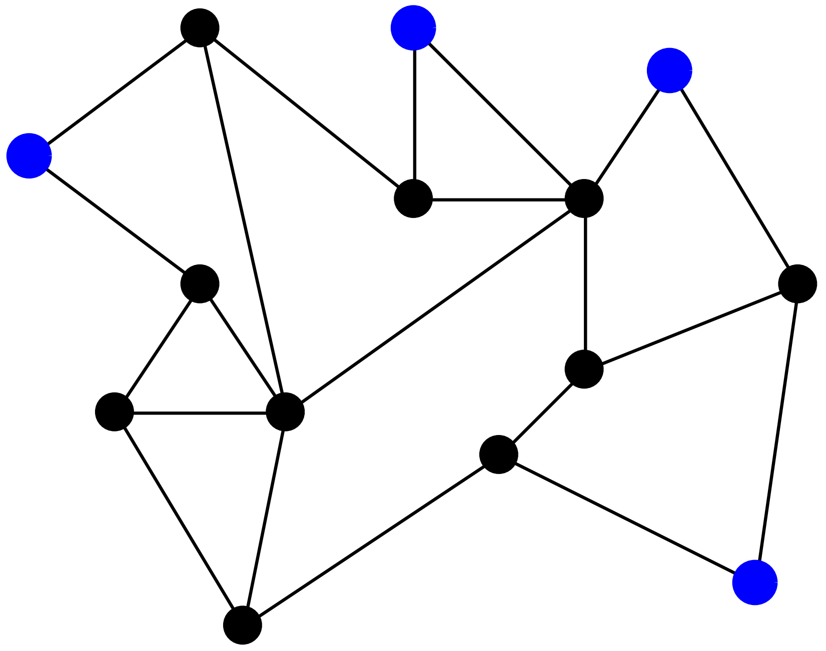
Planar graphs have degeneracy at most 5

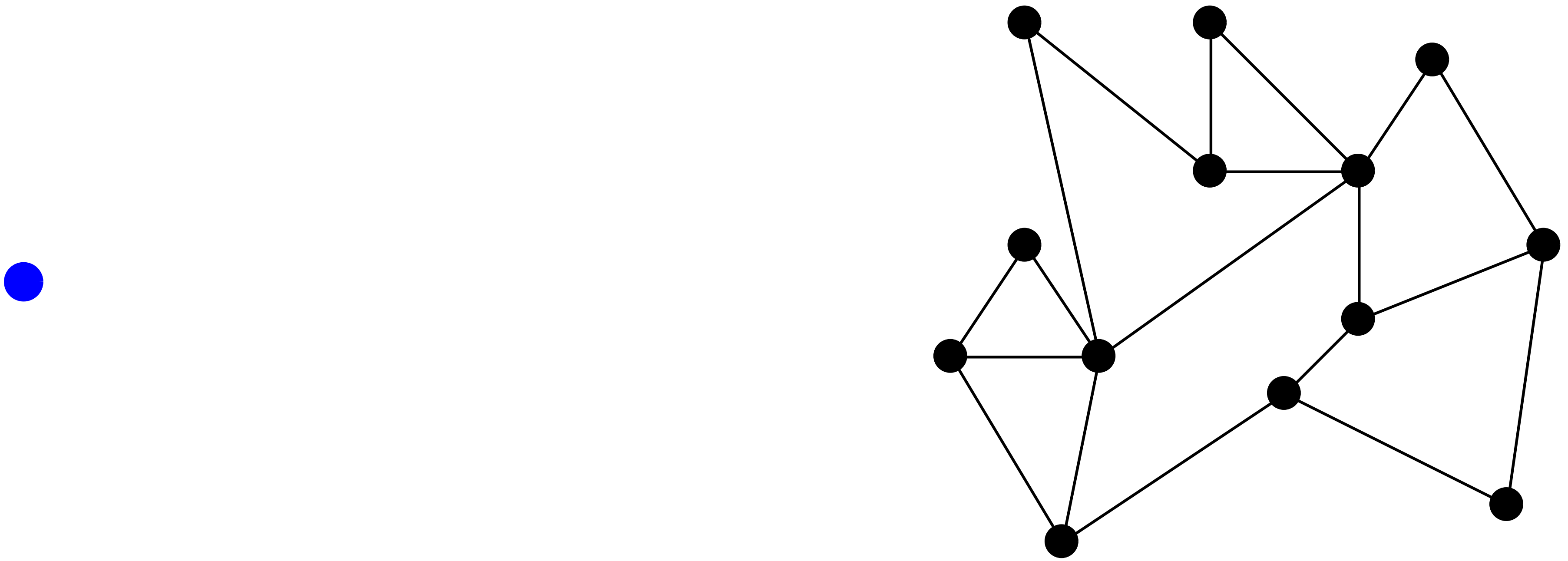


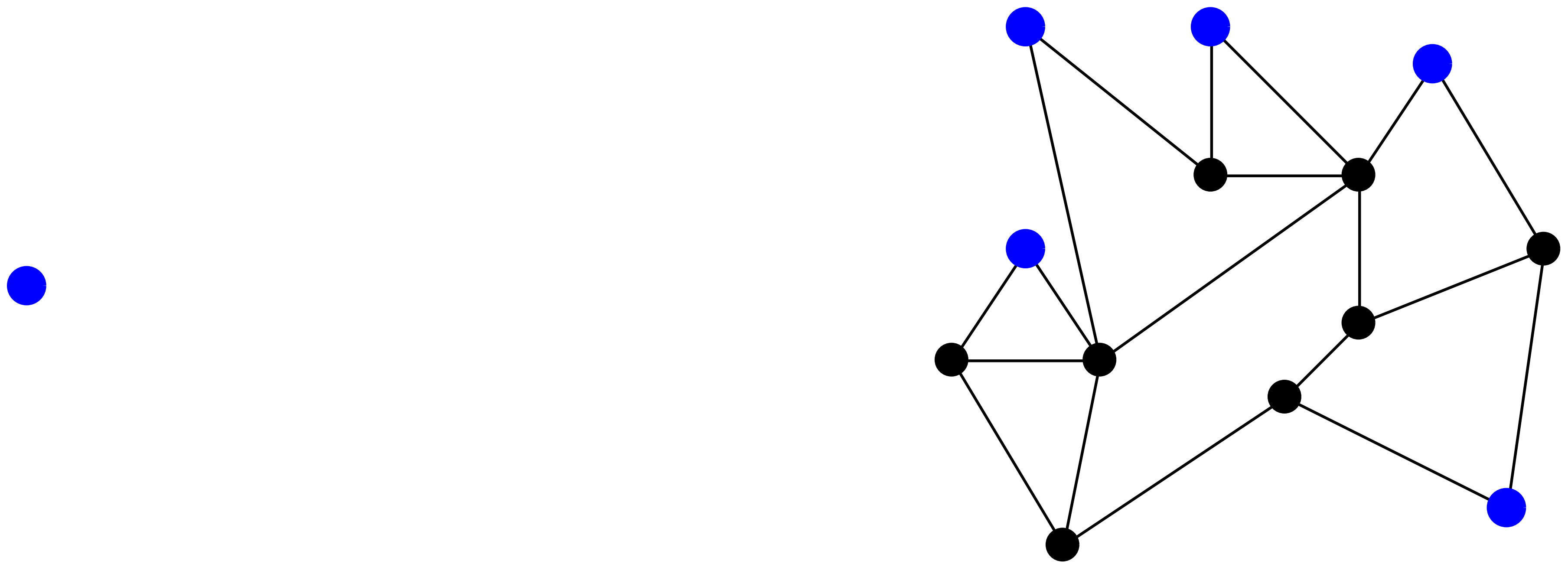
Degeneracy is easy to compute

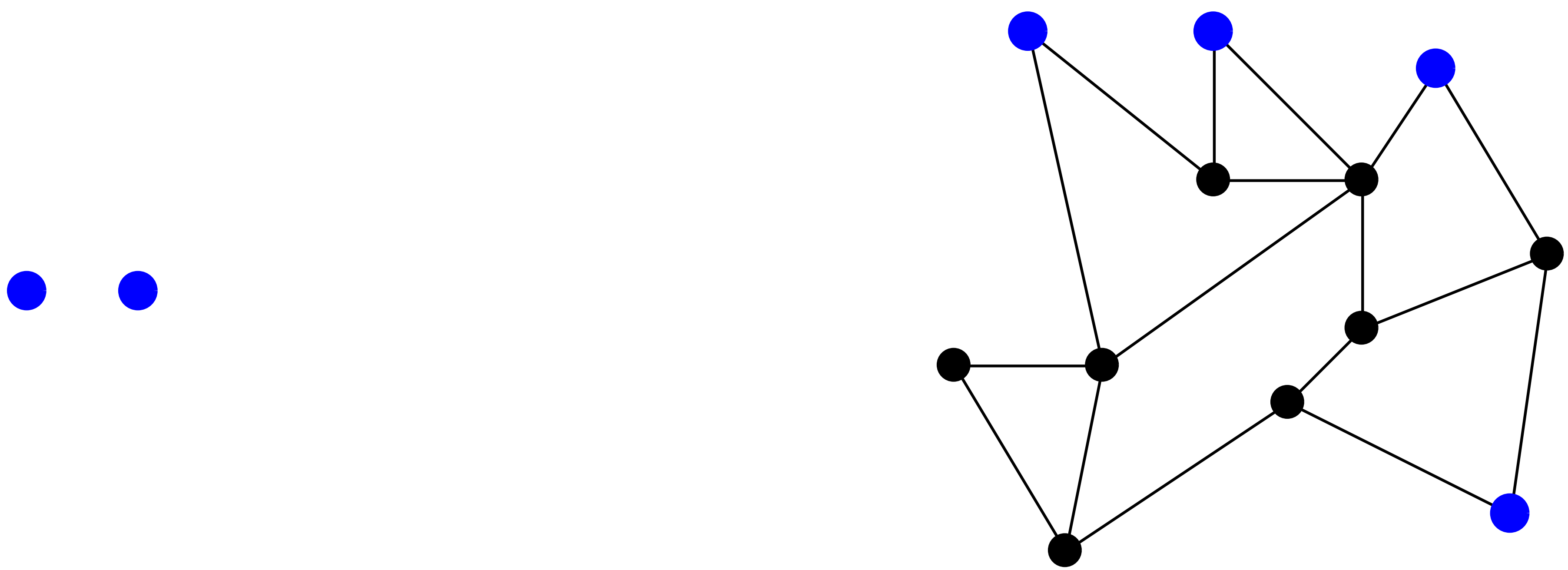


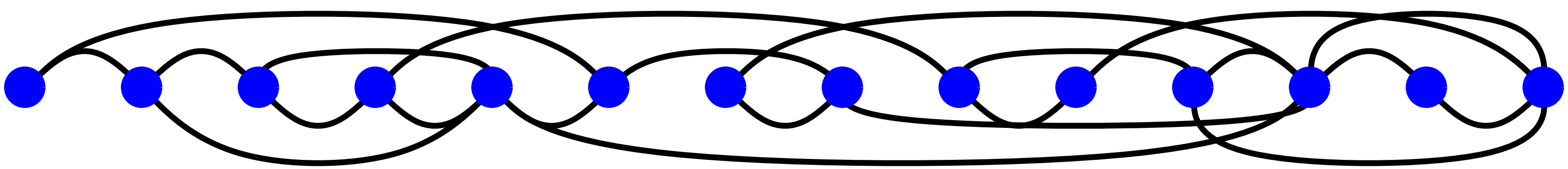












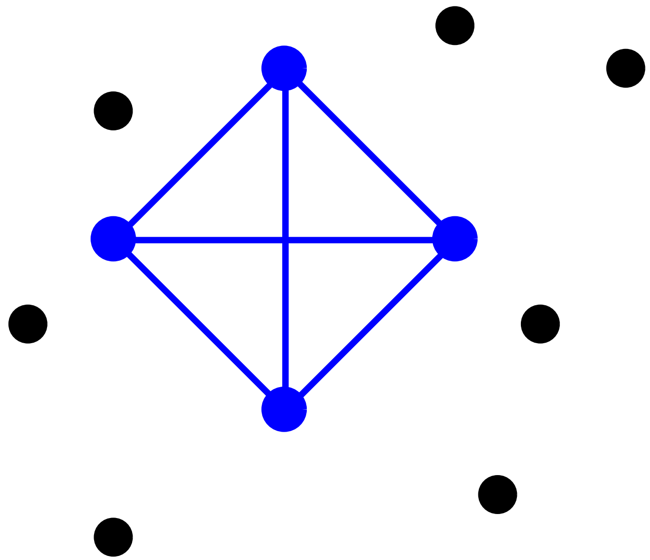
d-degenerate graphs...

d-degenerate graphs...

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

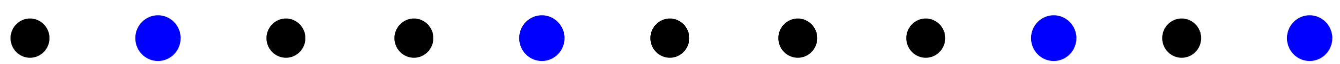
d-degenerate graphs...

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



d-degenerate graphs...

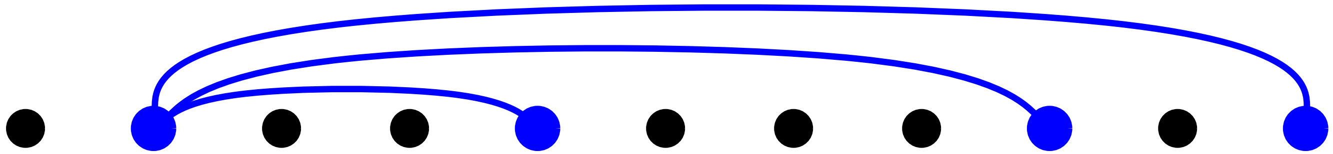
cannot contain cliques with more than $d+1$ vertices



d-degenerate graphs...

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

$> d$ later neighbors.



d-degenerate graphs...

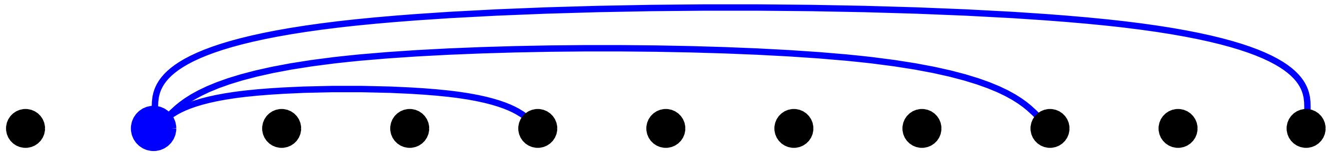
d-degenerate graphs...

have fewer than dn edges.

d-degenerate graphs...

have fewer than dn edges.

$\leq d$ later neighbors.



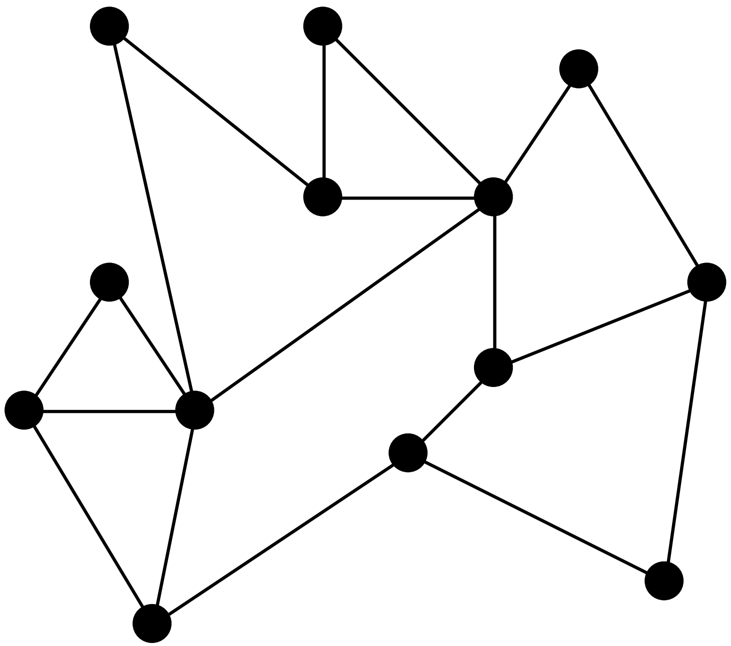
A few more facts about degeneracy...

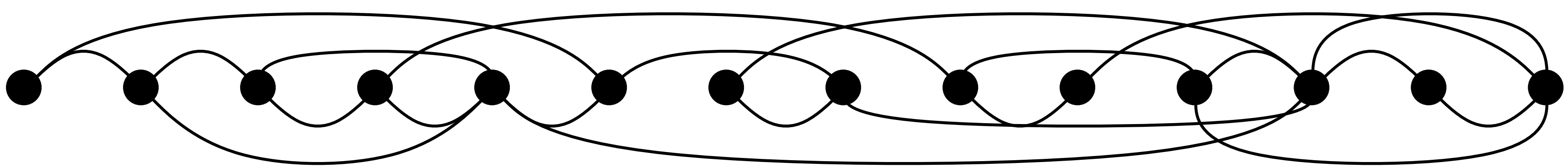
Degeneracy is within a constant factor of other popular sparsity measures.

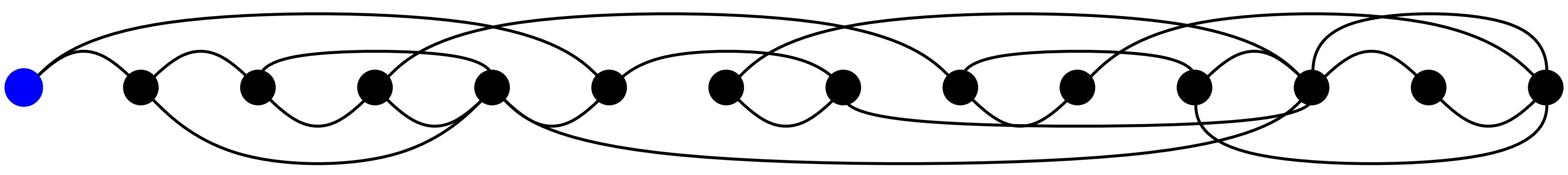
A few more facts about degeneracy...

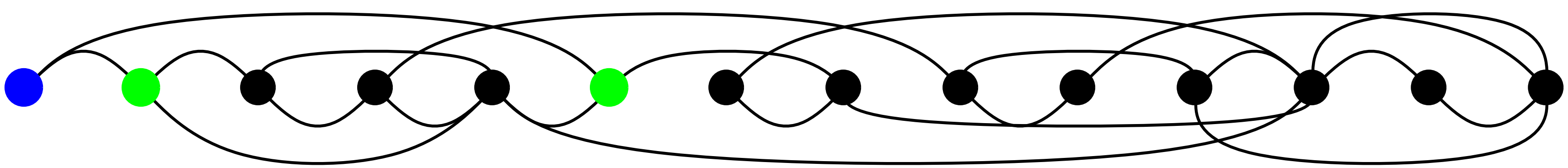
Degeneracy is within a constant factor of other popular sparsity measures.

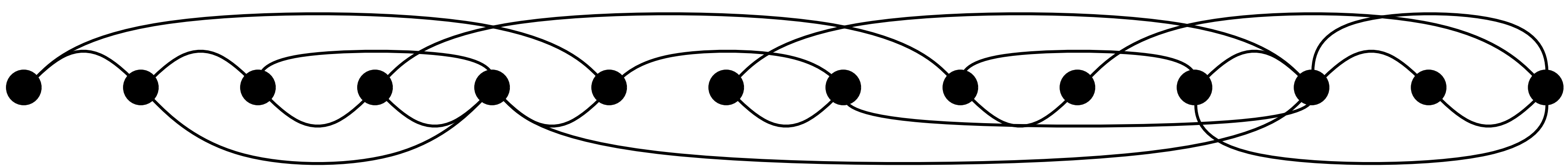
Graphs generated by the preferential attachment mechanism of Barabási and Albert have low degeneracy.

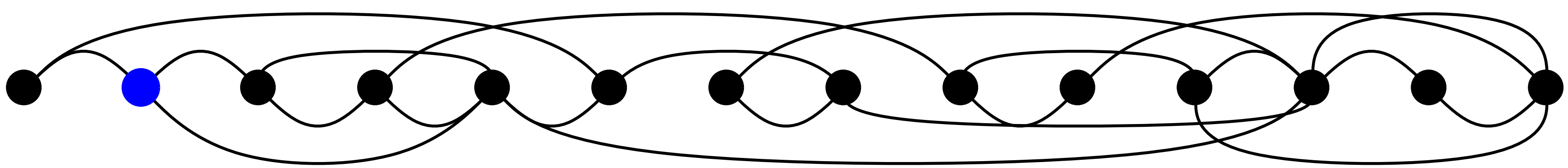


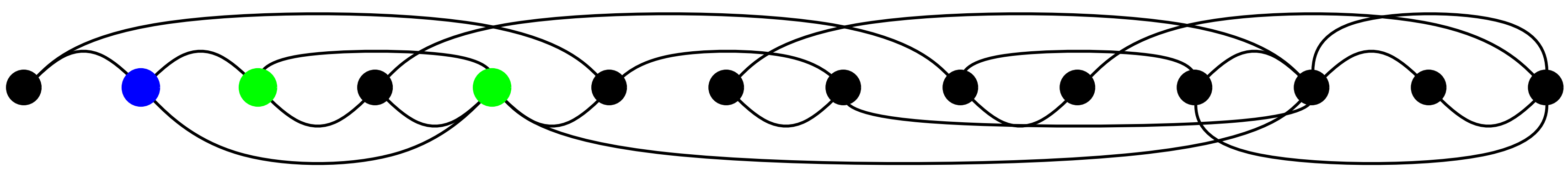


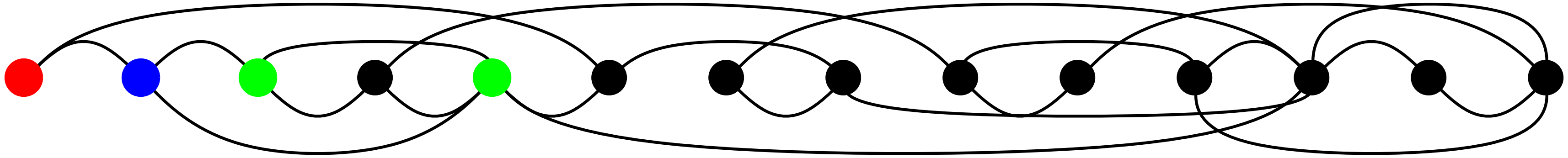












proc BronKerboschDegeneracy(V, E)

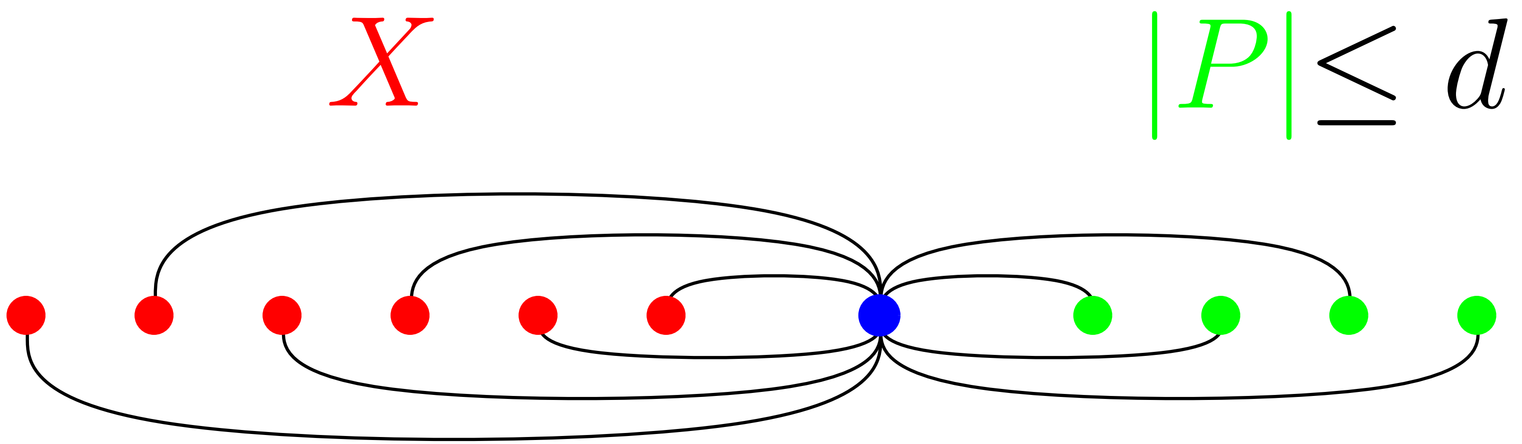
for each vertex v_i in a degeneracy ordering v_0, v_1, v_2, ... of (V, E) **do**

  $P \leftarrow \Gamma(v_i) \cap \{v_{i+1}, \ldots, v_{n-1}\}$

  $X \leftarrow \Gamma(v_i) \cap \{v_0, \ldots, v_{i-1}\}$

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

end for



X
$|P| \leq d$

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

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