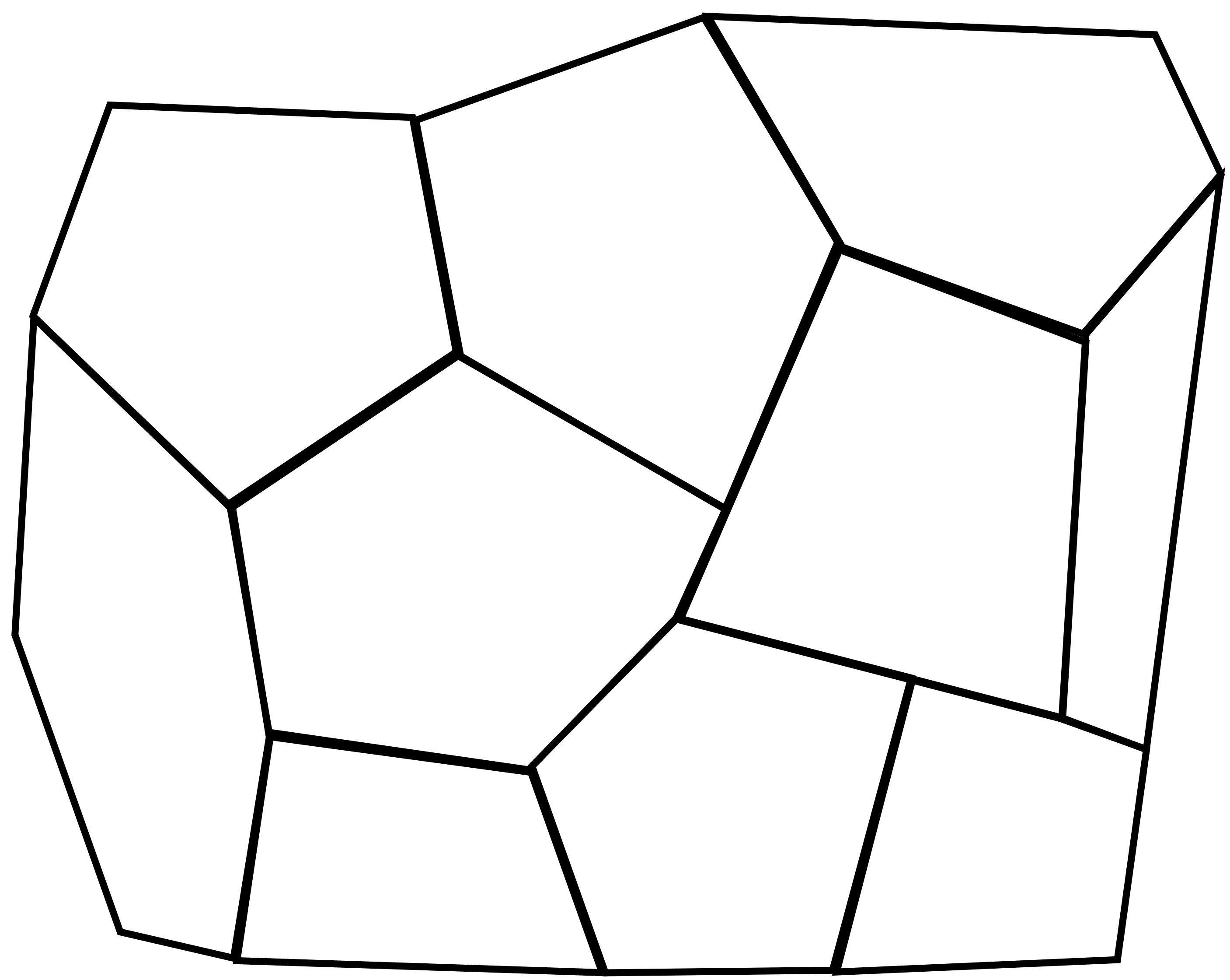
Graph Coloring

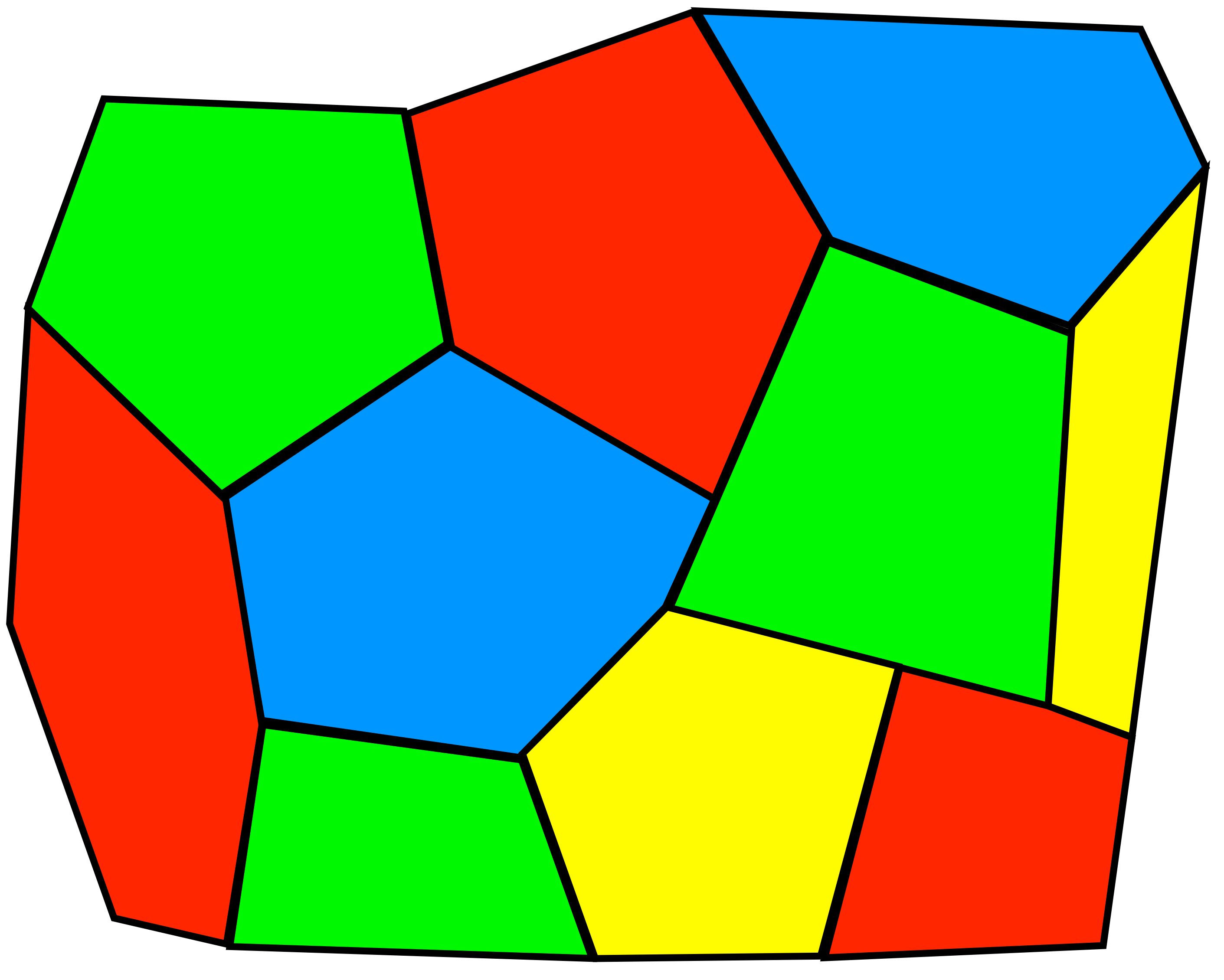
Andreas Björklund

Coloring a Map

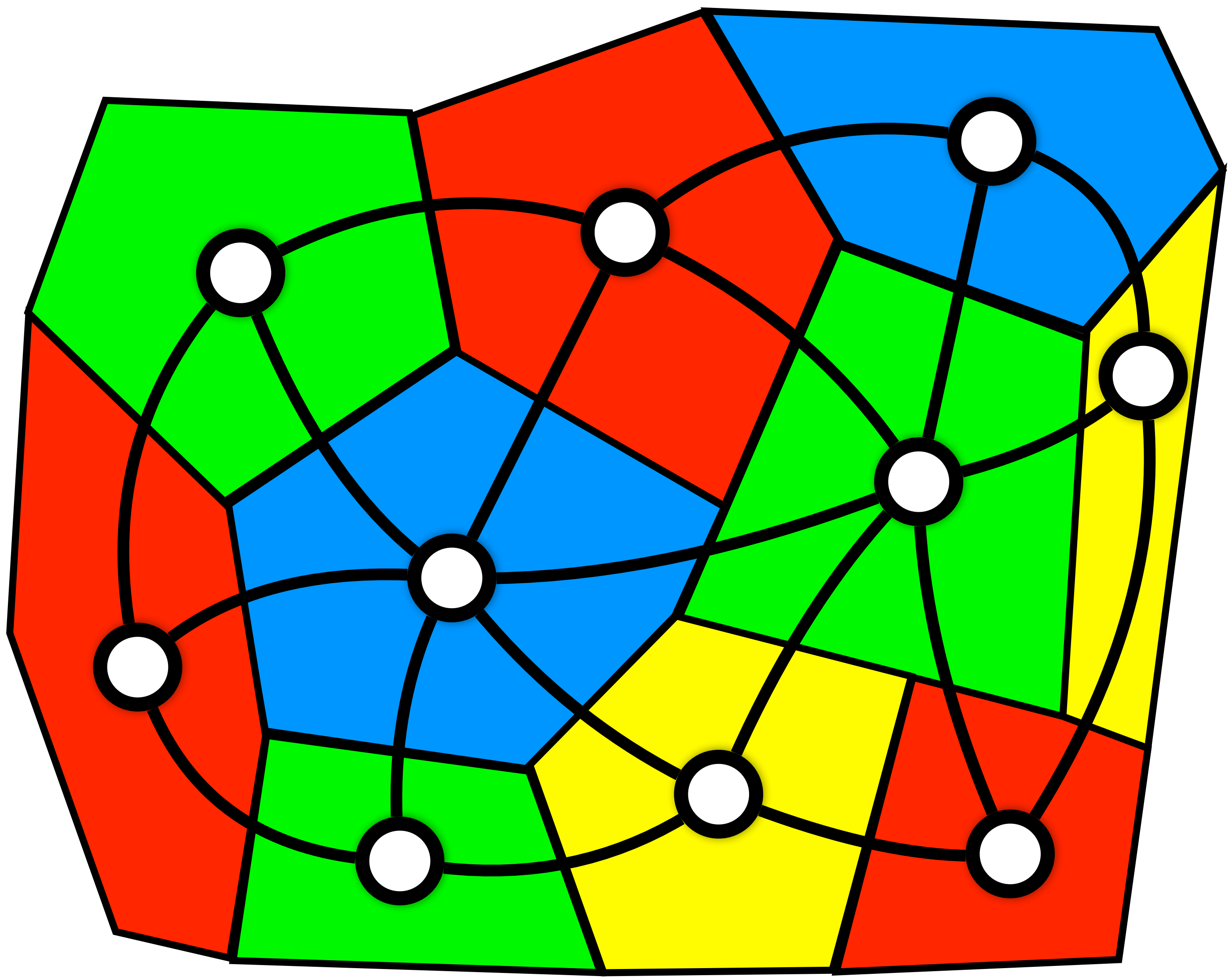
[Francis Guthrie 1852]



Coloring a Map

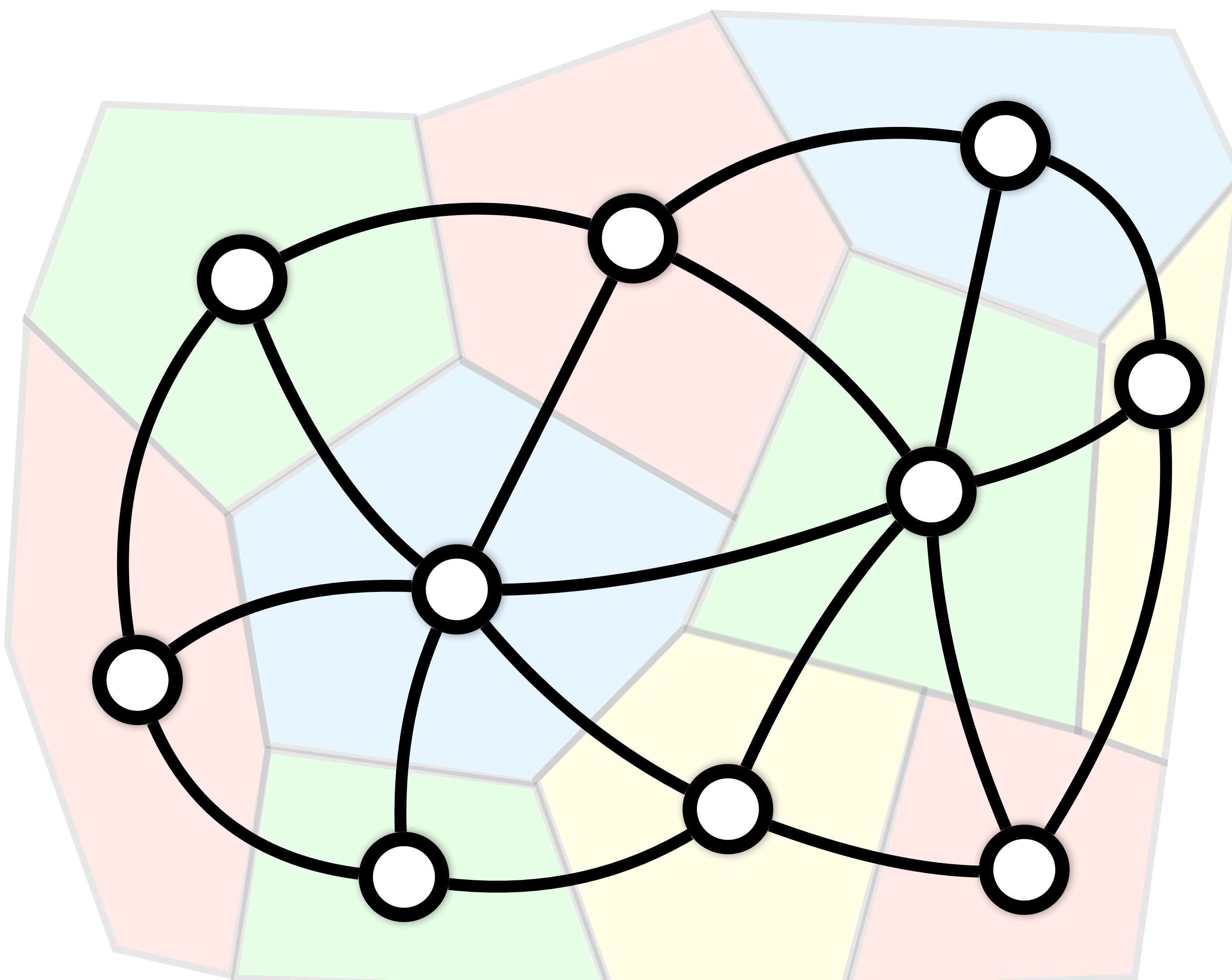


Graph Representation



Compact Description

G=(V,E)

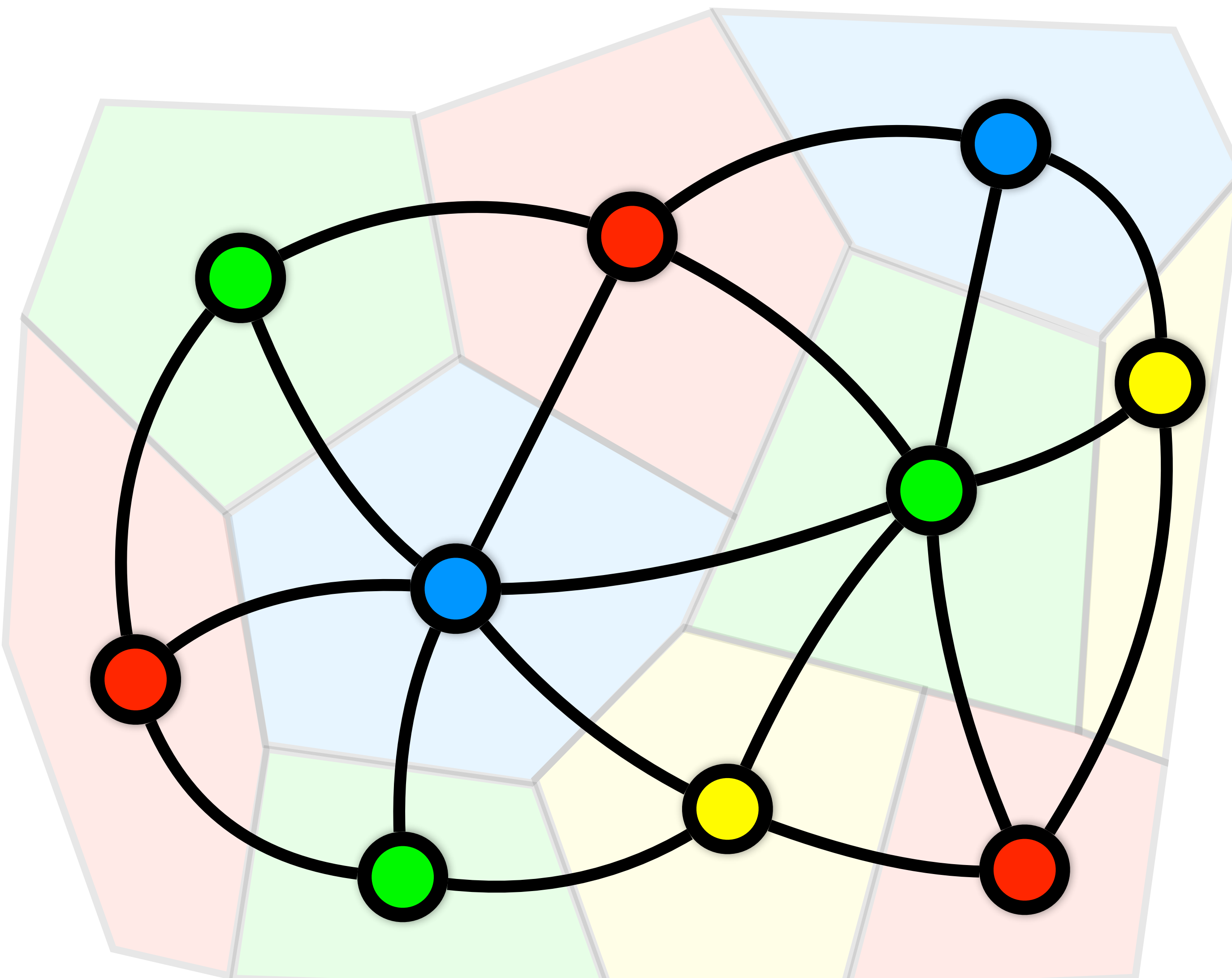


V=Set of vertices

E=Set of edges,
pairs of vertices

Vertex Coloring

G=(V,E)



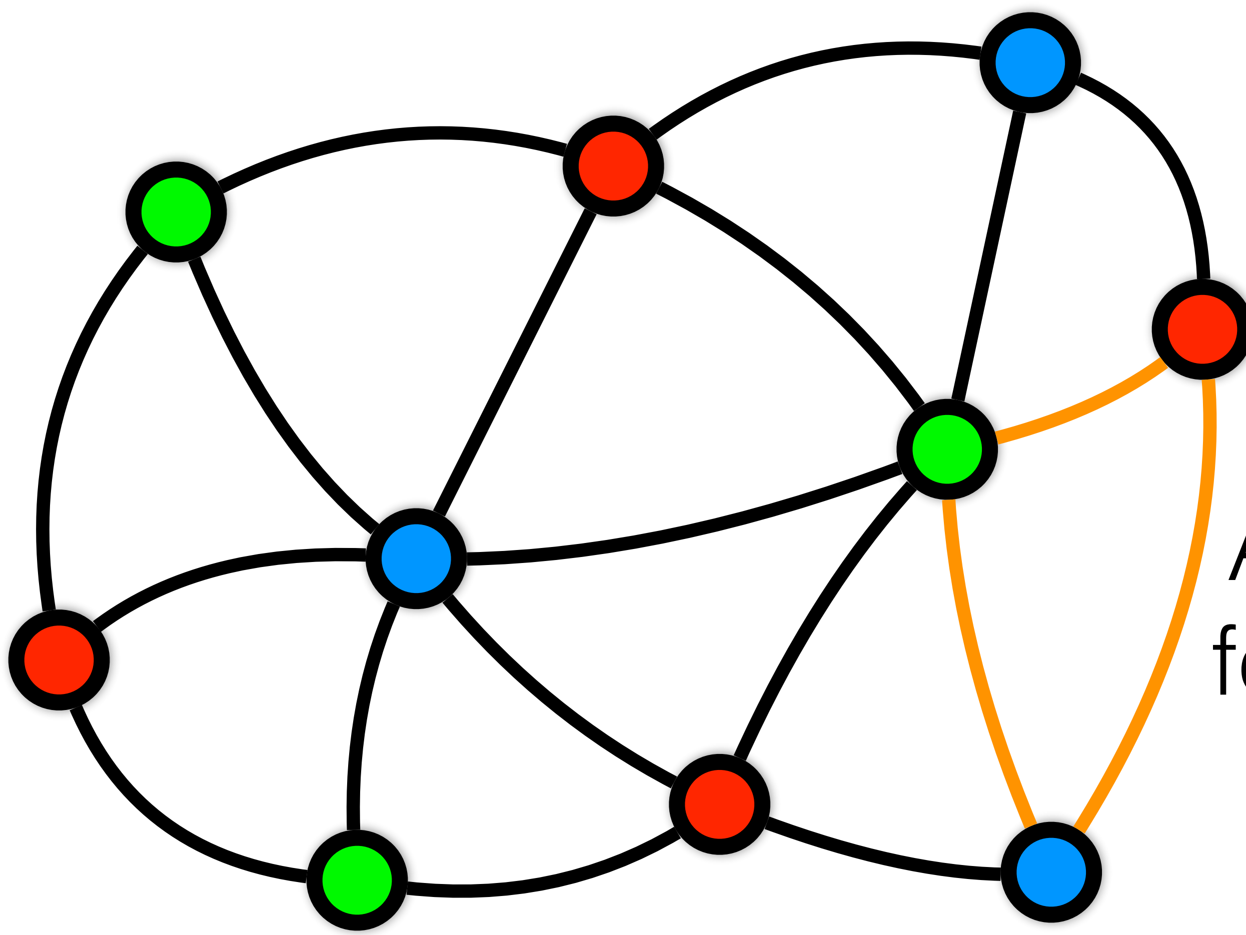
V=Set of vertices

E=Set of edges,
pairs of vertices

Chromatic Number

- Given a graph $G=(V,E)$, the chromatic number $\chi(G)$ is the smallest number of colors needed to color the vertices V so that for every edge e in E, the endpoints have different colors.

The Chromatic Number is Three



An odd cycle forces at least three colors

Planar Graphs

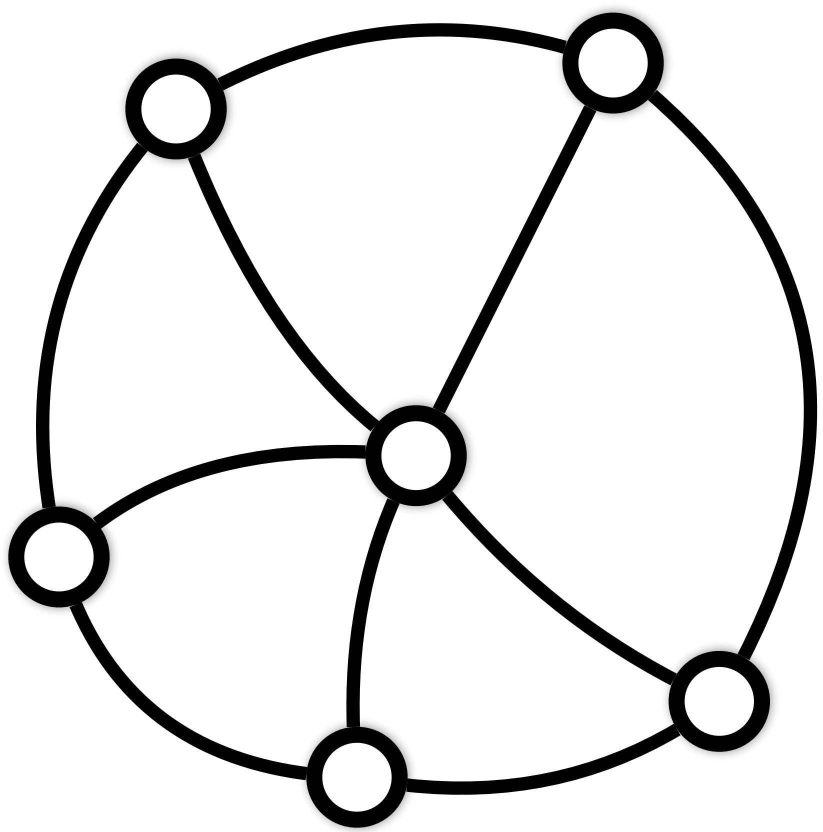
- A graph is planar if it can be drawn in the plane without any edges crossing each other.

- [Appel&Haken 1976] The chromatic number of a planar graph is at most four.

  - Proof through computer assisted case analysis, it gives a (complicated) polynomial time algorithm.

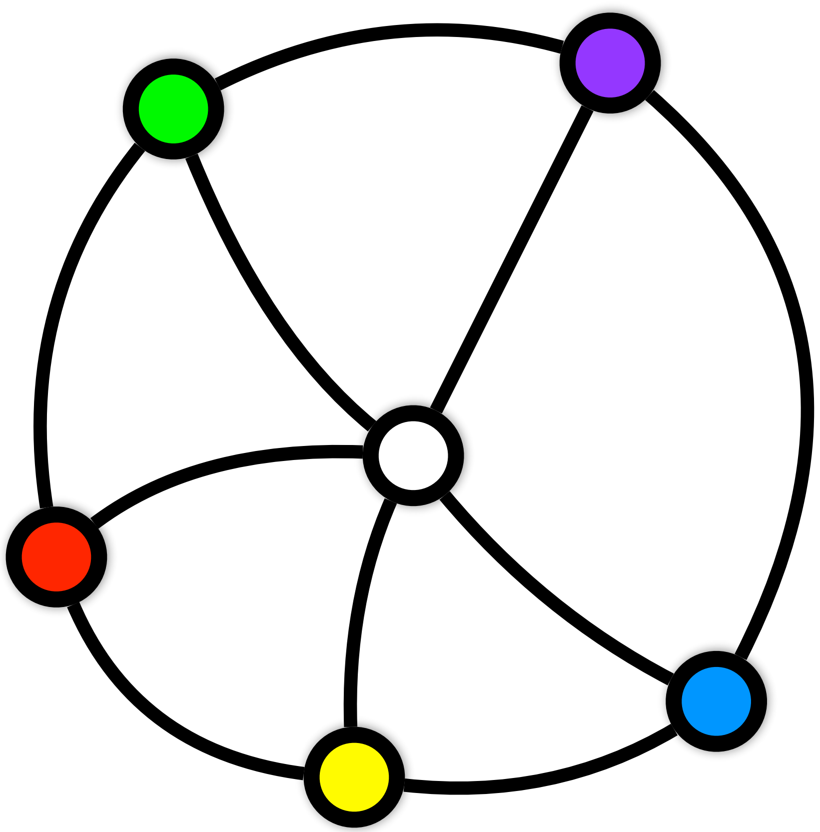
Simple Argument $\chi(G)<6$

[Heawood 1890]



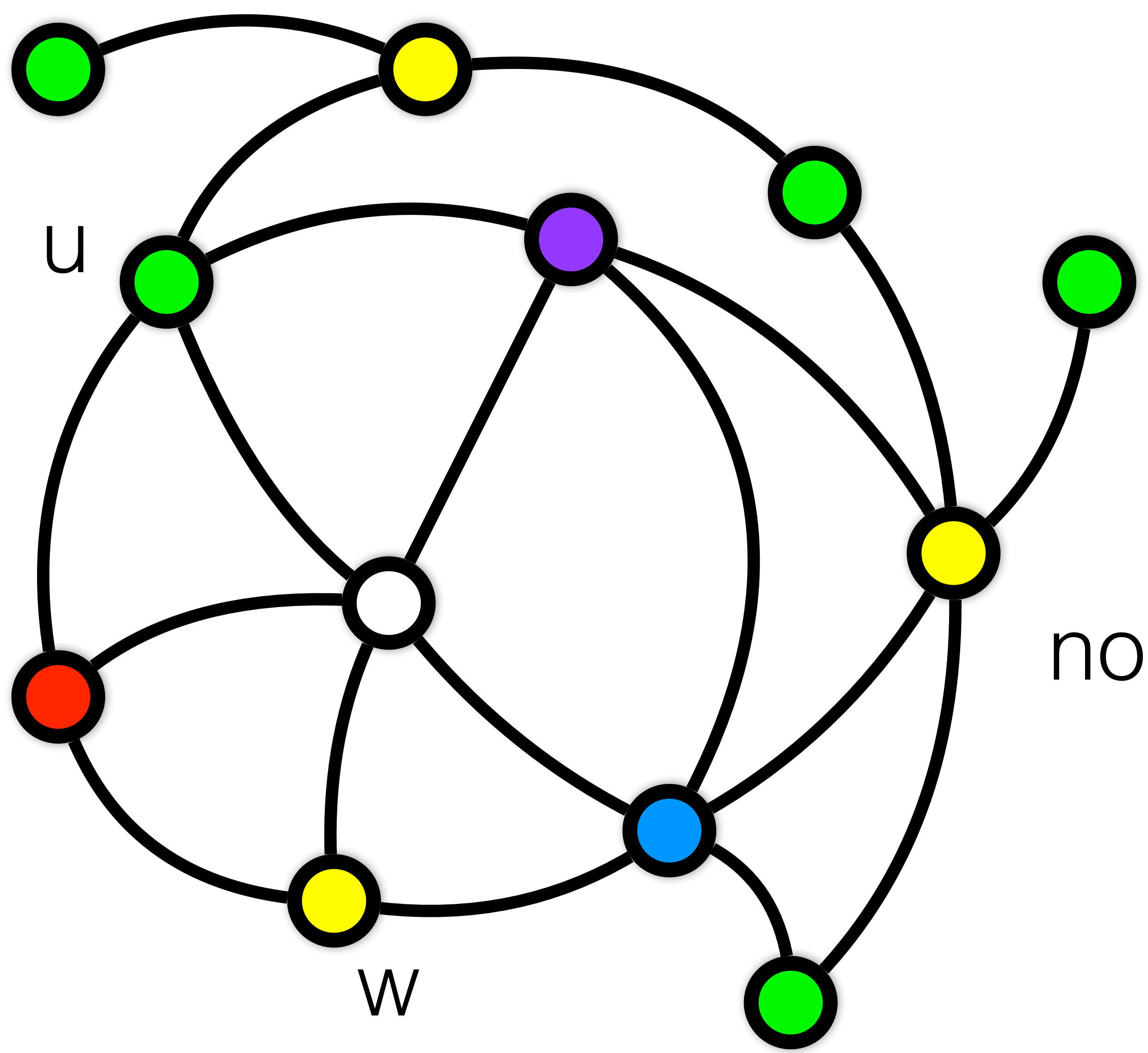
There must be a vertex of degree at most 5.

Simple Argument $\chi(G)<6$



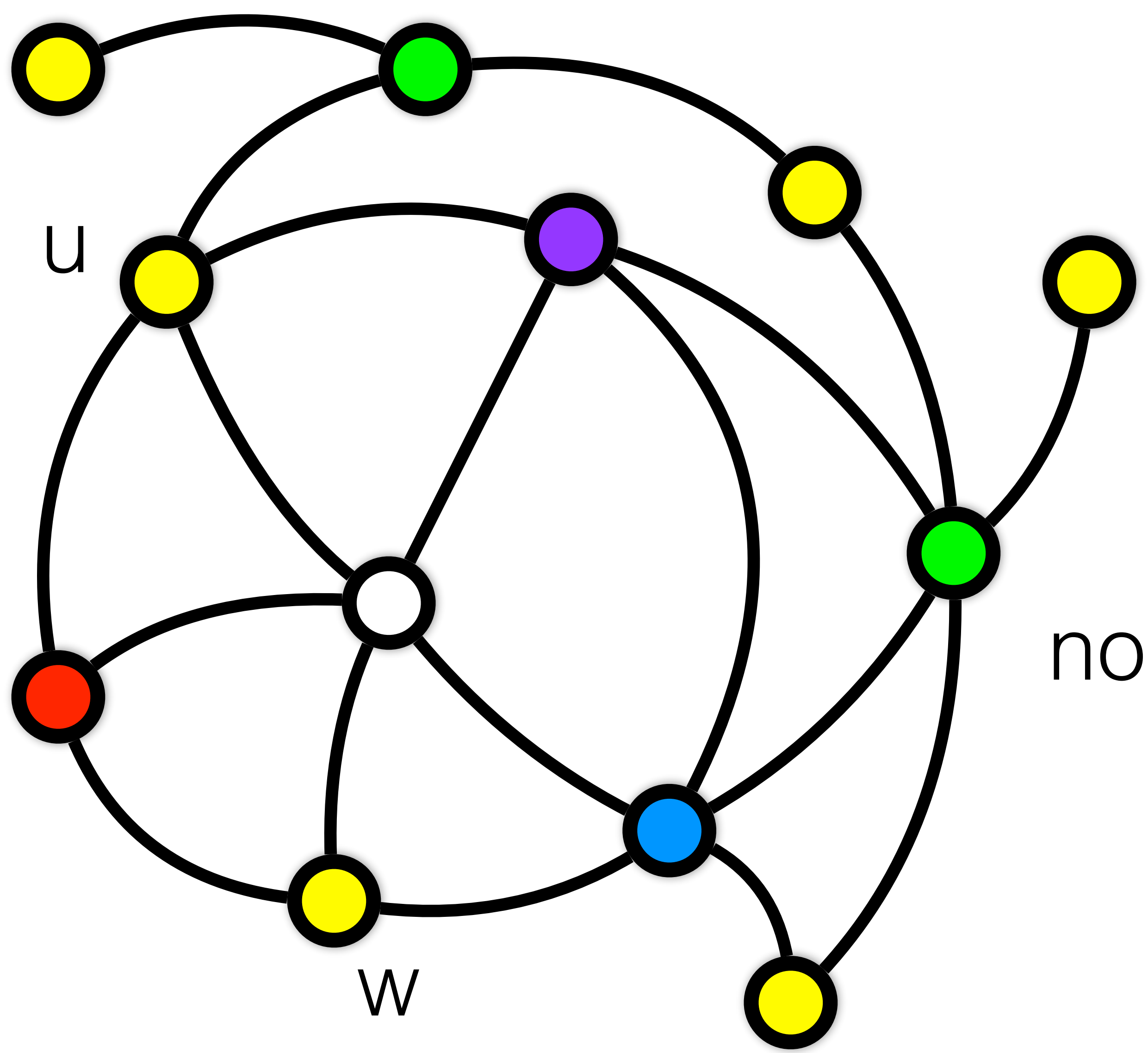
Assume its neighbors are colored in 5 colors.

Simple Argument $\chi(G)<6$



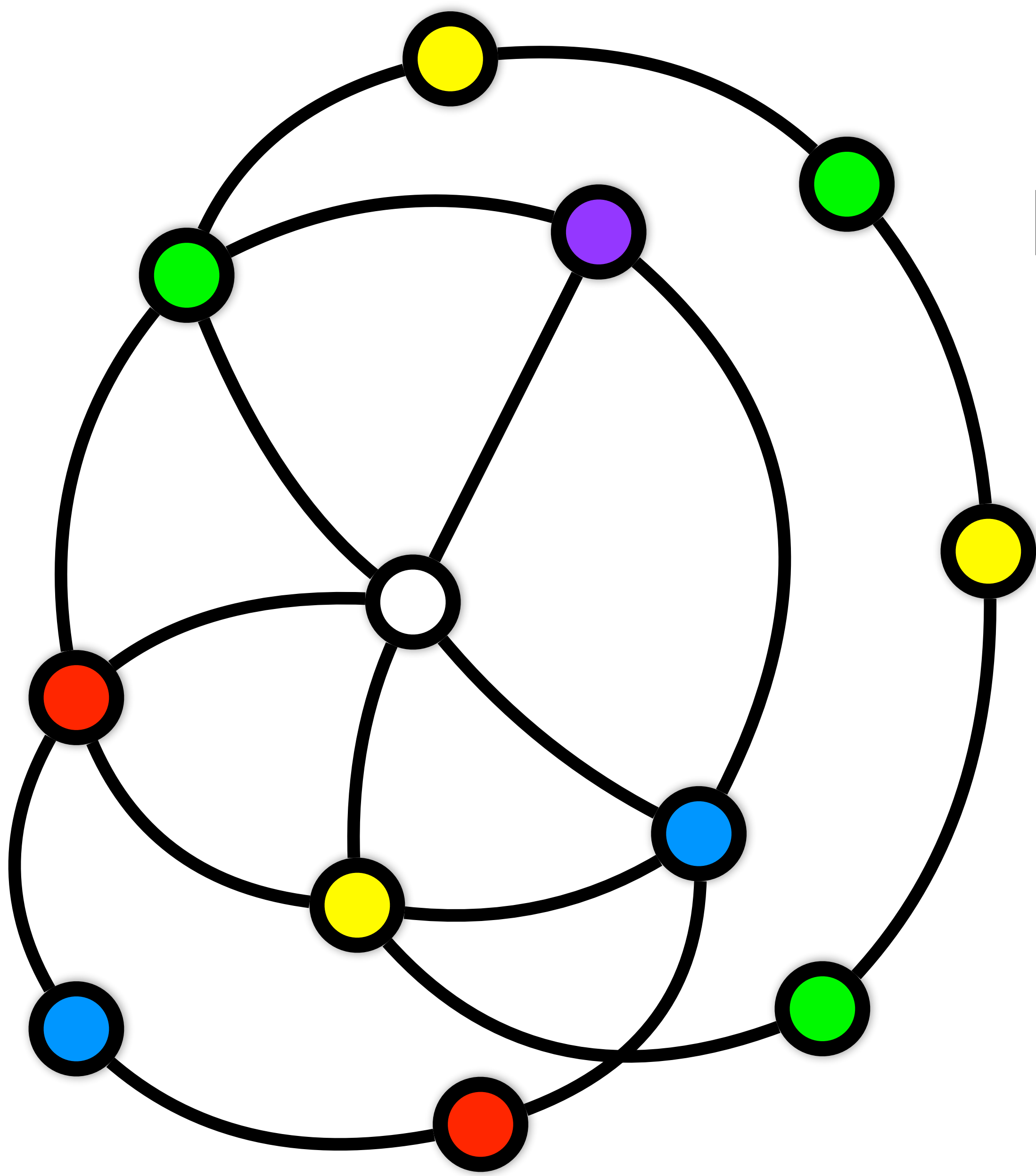
If one neighbor u is in alternating colored component that is not connected to neighbor w of the other color, then Recoloring is possible!

Simple Argument χ(G)<6



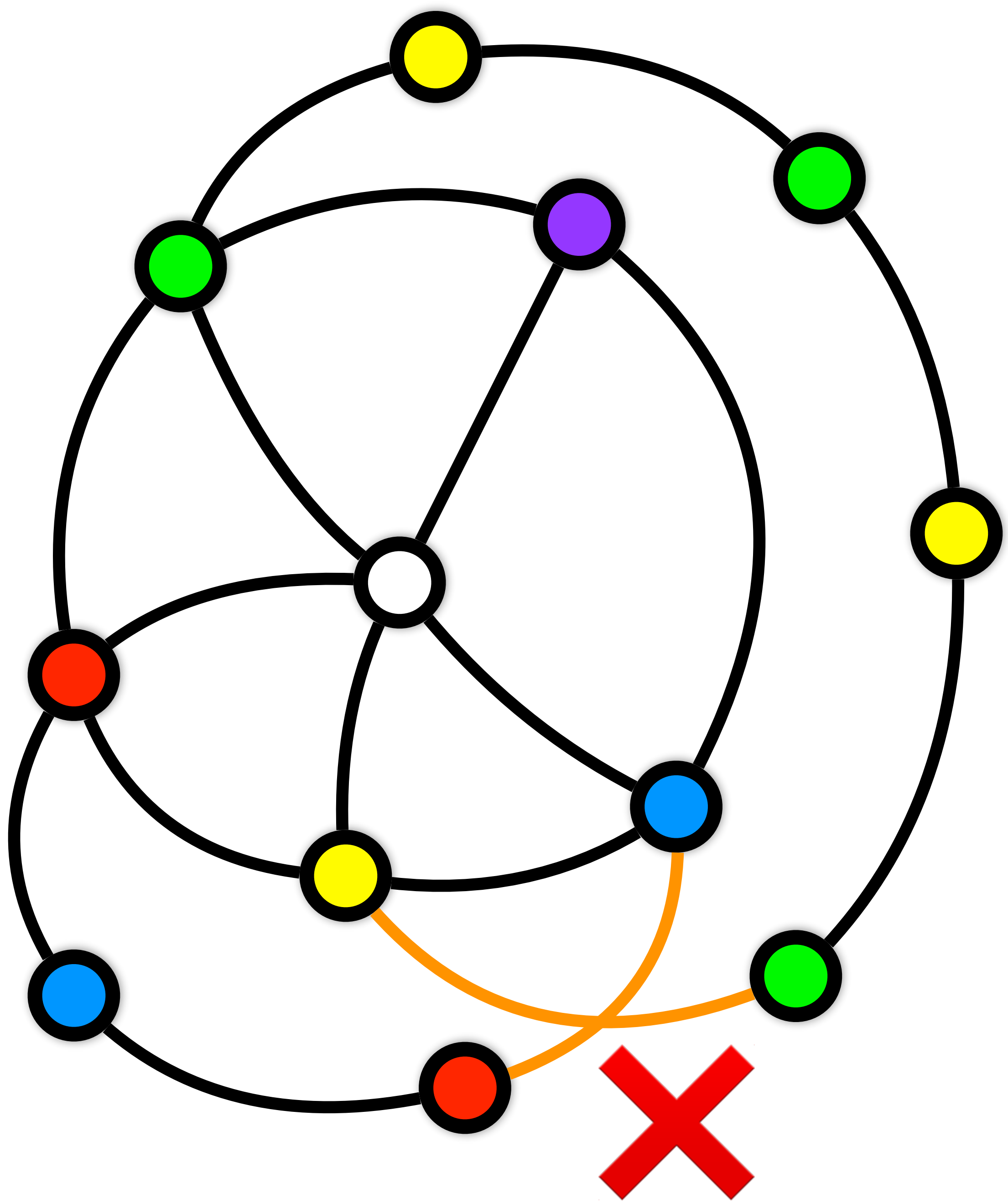
If one neighbor u is in alternating colored component that is not connected to neighbor w of the other color, then Recoloring is possible!

Simple Argument χ(G)<6



Must be alternatingly colored paths between neighbors.

Simple Argument $\chi(G)<6$



Paths must cross each other
=> contradiction!

Polynomial Time Algorithm for Five Coloring Planar Graphs

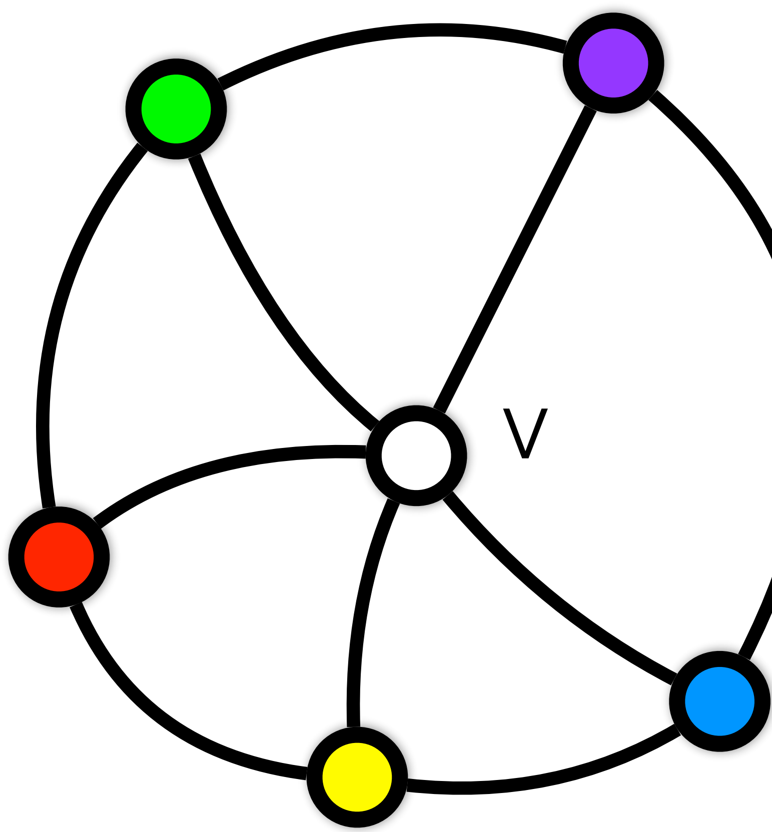
- [**Preprocessing**] Remove the vertex of smallest degree and put it on a stack. Repeat until all vertices are on the stack.

- [**Coloring**] As long as not all vertices have been colored, pop the stack and color the current vertex with the first available valid color given the already coloured vertices.

  - If no valid color is available, it must be possible to recolor a colored neighbor by Heawood's argument.

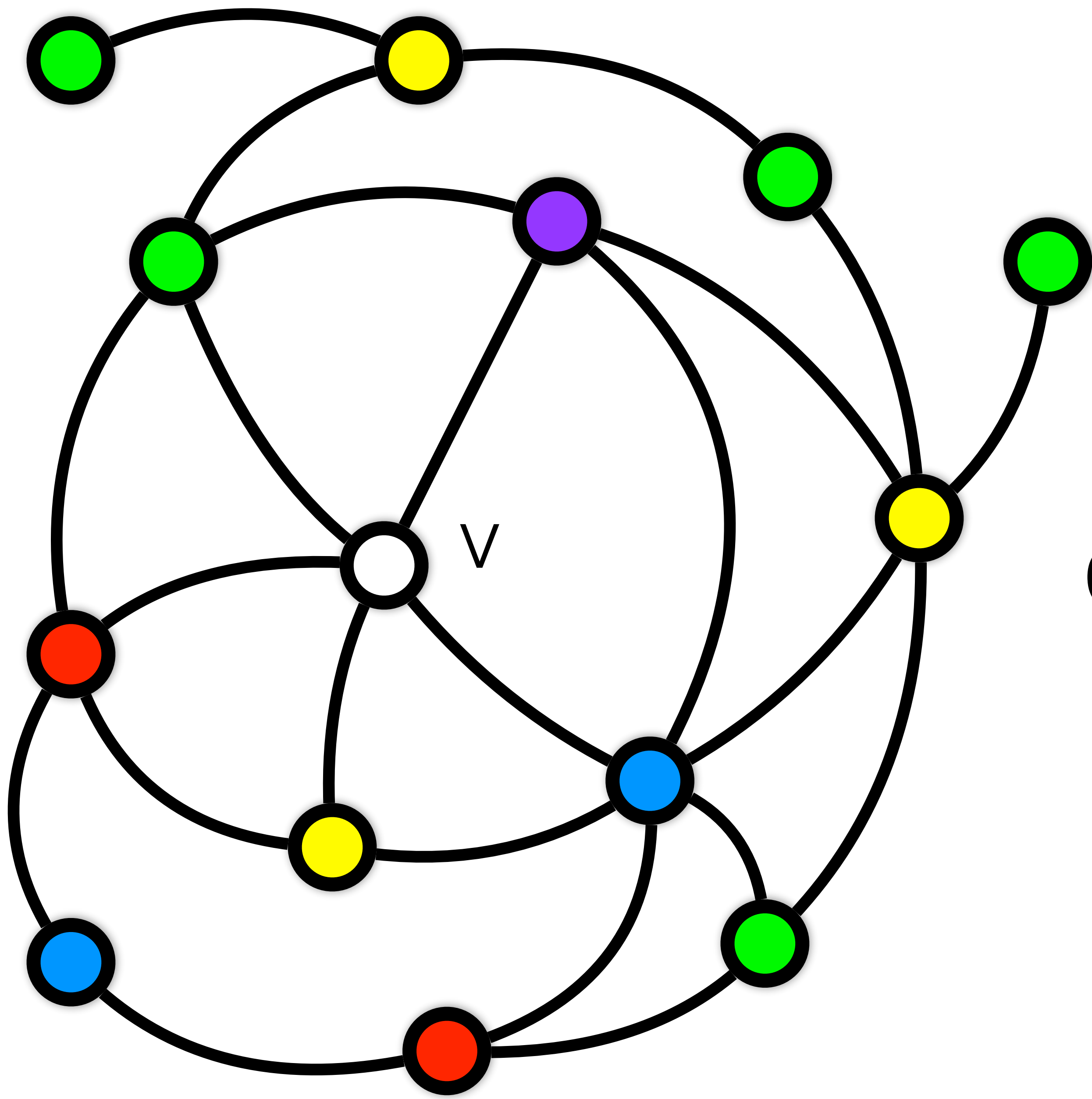
  - Find a vertex that can be recolored, do that and continue.

Recoloring a Neighbor



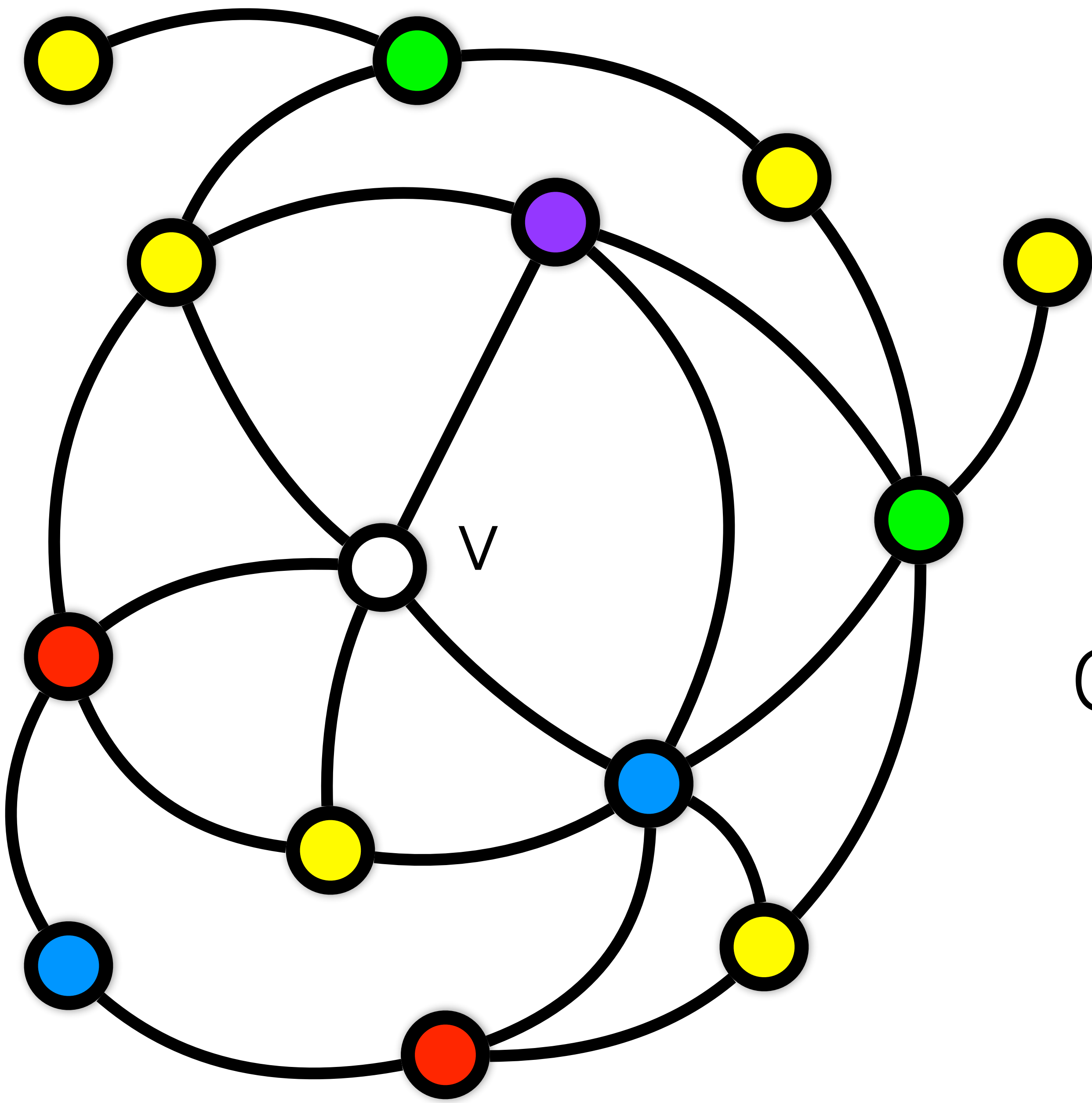
None of the five colors are valid for v.

Recoloring a Neighbor



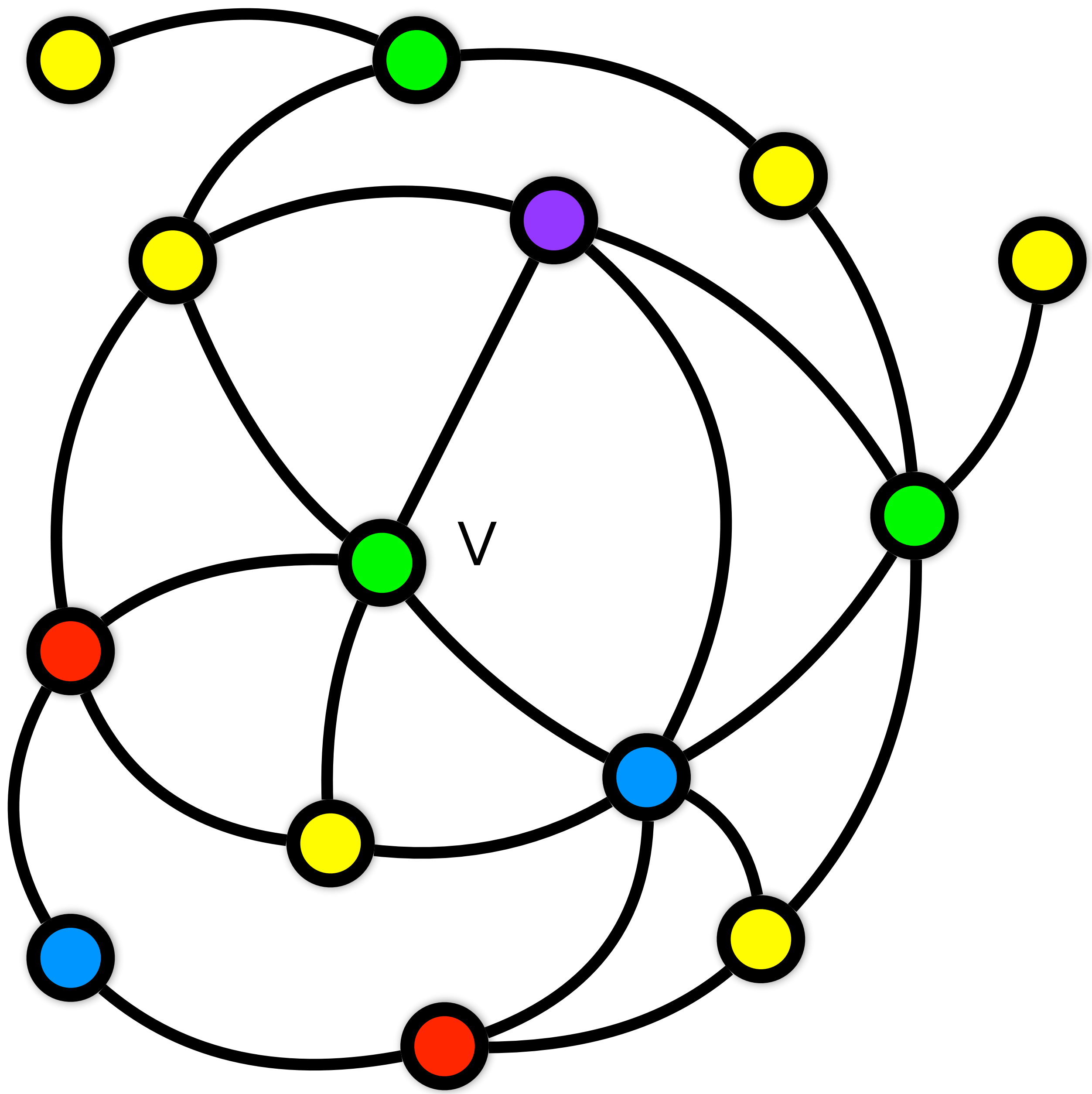
Green-Yellow component only incident with one neighbor of v.

Recoloring a Neighbor



Swap colors on
Green-Yellow component.

Recoloring a Neighbor



Green color is now valid for v.

Hardness of Coloring

- Computing the chromatic number in a general graph is NP-complete. In fact deciding if $\chi(G)=3$ in a planar graph is NP-complete.

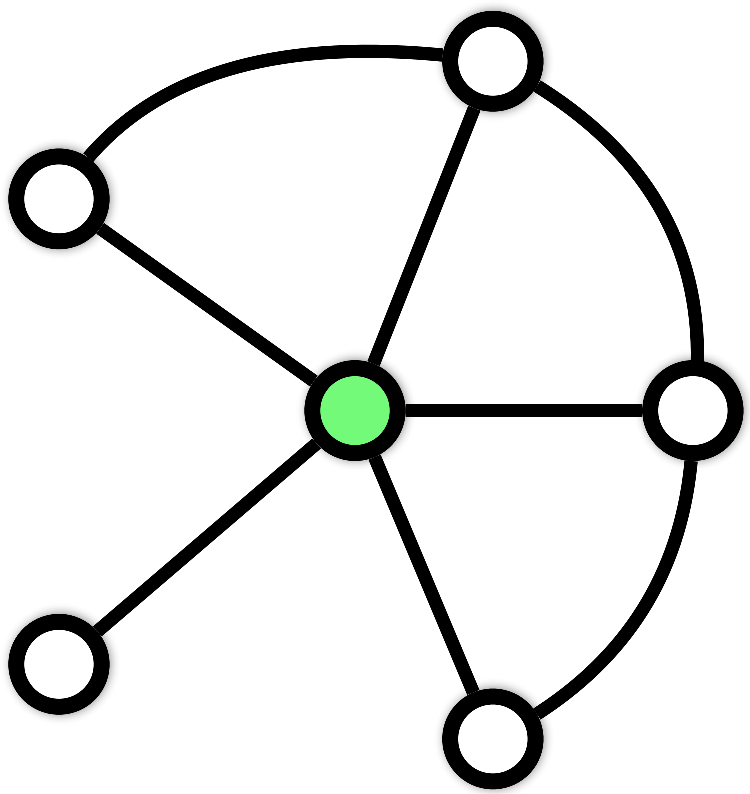
- The best polynomial time algorithms for coloring a 3-colorable graph uses more than $n^{1/6}$ colors!

Coloring a 3-colorable Graph with $O(n^{1/2})$ Colors in Polynomial Time

[Wigderson 1982]

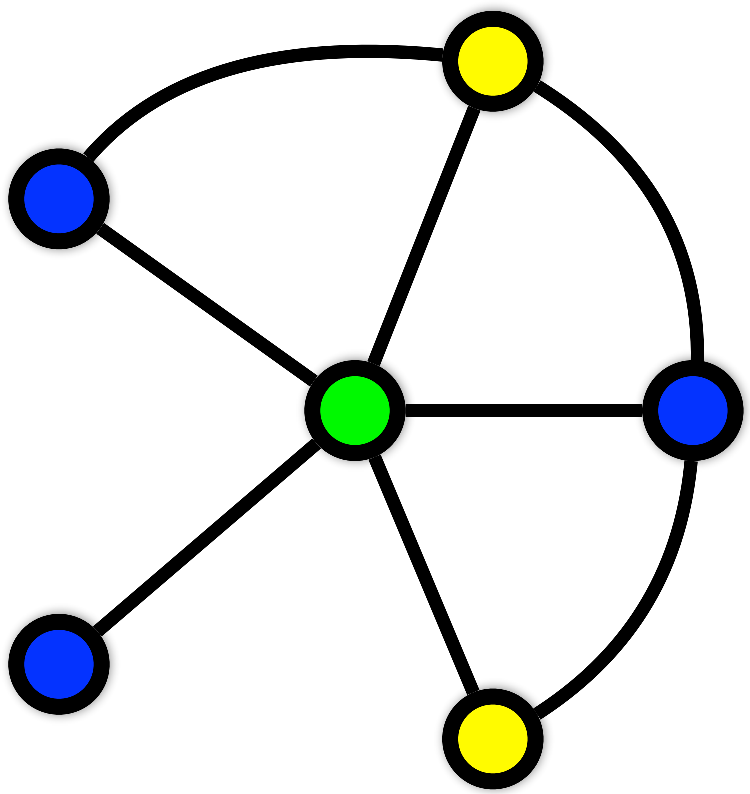
Note: Graphs need not be planar,
(they are just easier to visualise).

Coloring a 3-colorable Graph with $O(n^{1/2})$ Colors



If there exists vertex of degree more than $n^{1/2}$, color it and its neighbours with three fresh colors.

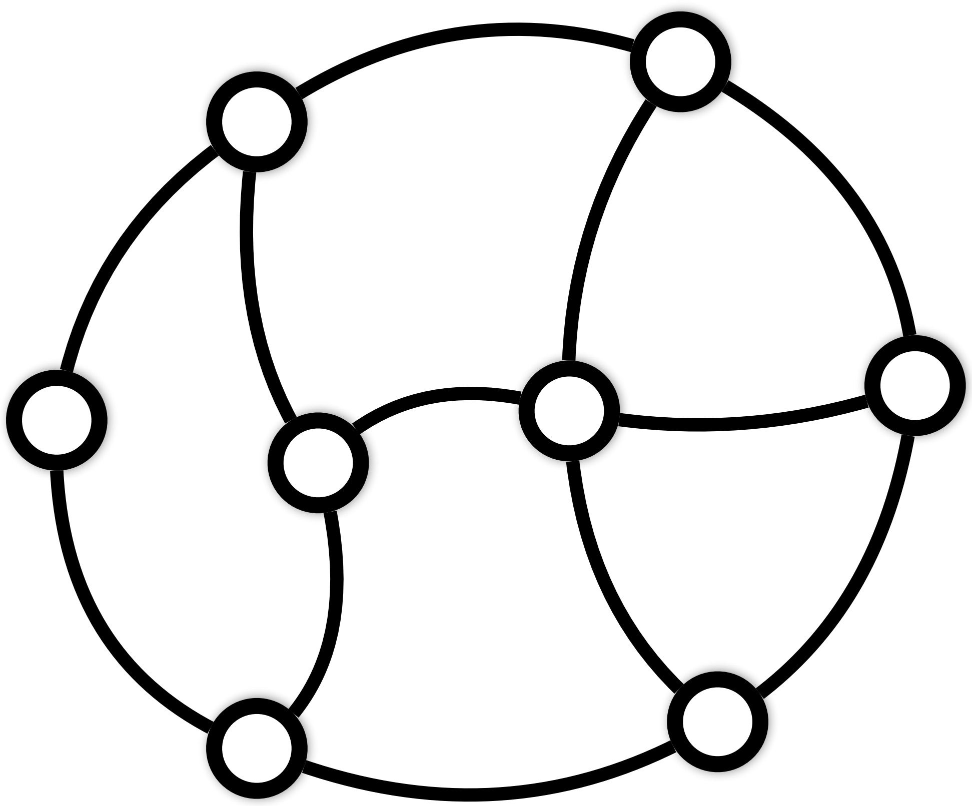
Coloring a 3-colorable Graph with $O(n^{1/2})$ Colors



If there exists vertex of degree more than $n^{1/2}$, color it and its neighbours with three fresh colors.

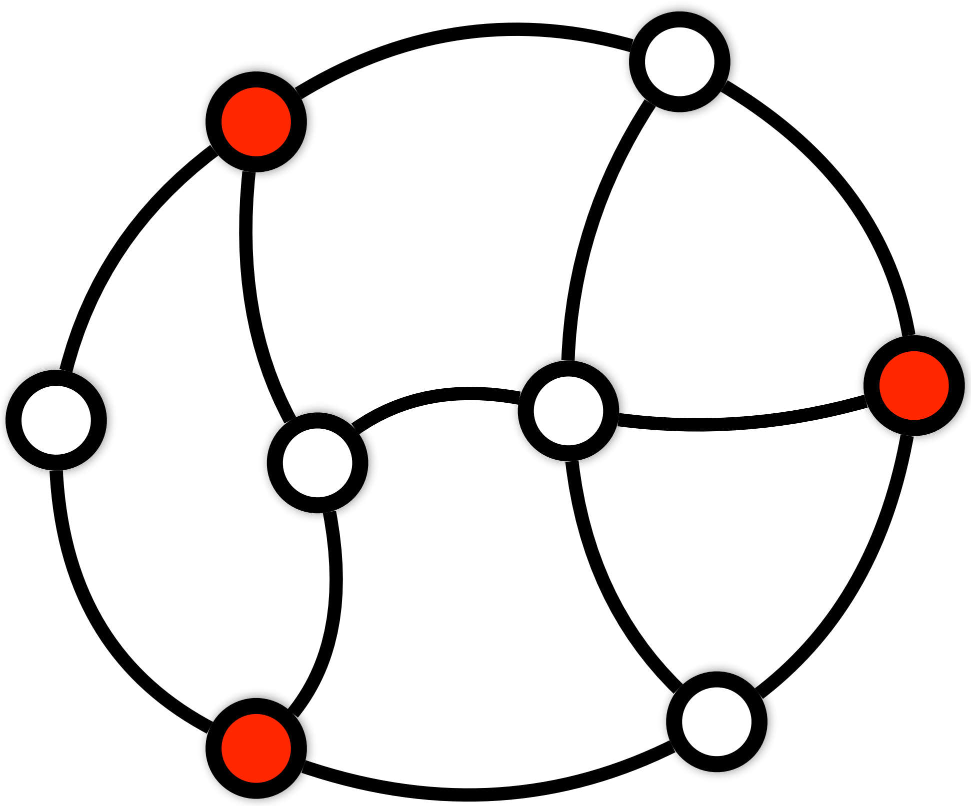
The neighbourhood must be bipartite, and hence it is easy to two-color.

Coloring a 3-colorable Graph with $O(n^{1/2})$ Colors



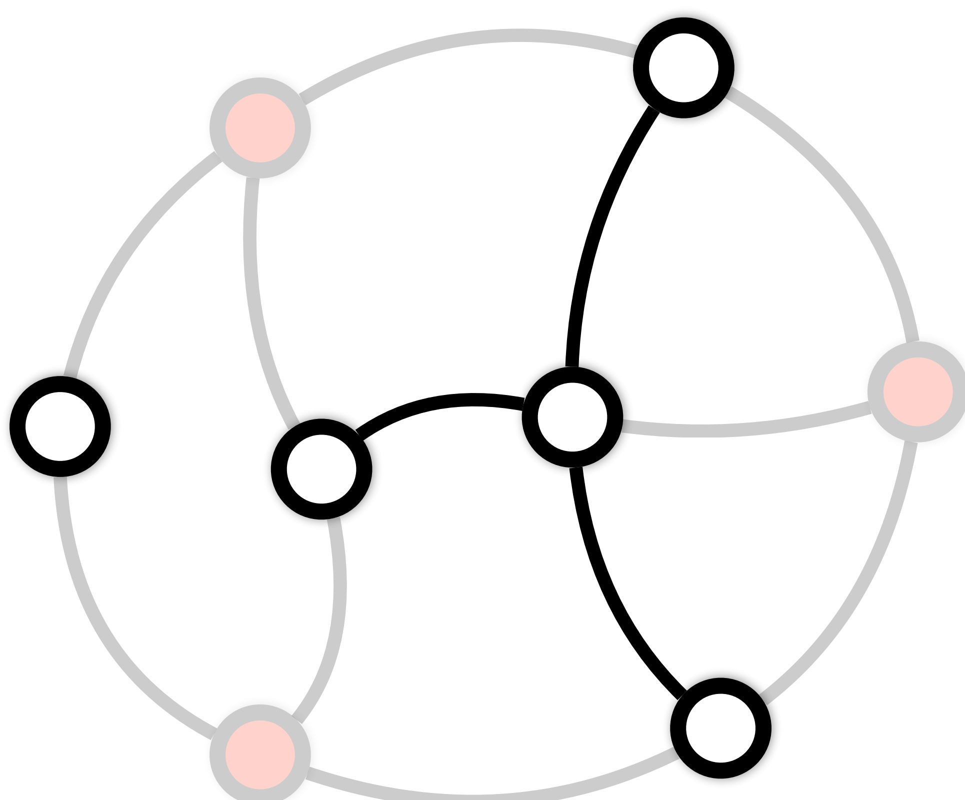
If no vertex exist with
degree more than $n^{1/2}$,
we can find a maximal
independent set of size $n^{1/2}$.
Color it with one fresh color.

Coloring a 3-colorable Graph with $O(n^{1/2})$ Colors



If no vertex exist with degree more than $n^{1/2}$, we can find a maximal independent set of size $n^{1/2}$. Color it with one fresh color.

Coloring a 3-colorable Graph with $O(n^{1/2})$ Colors



If no vertex exist with degree more than $n^{1/2}$, we can find a maximal independent set of size $n^{1/2}$. Color it with one fresh color.

Repeat strategy on remaining graph.

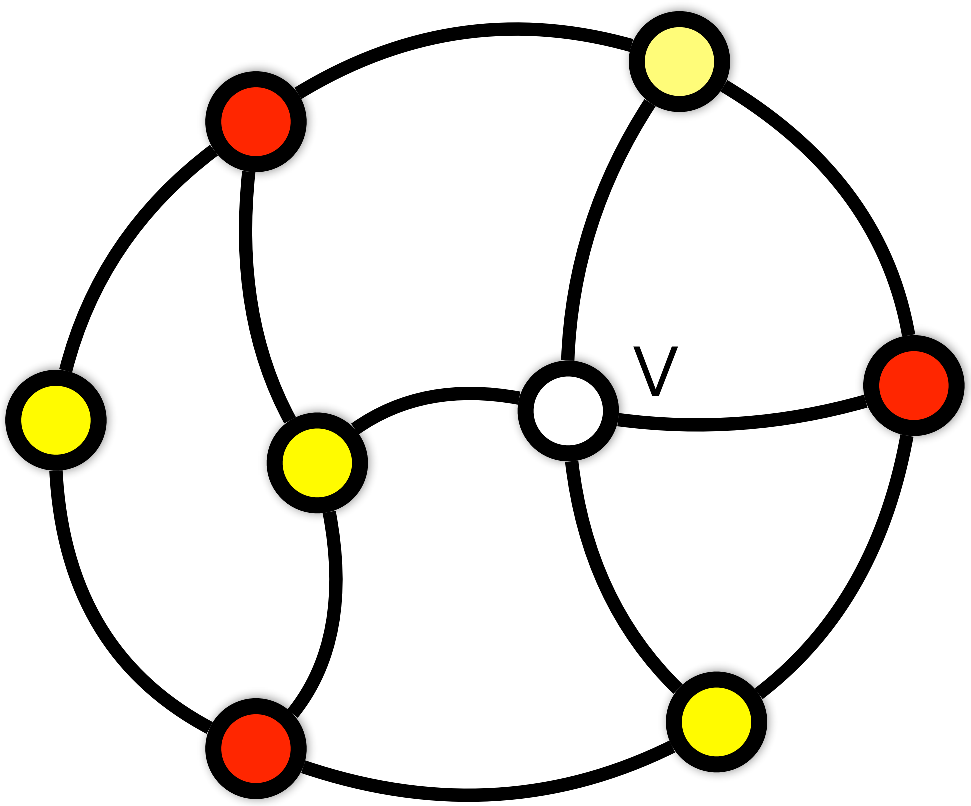
Color Usage Analysis

- In the first rule, we color $n^{1/2}$ vertices with 3 new colors.
- In the second rule, we color $n^{1/2}$ vertices with 1 new color.
- There can be at most $n^{1/2}$ steps before all vertices have been colored, in total $O(n^{1/2})$ colors used.

Exponential Time Algorithms

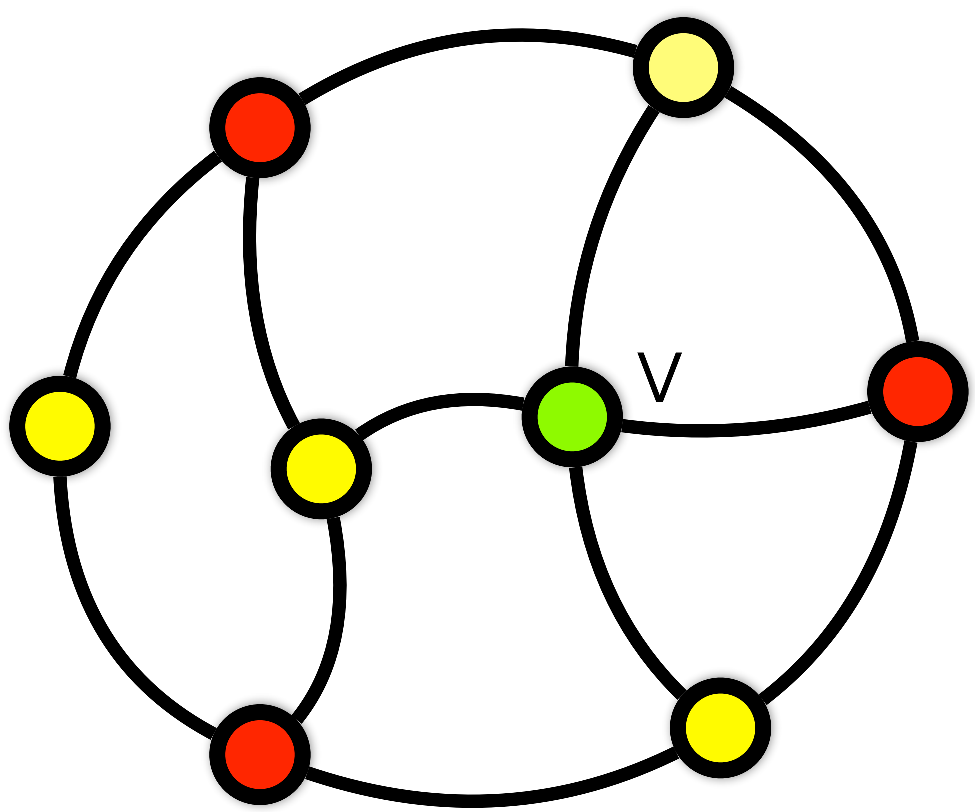
- What is the fastest worst case running time algorithm we can guarantee for *exactly* computing the chromatic number?
  - We want to make sure that if a k-coloring exists the algorithm will learn about it, not just an approximation.
  - We want to guarantee that the running time is bounded by as small as possible an (exponential) function in n, the number of vertices.

Naive algorithm uses k^n time.



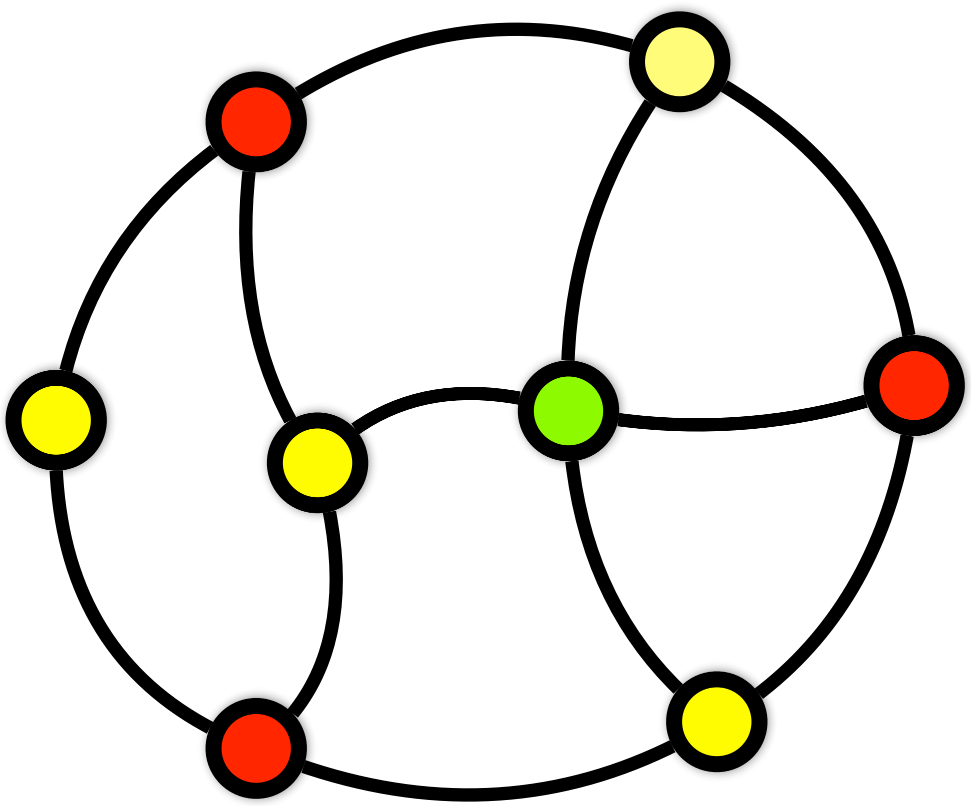
For every vertex, try each of the k colors.

Naive algorithm uses k^n time.



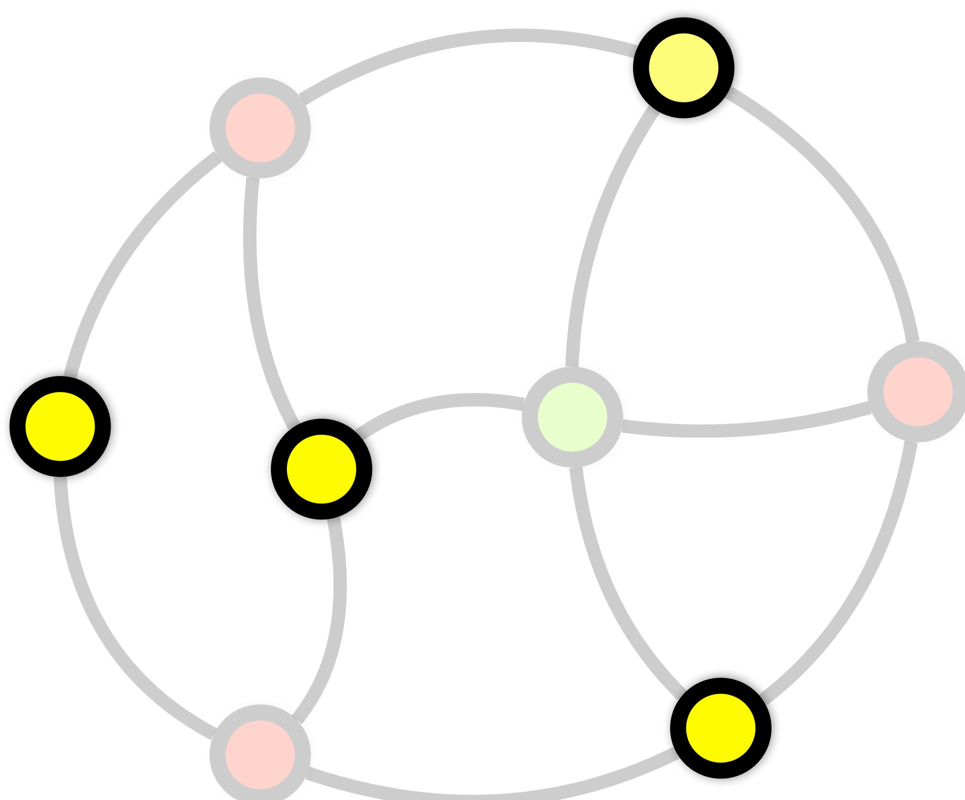
v cannot be yellow or red => we might need to check k colors for many vertices, and backtrack many times.

Colorclasses



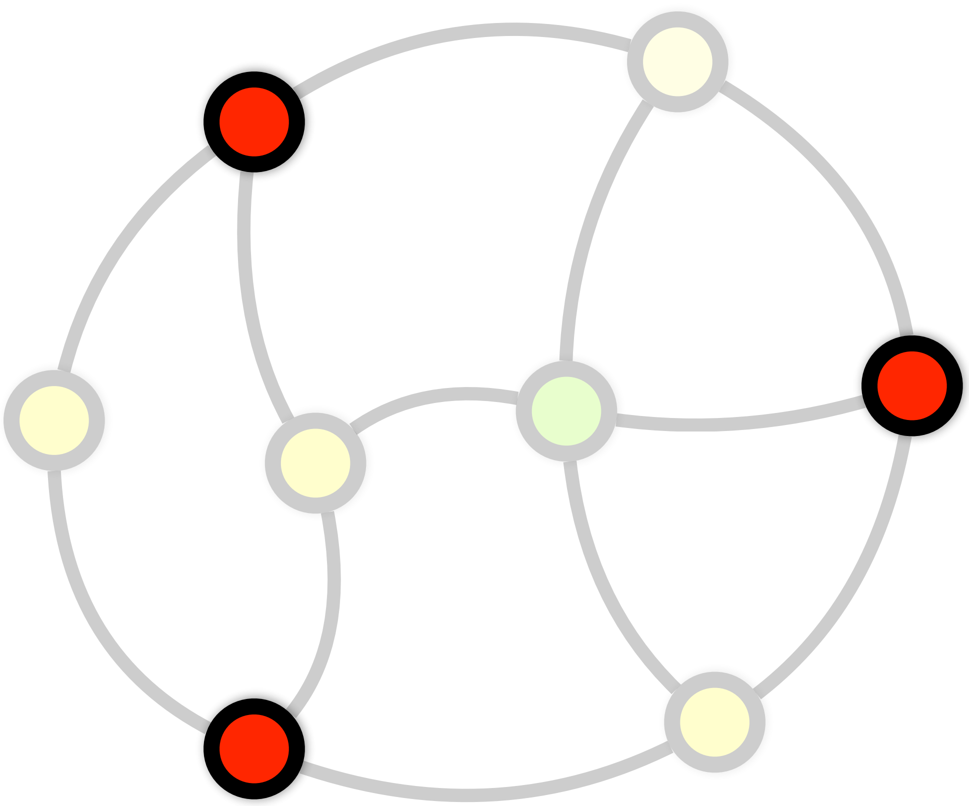
A *colorclass* is the set of vertices colored by the same color.
A k-coloring consists of k disjoint colorclasses.

Colorclasses



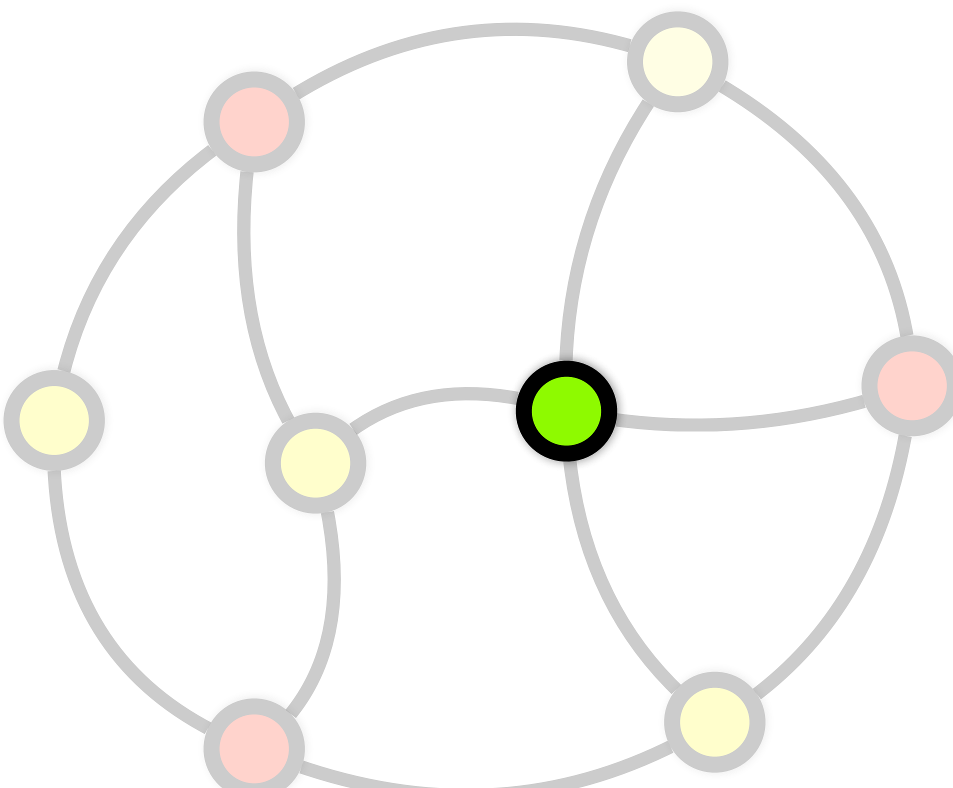
The Yellow colorclass

Colorclasses



The Red colorclass.

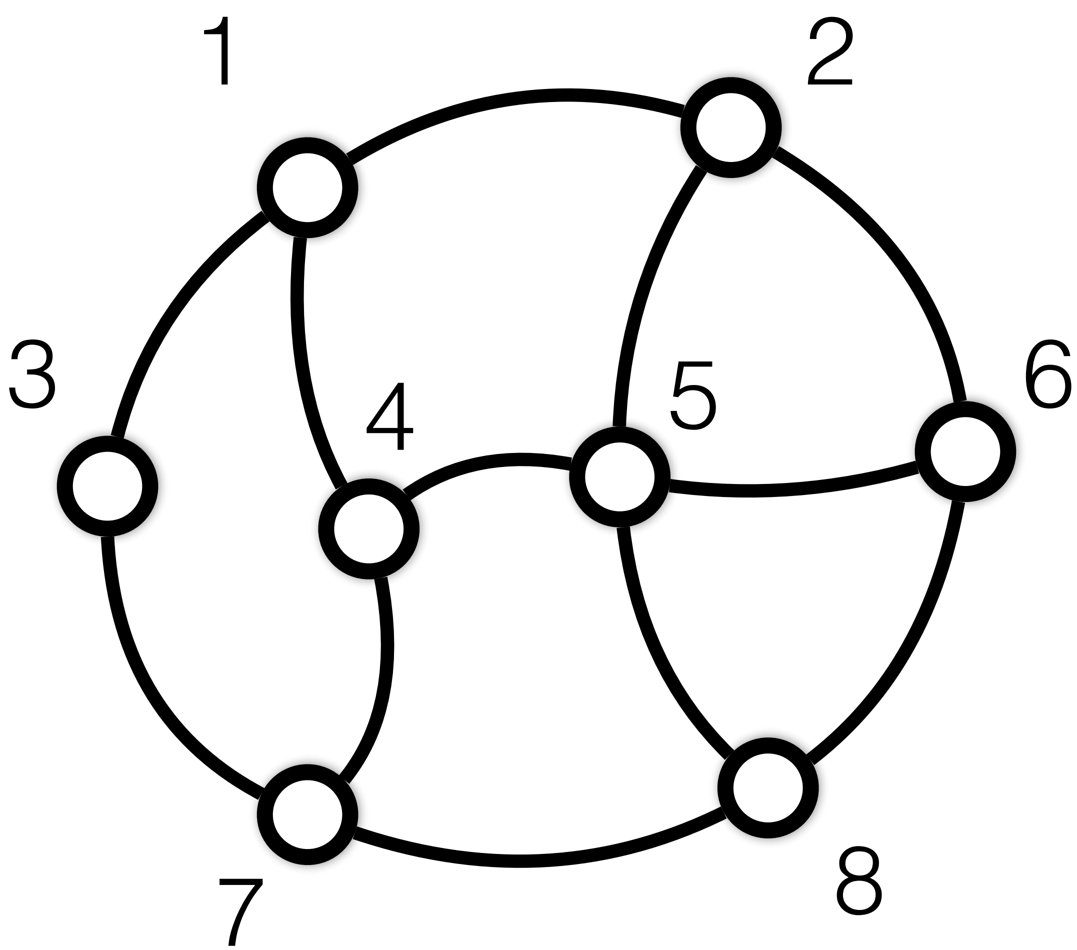
Colorclasses



The Green colorclass.

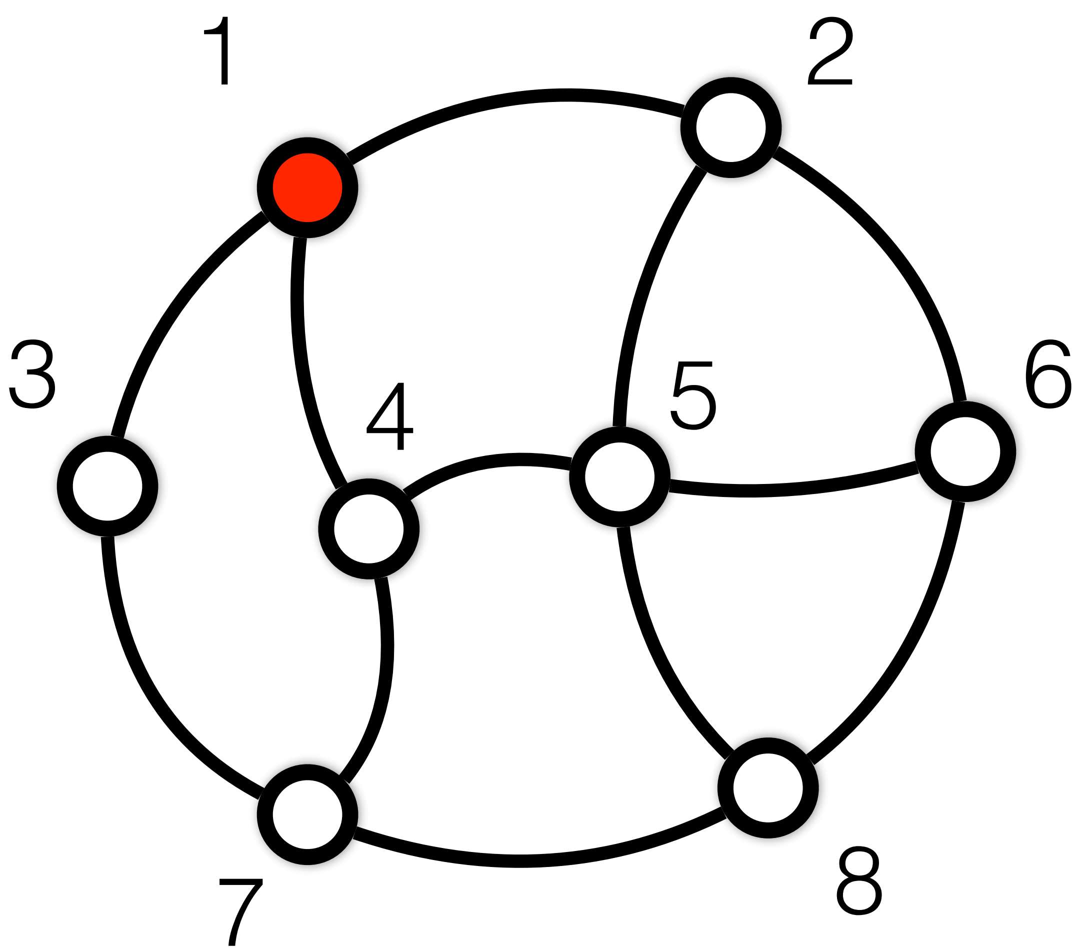
All Candidate Colorclasses

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8

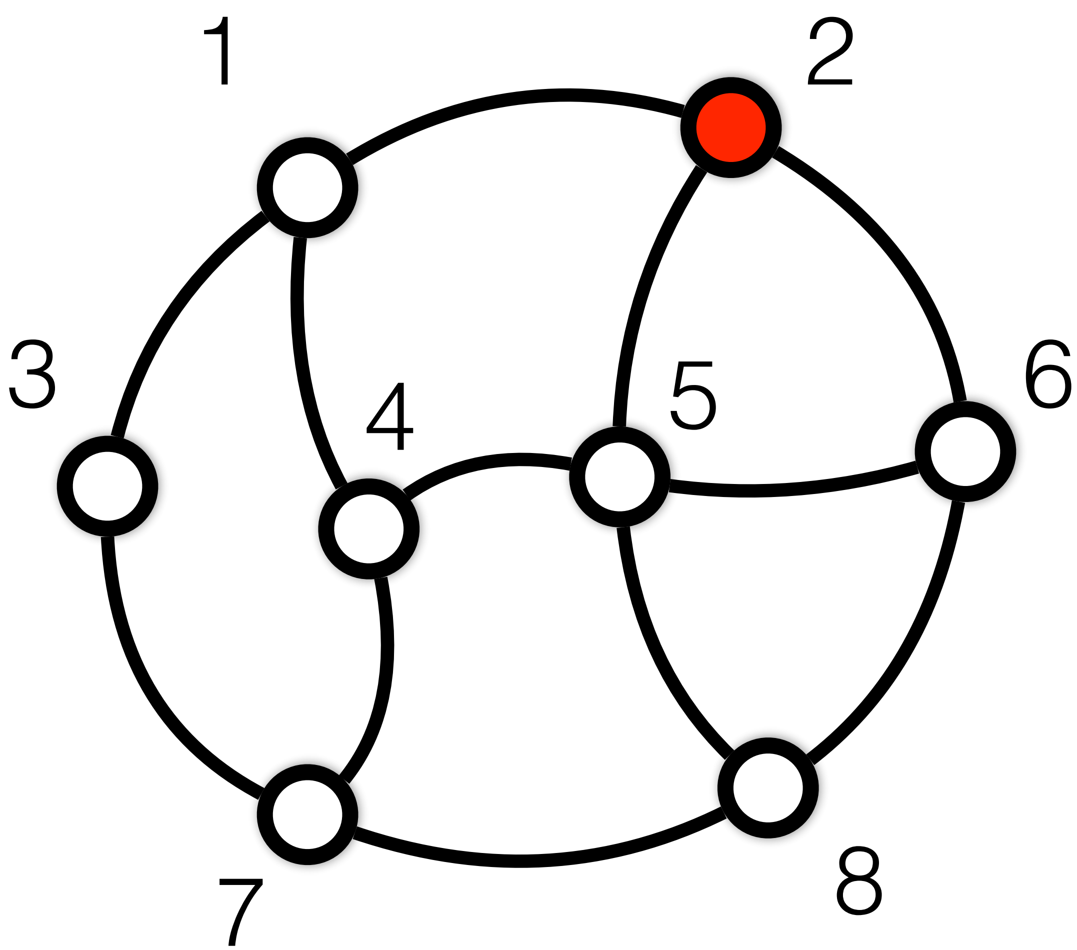


All Candidate Colorclasses

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								

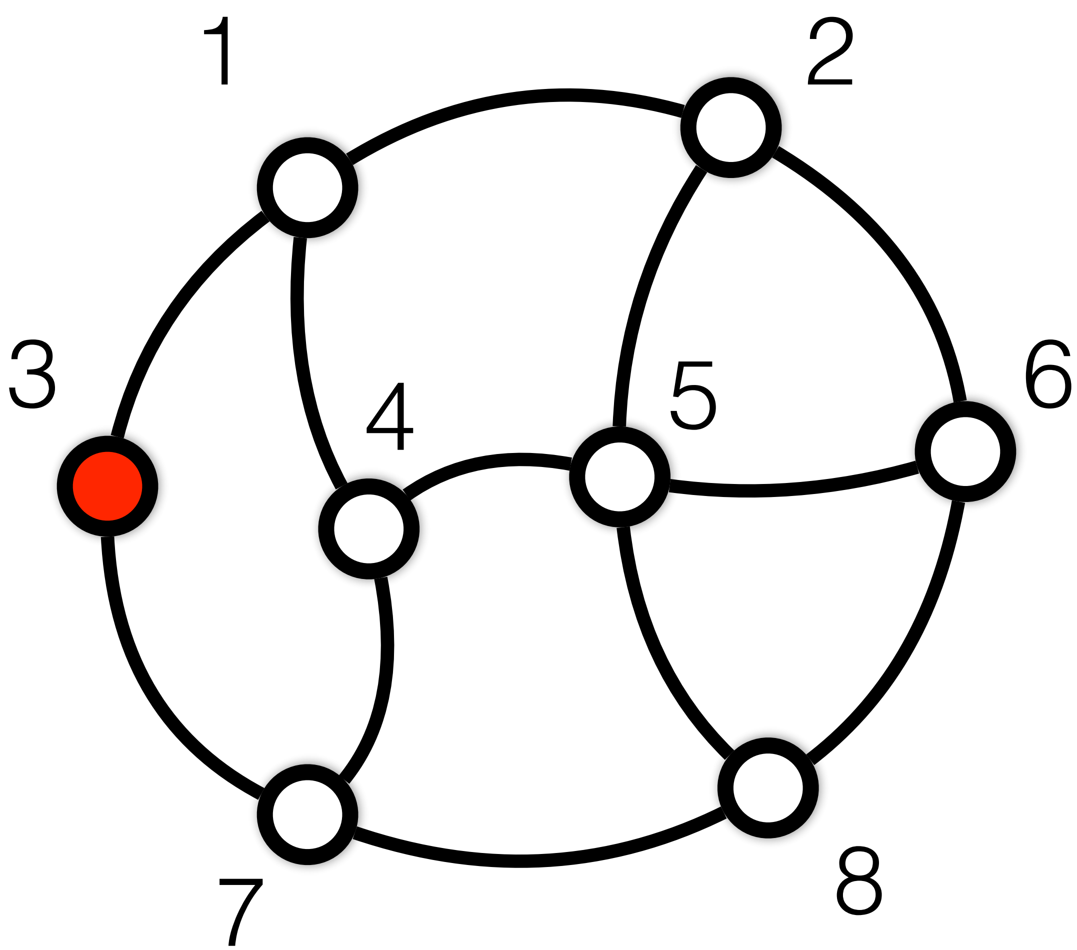


All Candidate Colorclasses



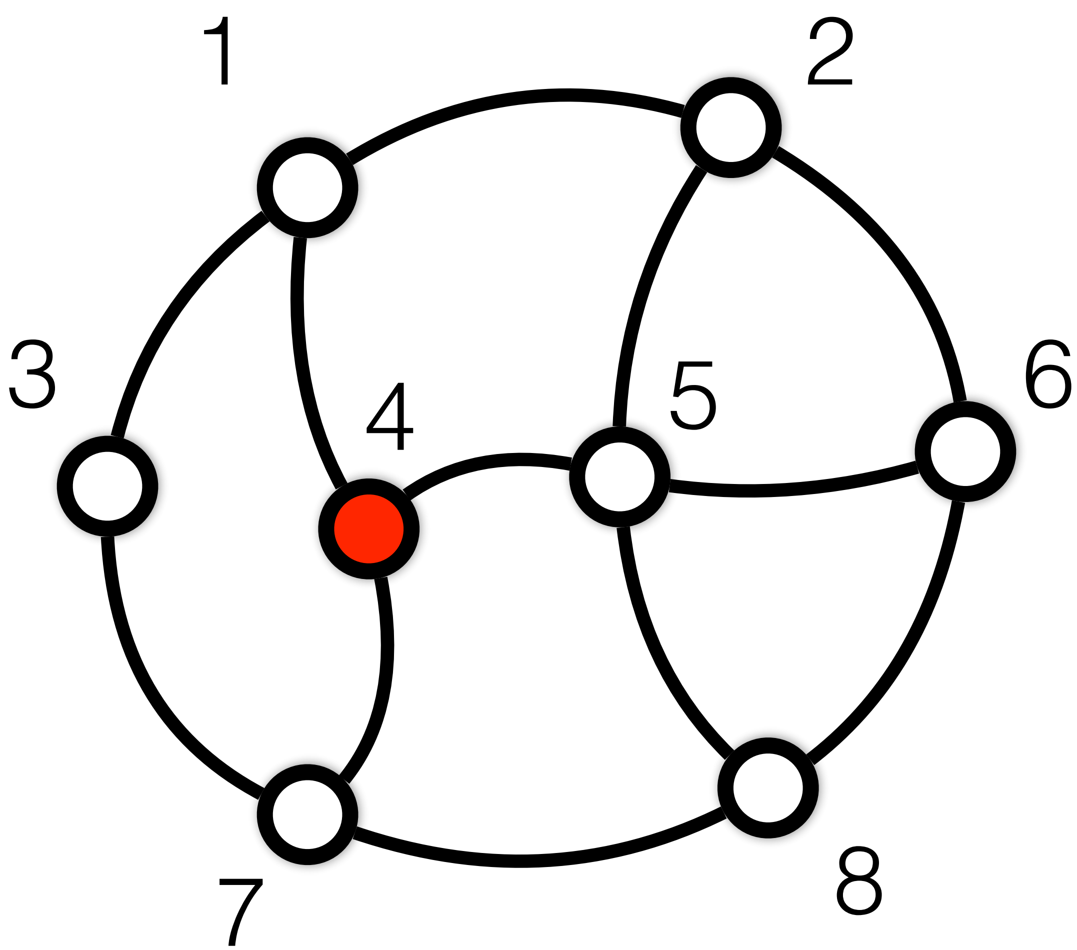
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								

All Candidate Colorclasses



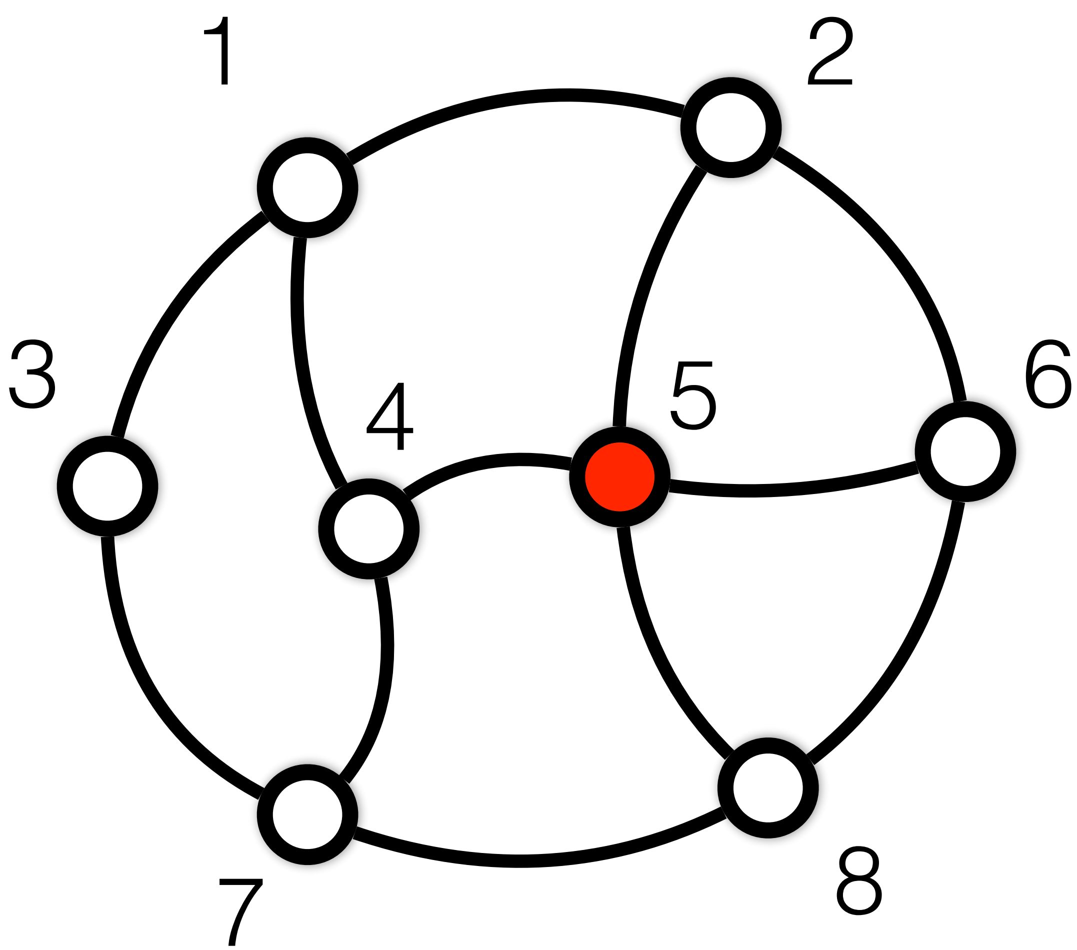
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								

All Candidate Colorclasses



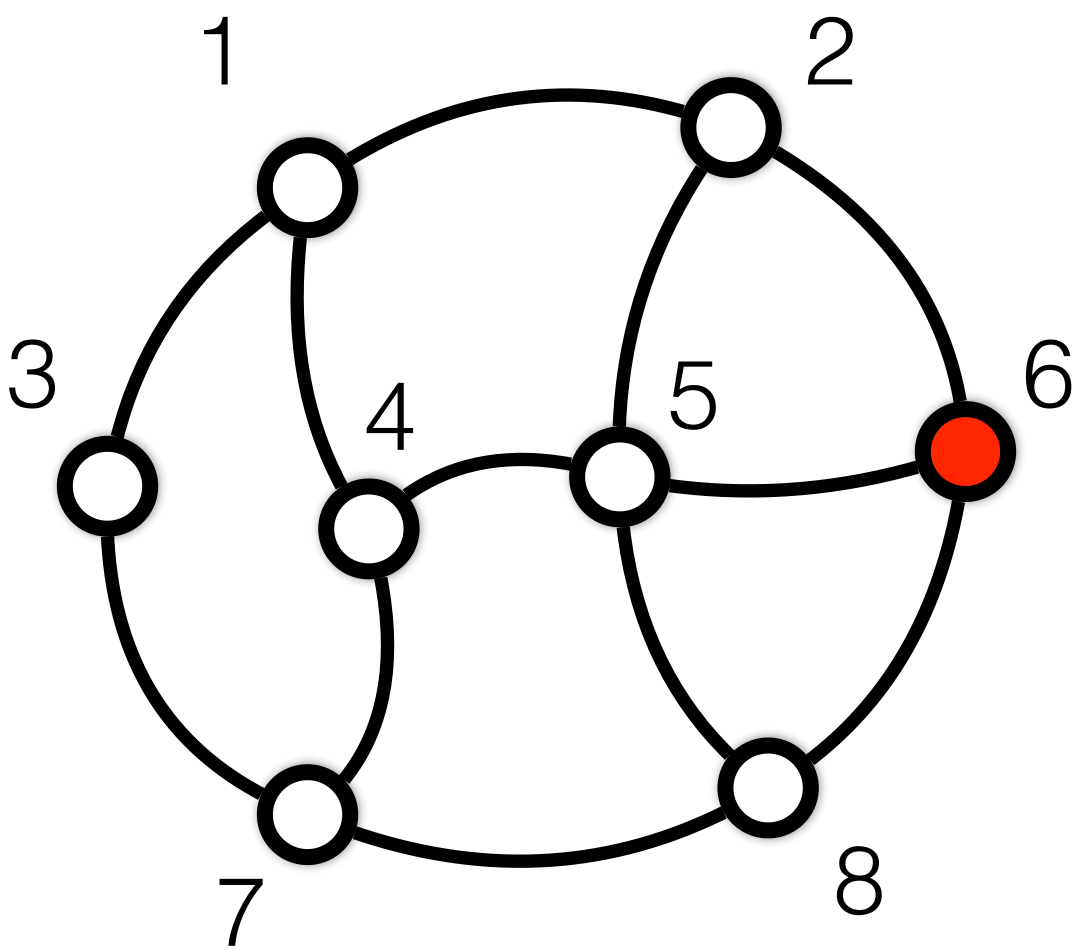
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								

All Candidate Colorclasses



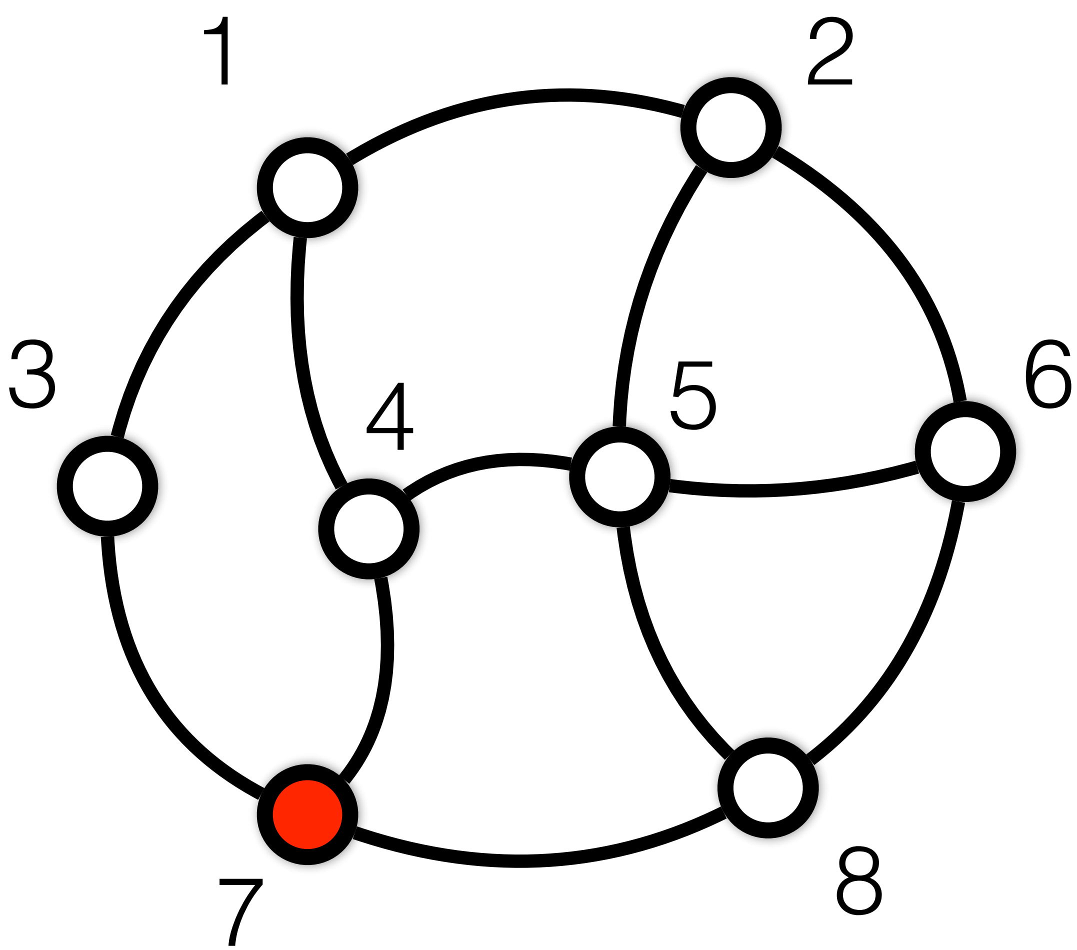
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								

All Candidate Colorclasses



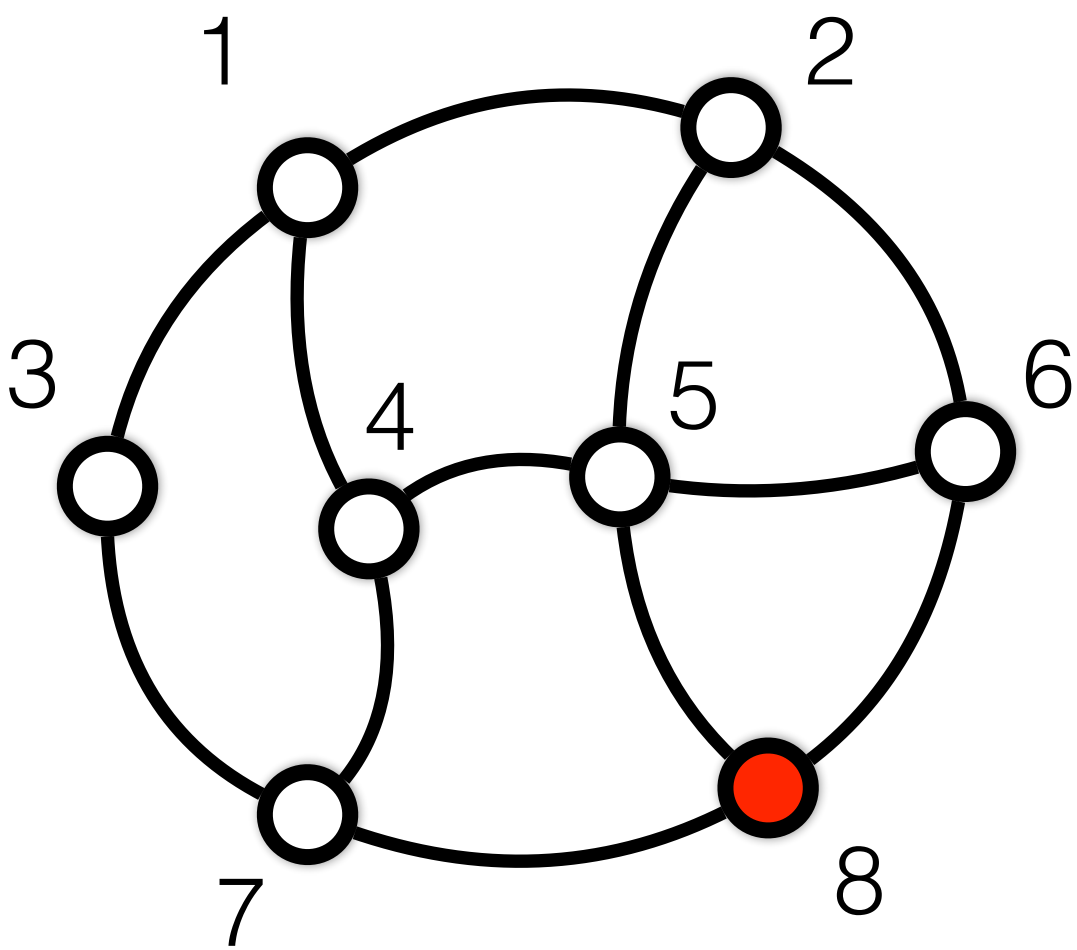
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								

All Candidate Colorclasses



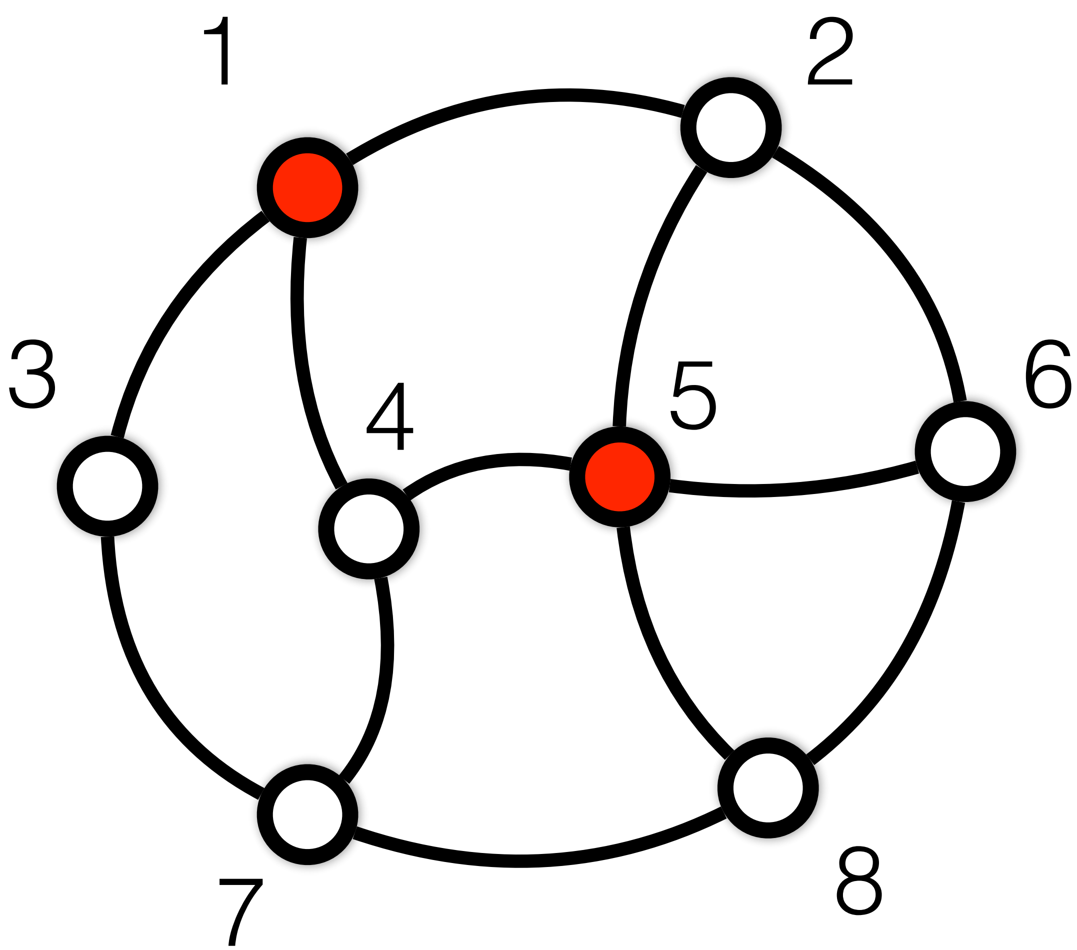
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								

All Candidate Colorclasses



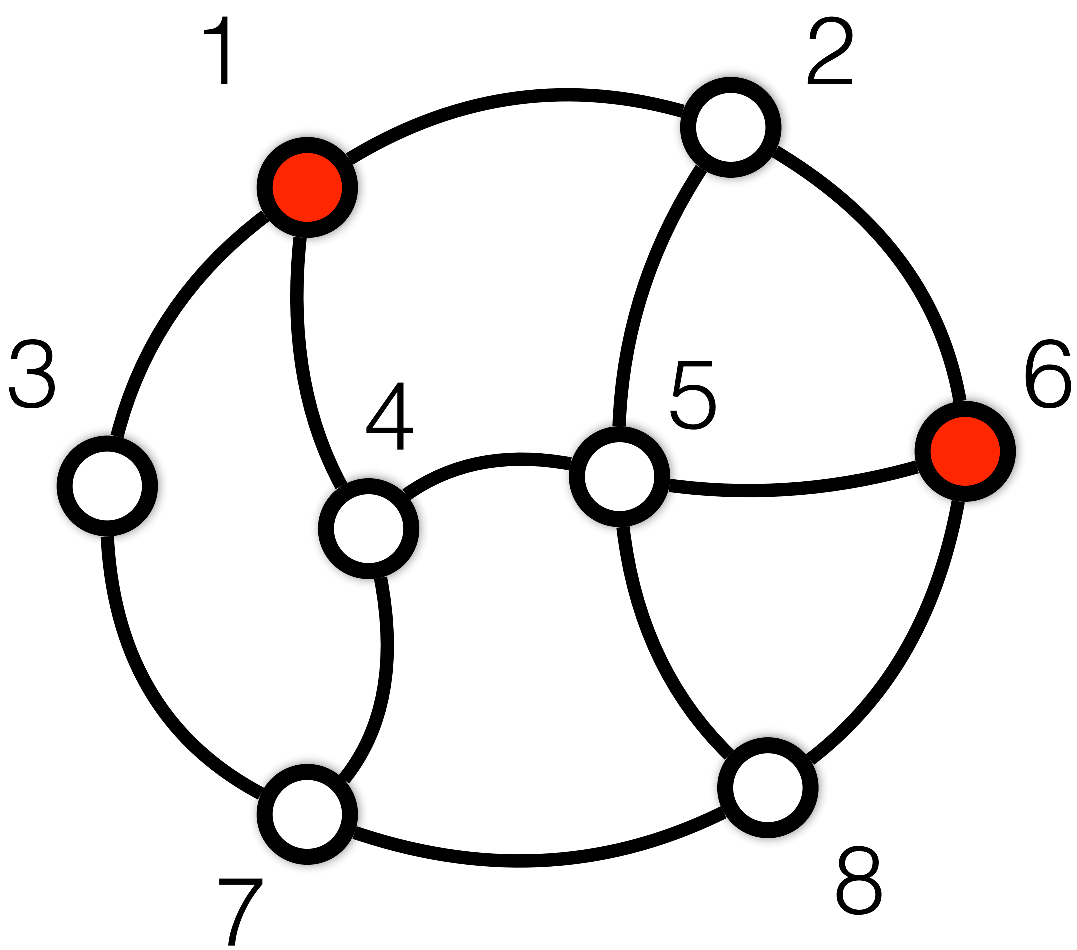
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								

All Candidate Colorclasses



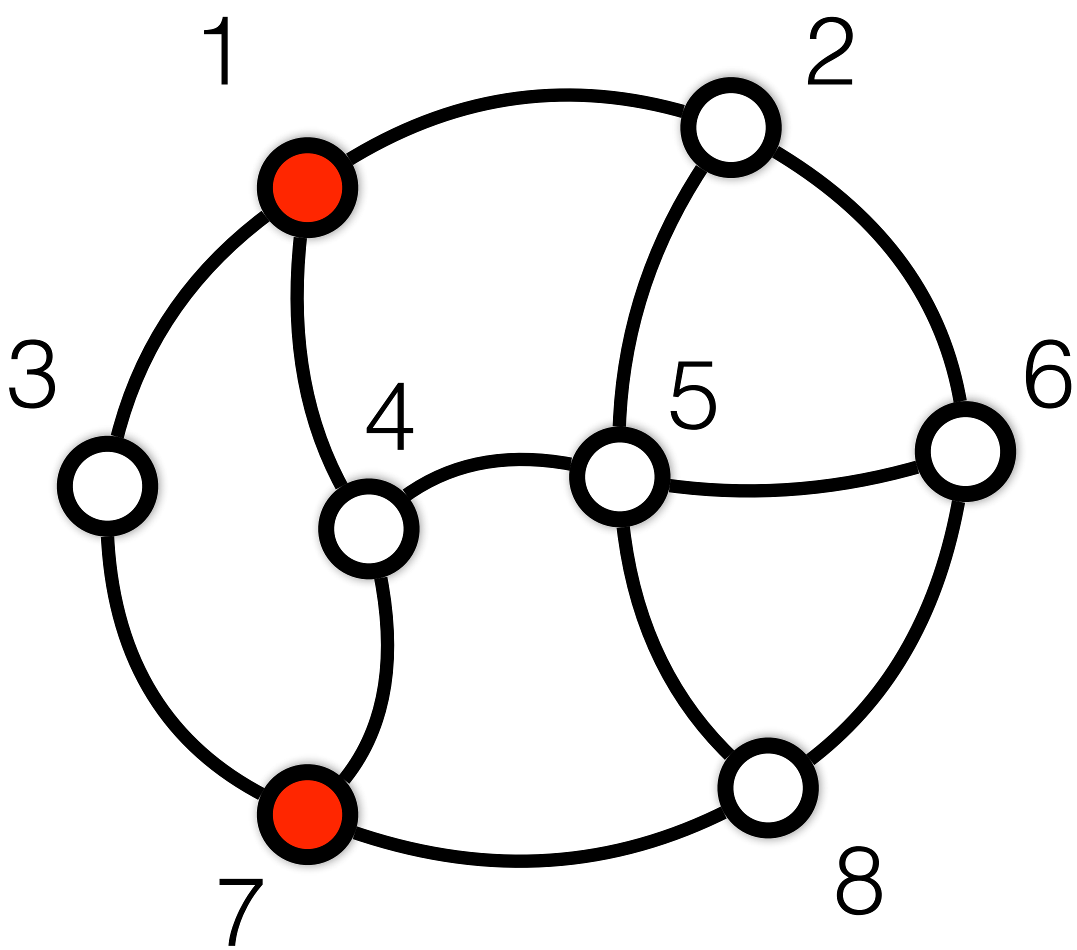
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								

All Candidate Colorclasses



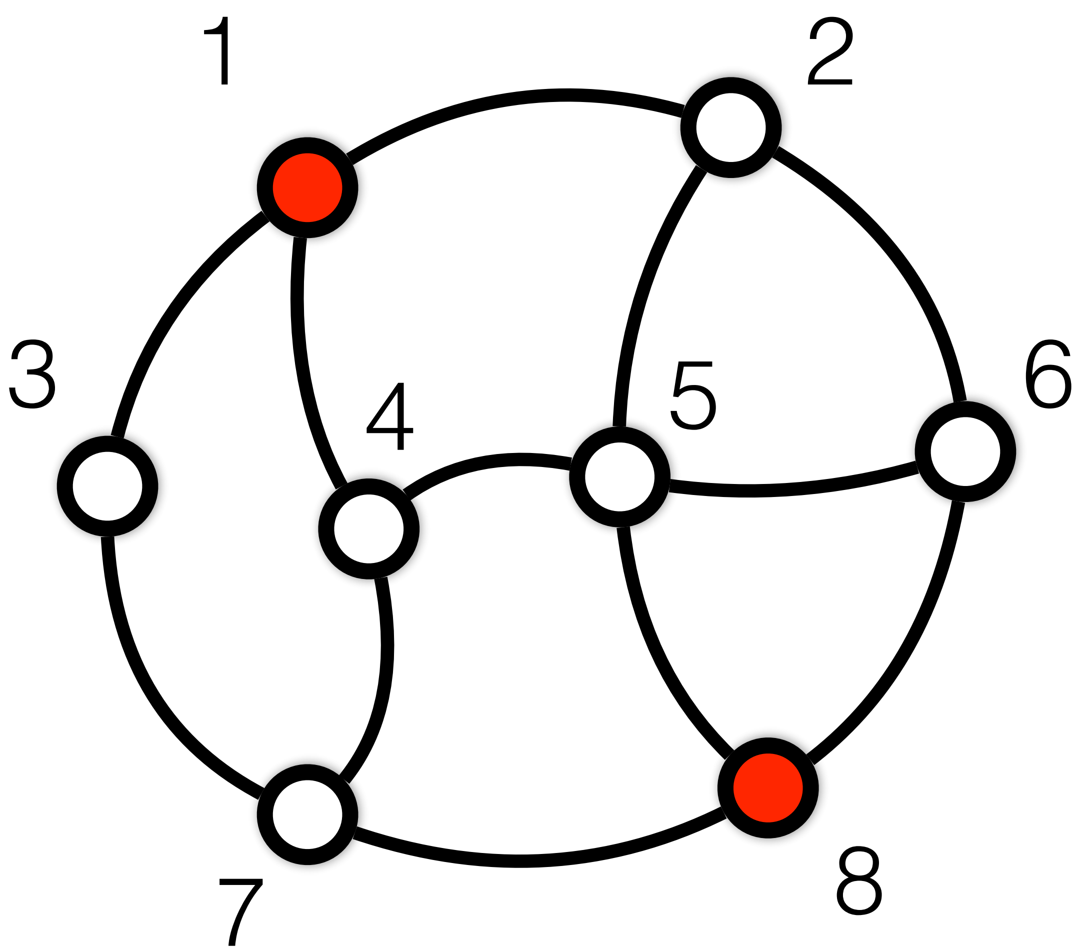
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								

All Candidate Colorclasses



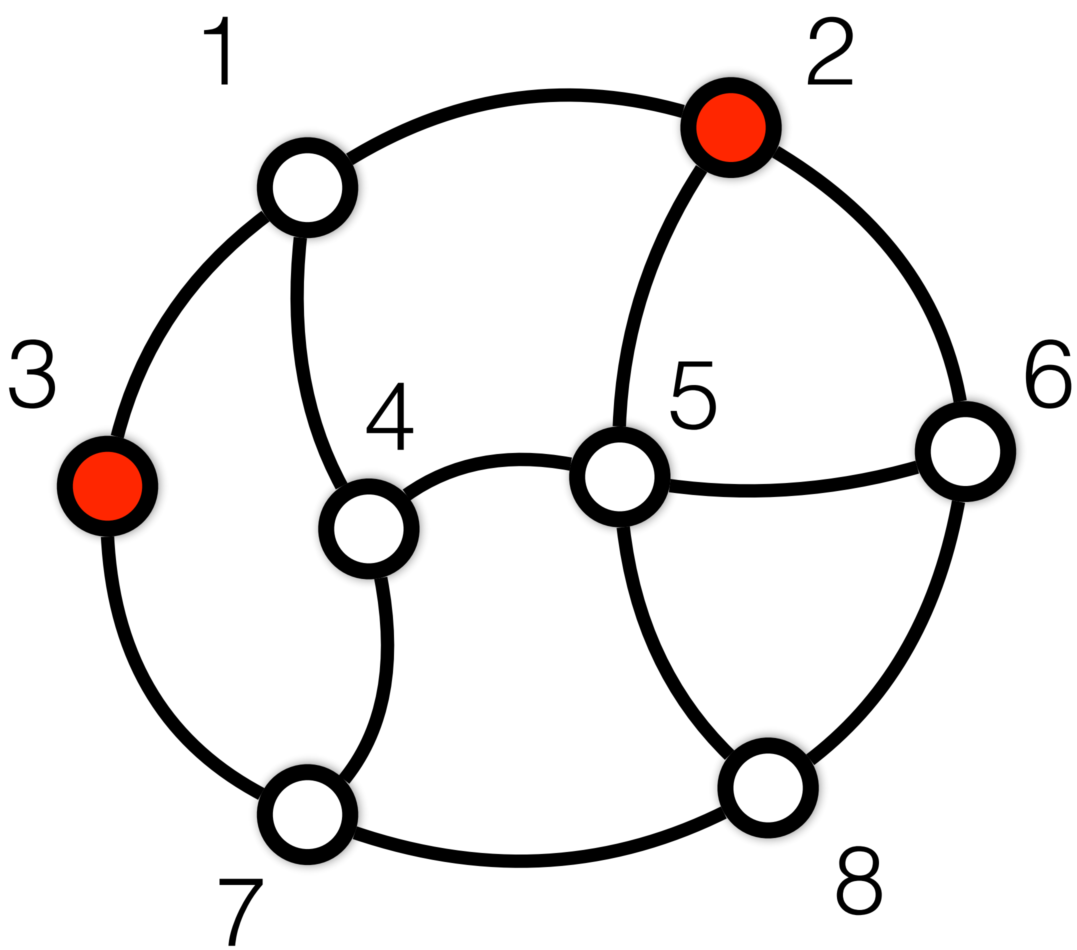
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								

All Candidate Colorclasses



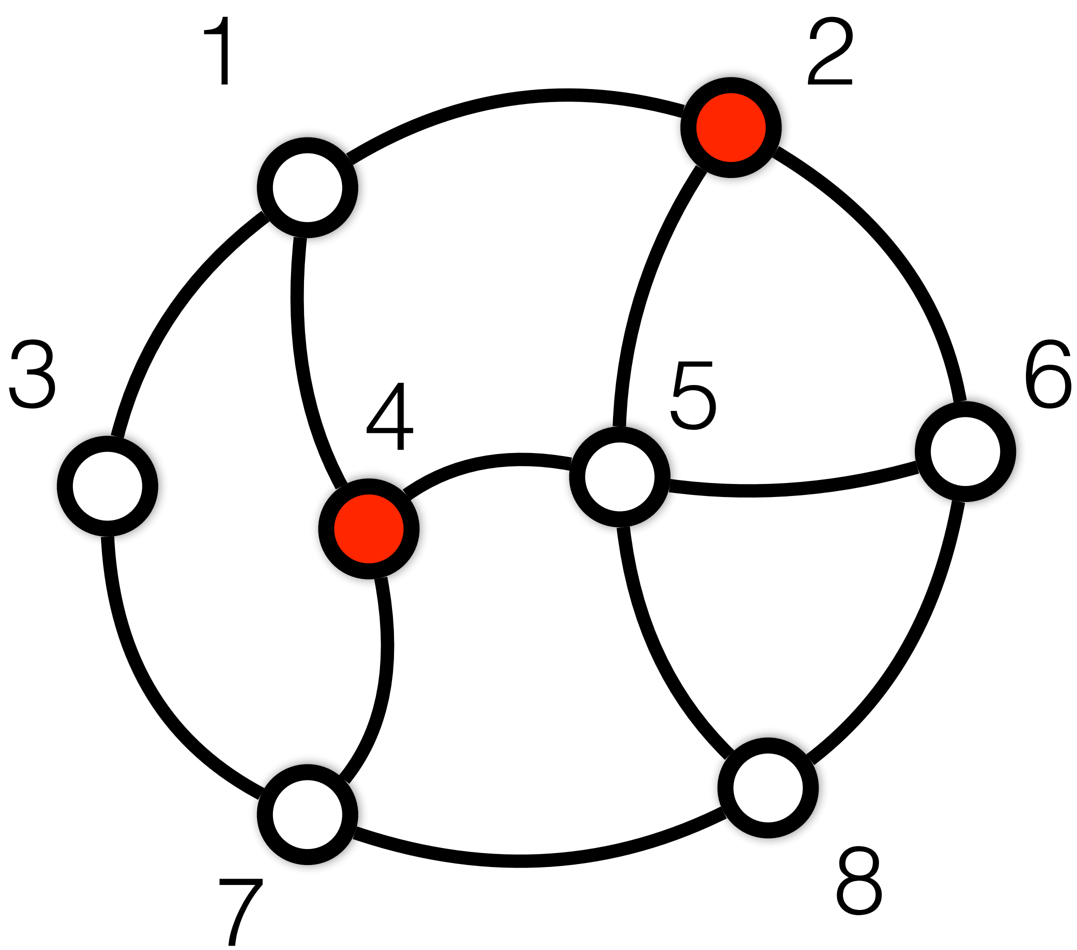
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								

All Candidate Colorclasses



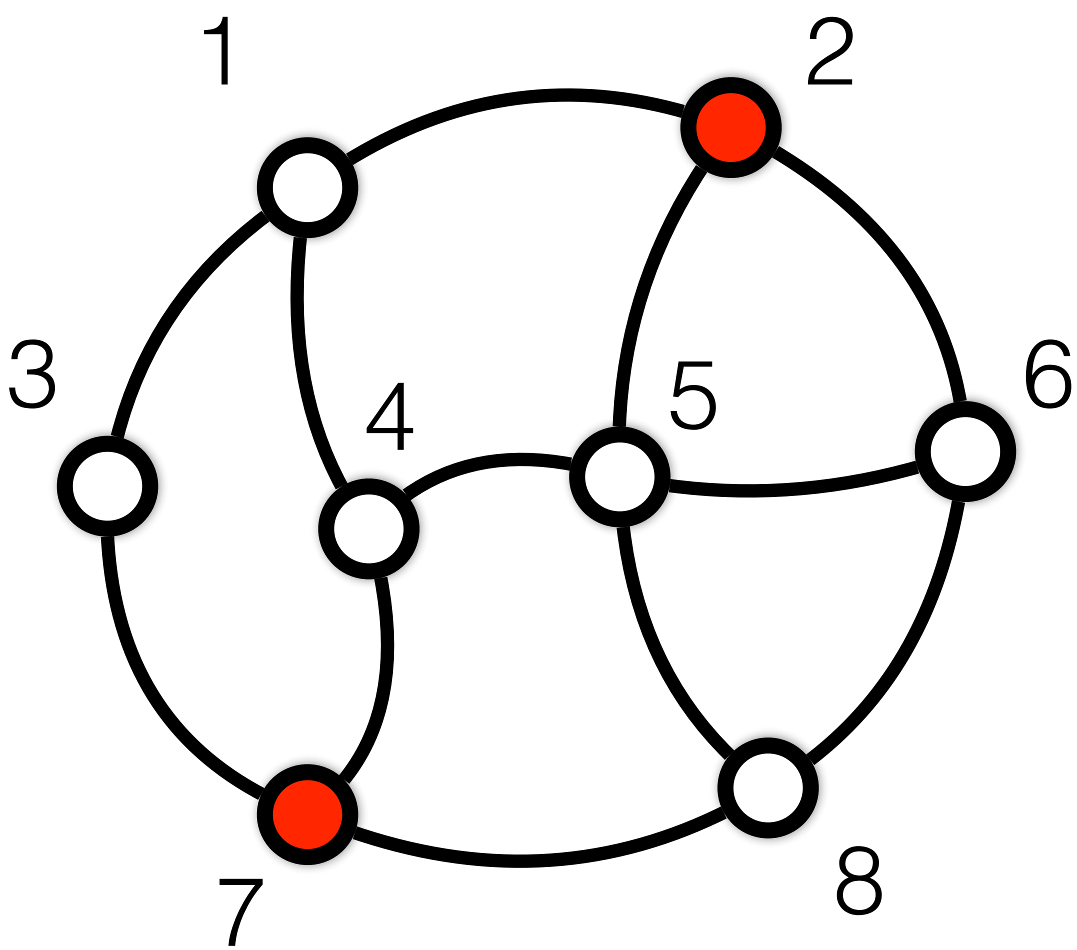
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								

All Candidate Colorclasses



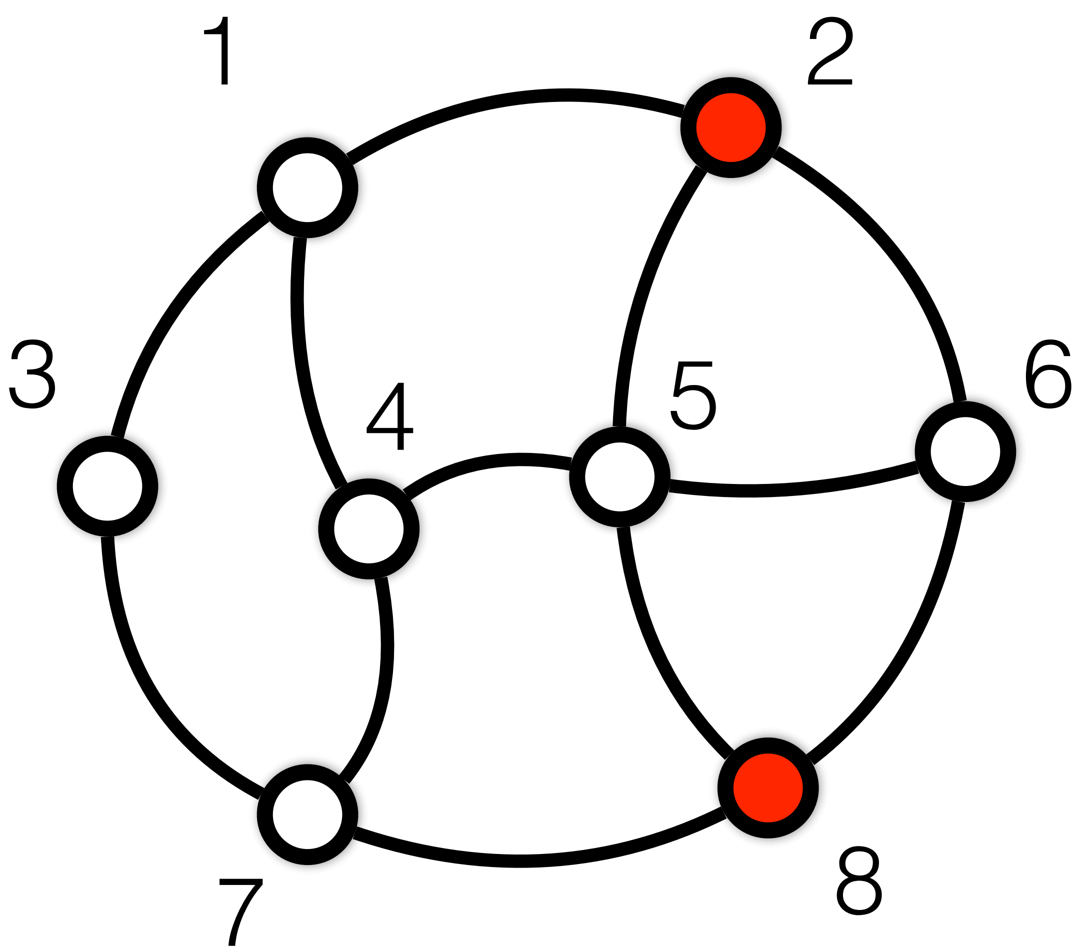
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								

All Candidate Colorclasses



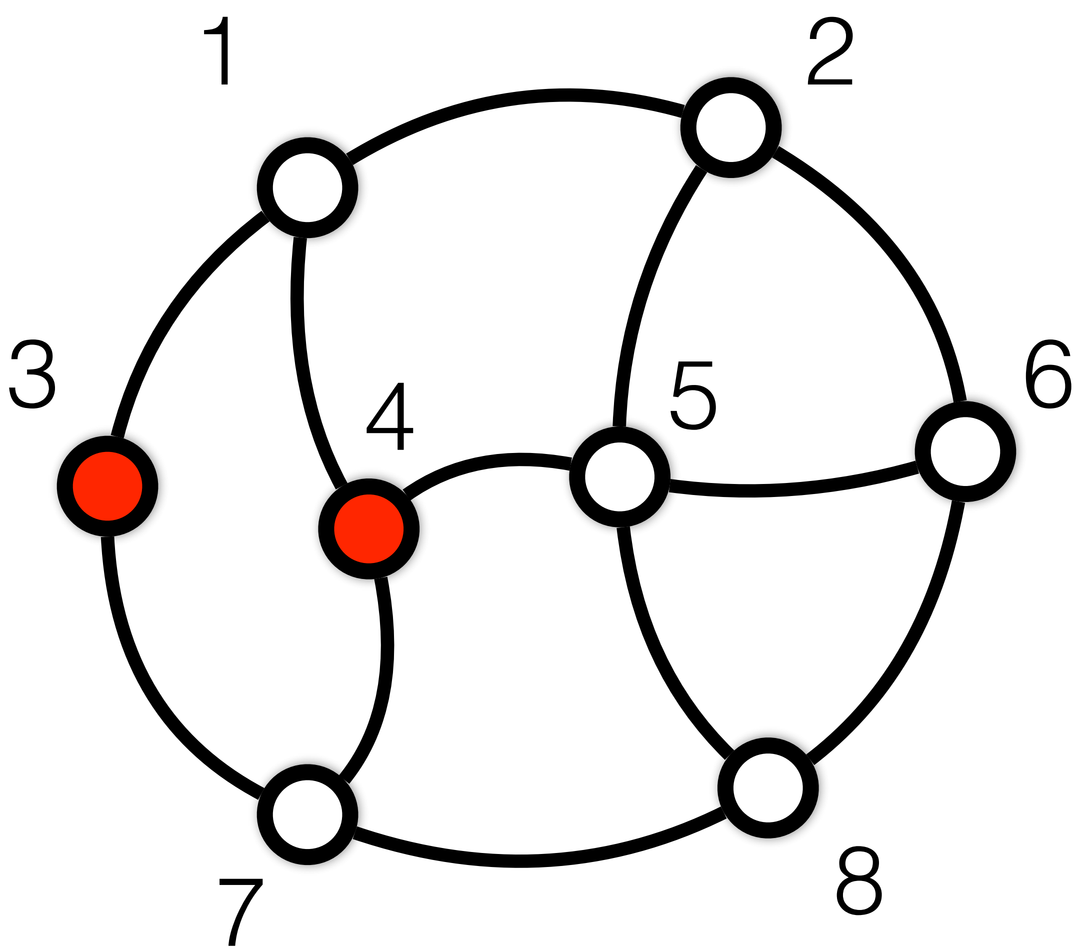
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								

All Candidate Colorclasses



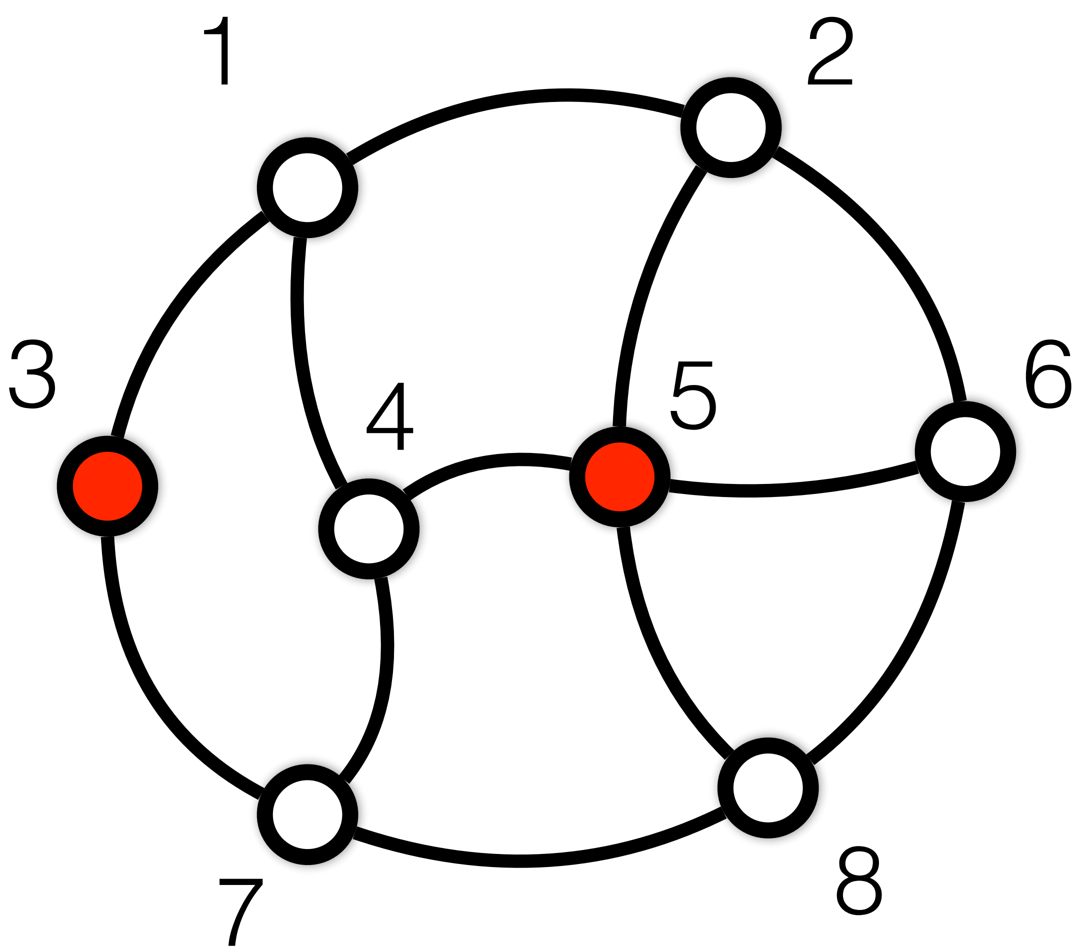
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0								
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



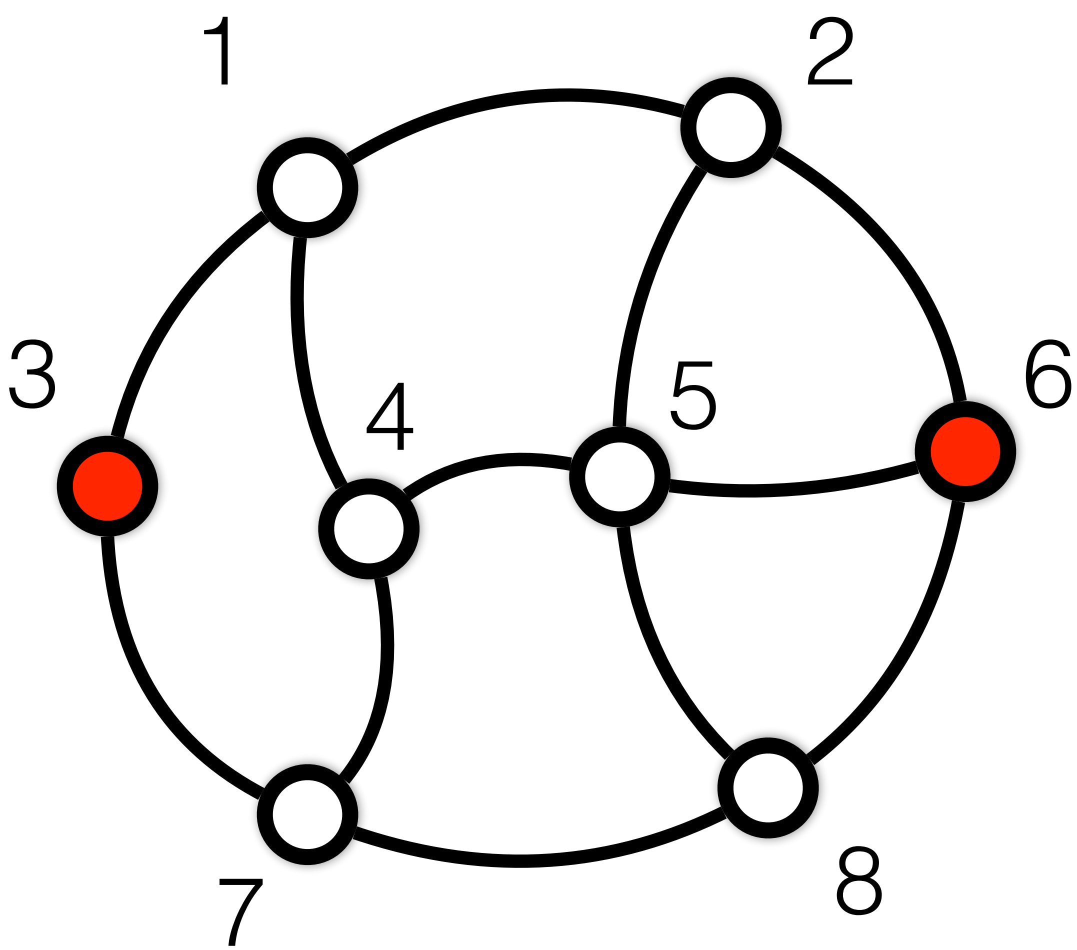
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0								
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



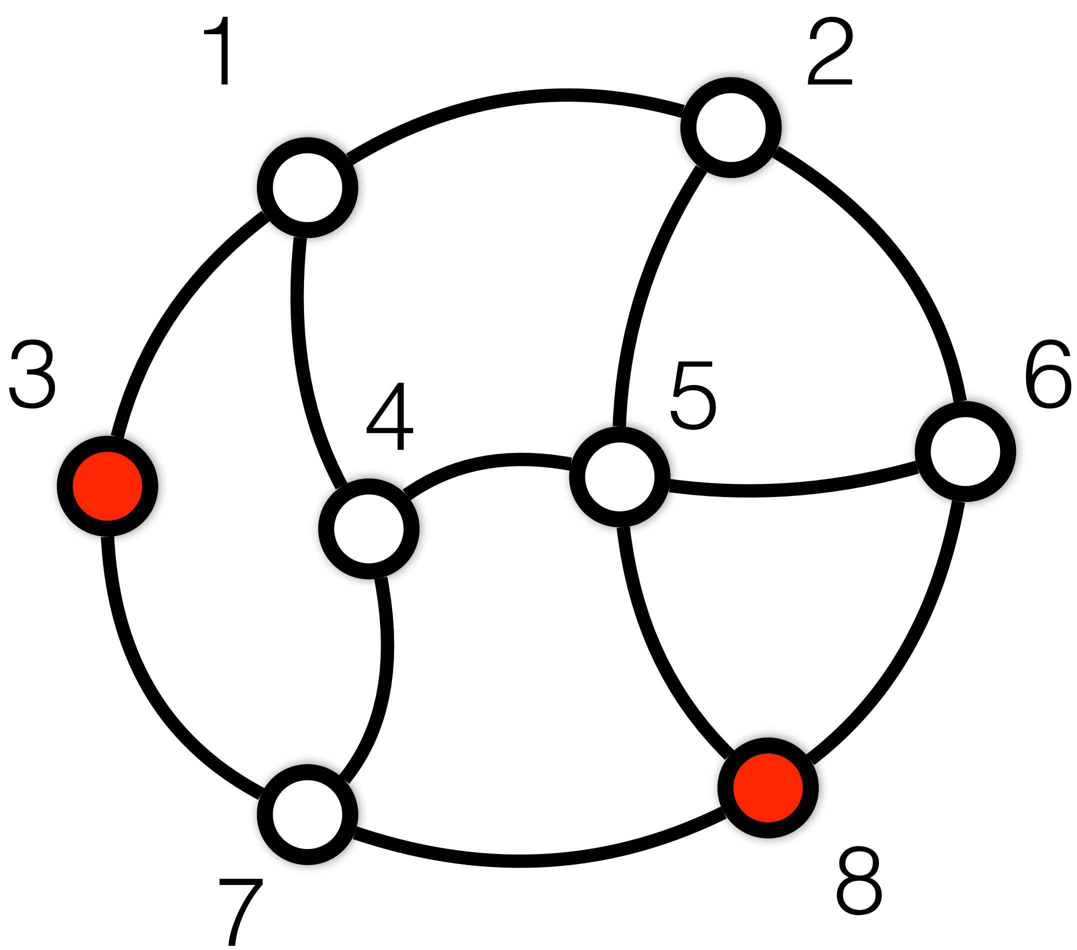
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0								
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



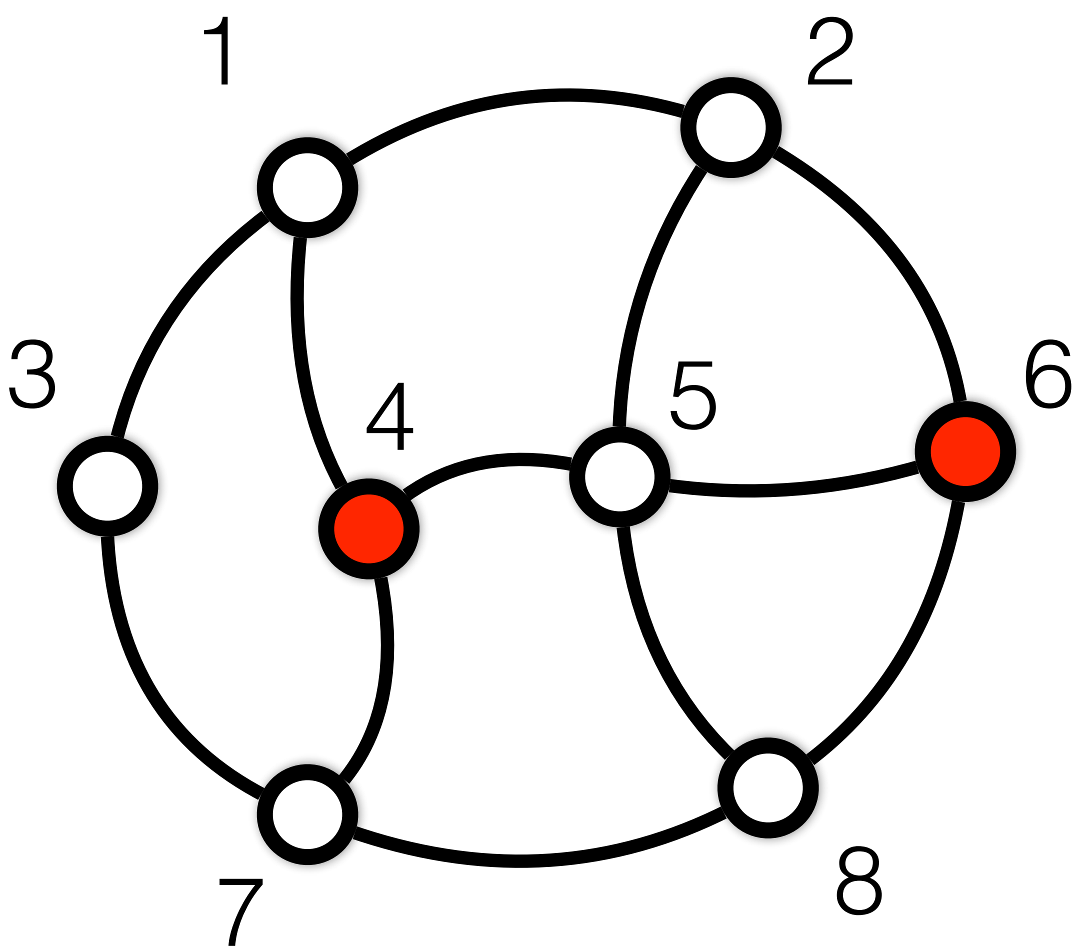
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0								
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



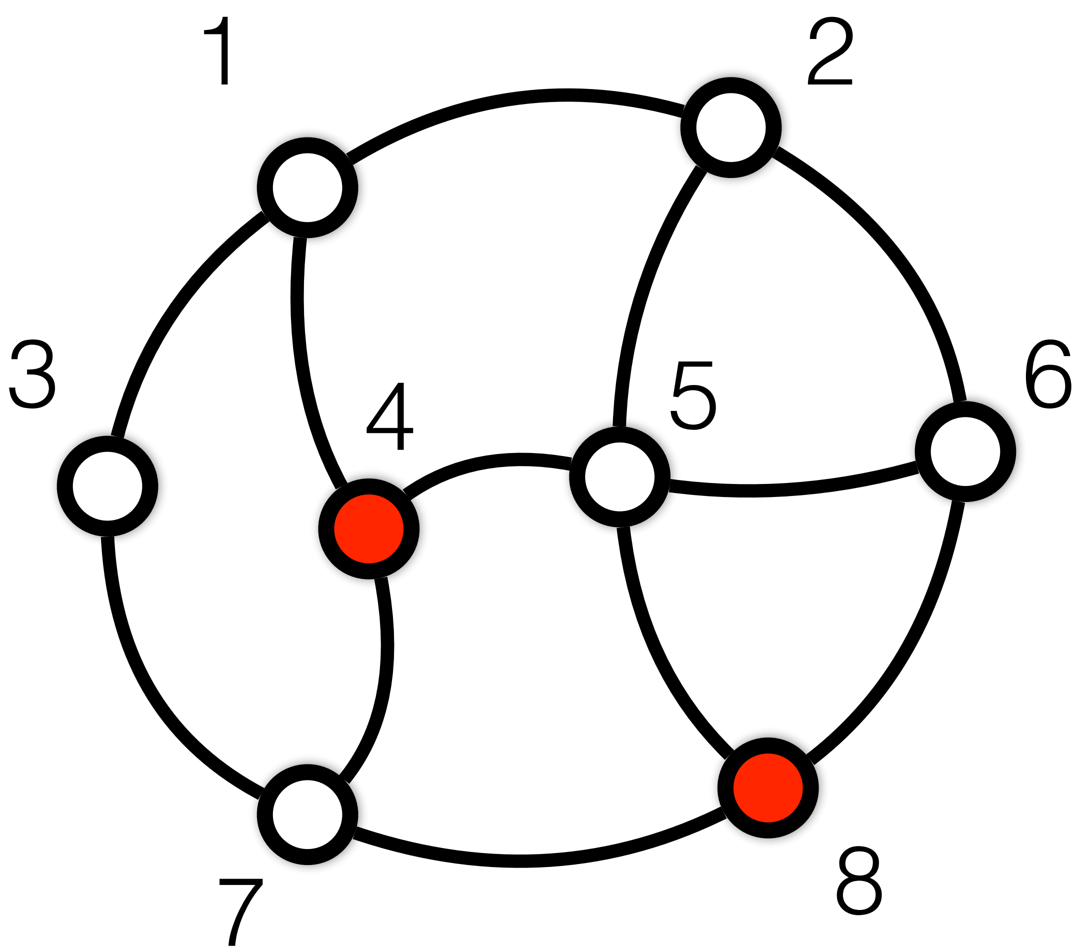
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0								
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0								
0	0	0	0	0	0	1	0								
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

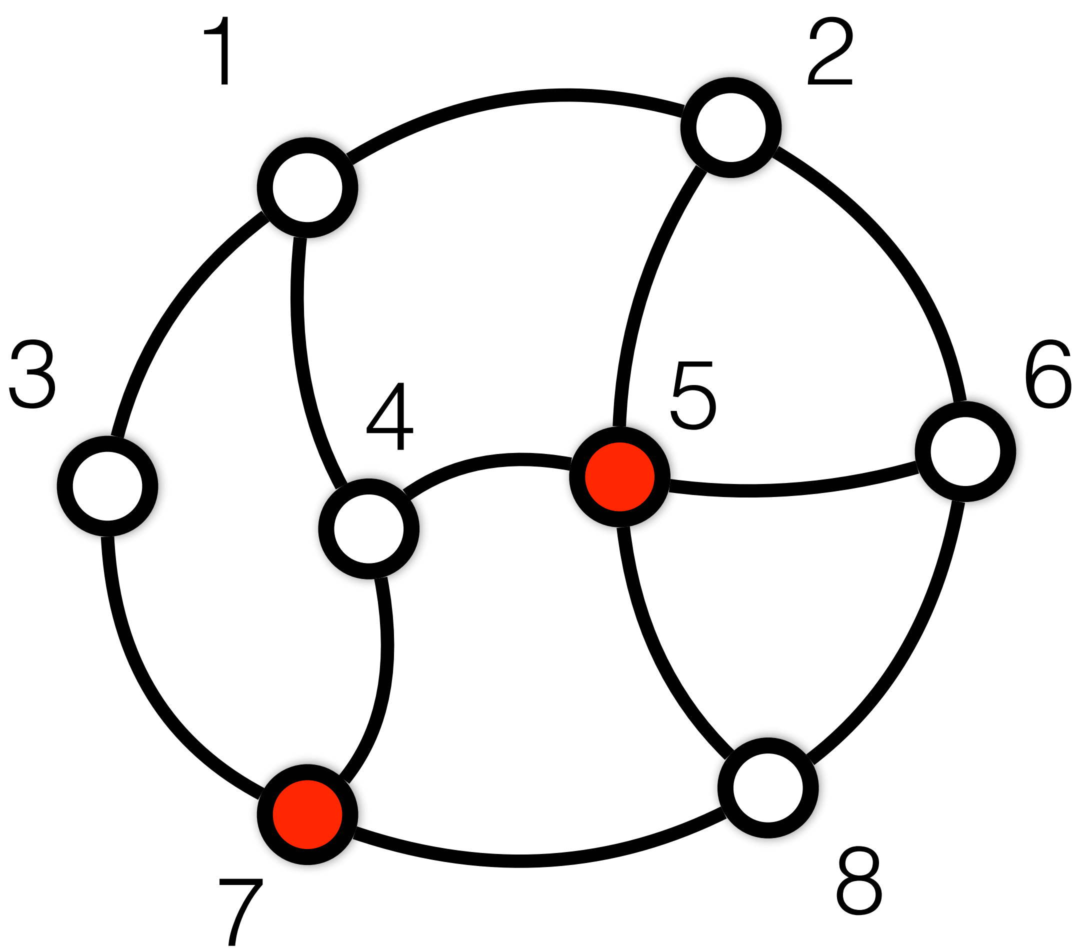
All Candidate Colorclasses



1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	0	1

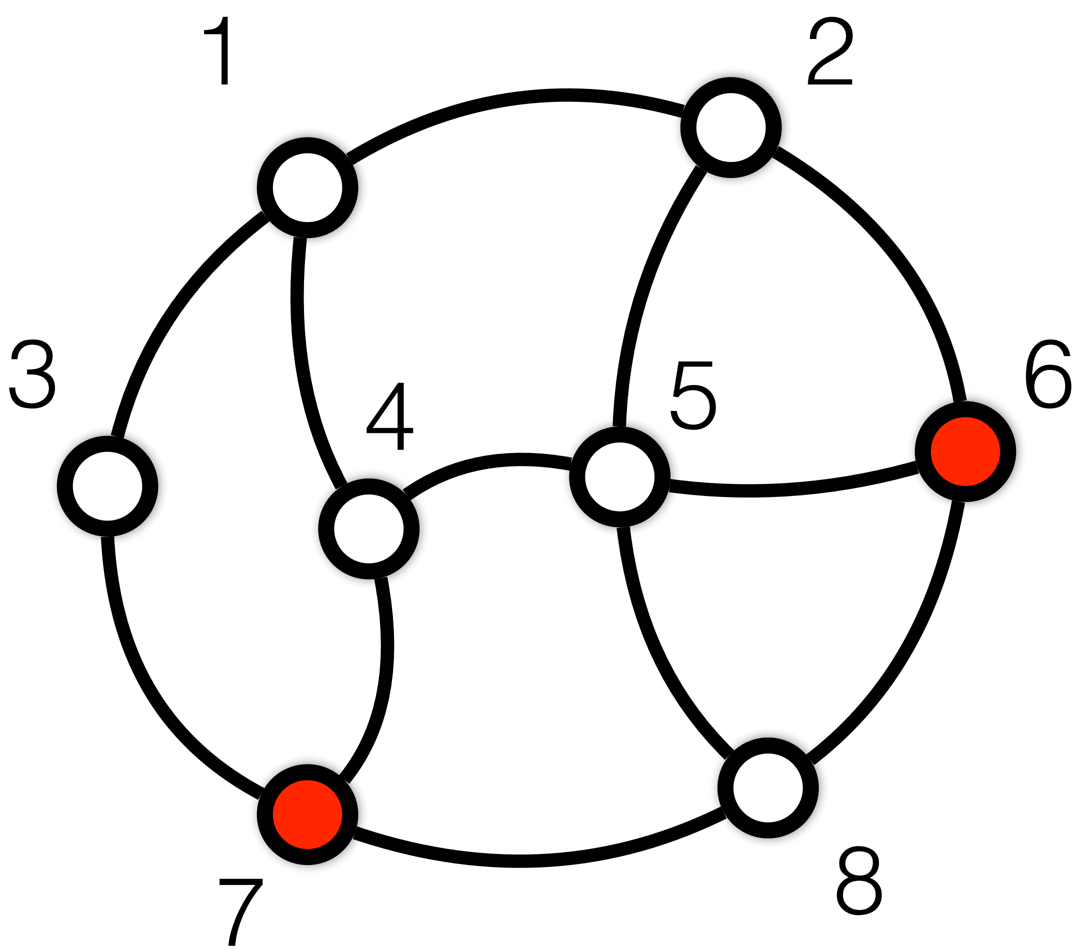
1	2	3	4	5	6	7	8
0	0	1	1	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	0
0	0	1	0	0	0	0	1
0	0	0	1	0	1	0	0
0	0	0	1	0	0	0	1

All Candidate Colorclasses



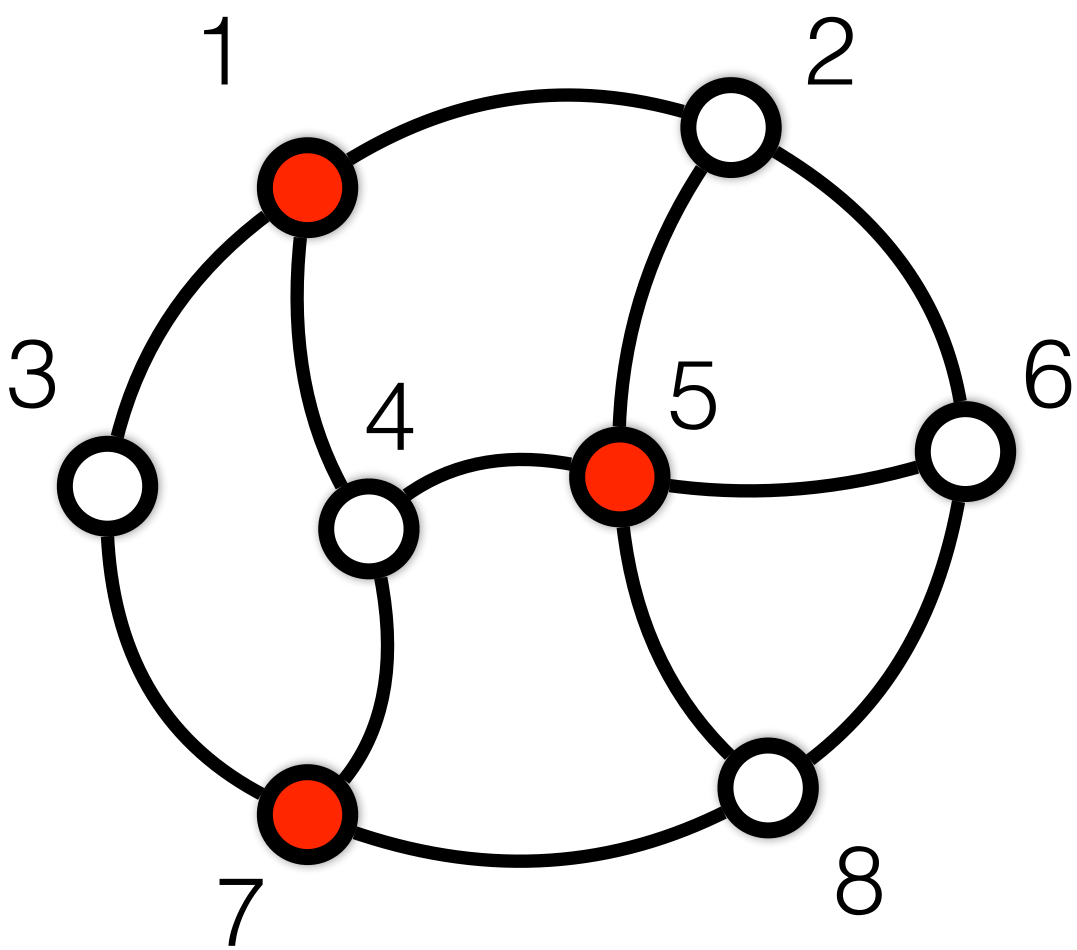
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1								
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0								
1	0	0	0	0	1	0	0								
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

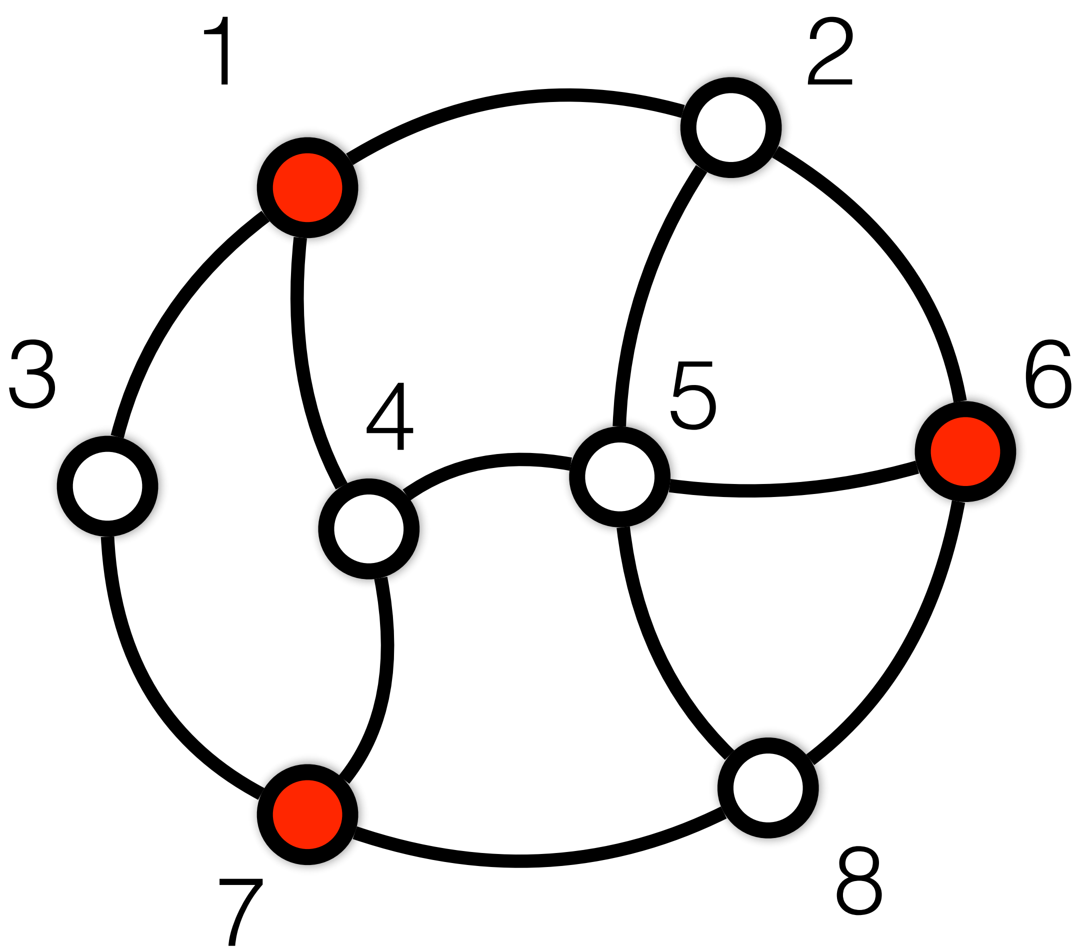
All Candidate Colorclasses



1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	0	1

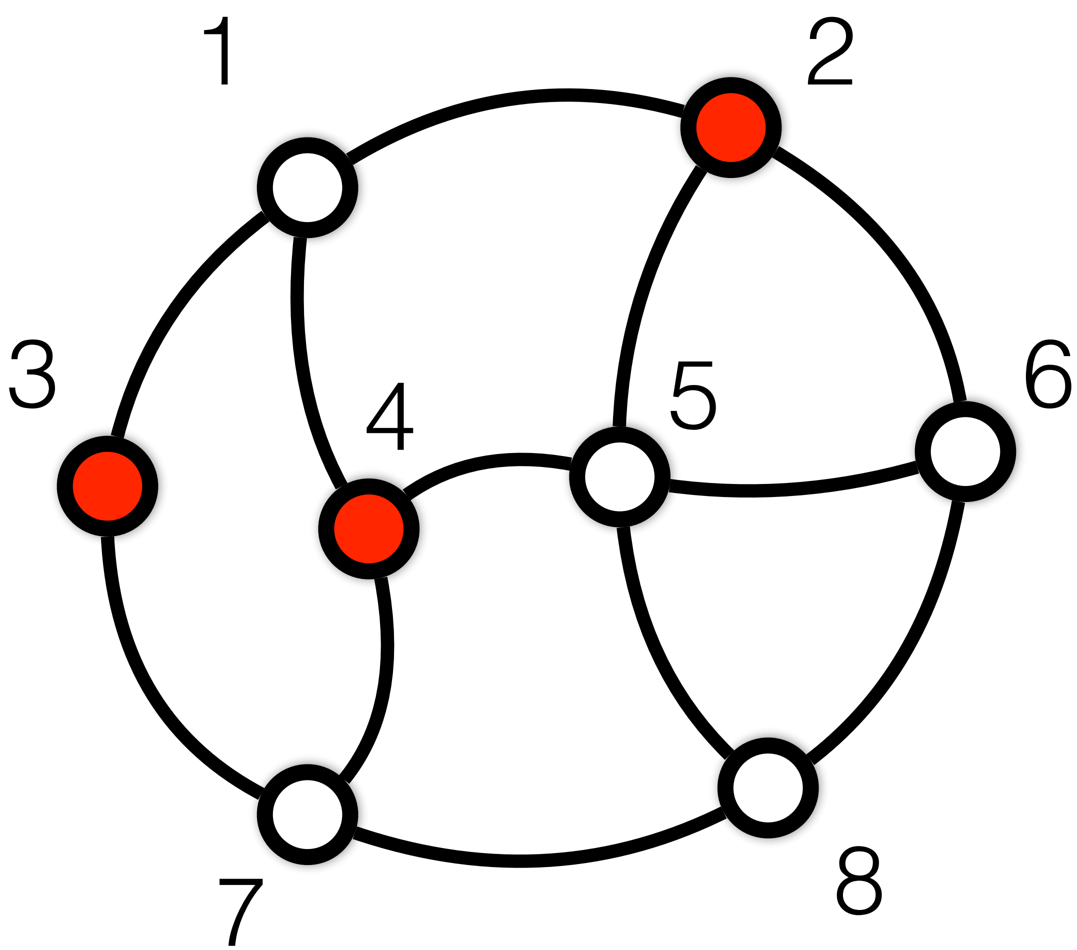
1	2	3	4	5	6	7	8
0	0	1	1	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	0
0	0	1	0	0	0	0	1
0	0	0	1	0	1	0	0
0	0	0	1	0	0	0	1
0	0	0	0	1	0	1	0
0	0	0	0	0	1	1	0
1	0	0	0	1	0	1	0

All Candidate Colorclasses



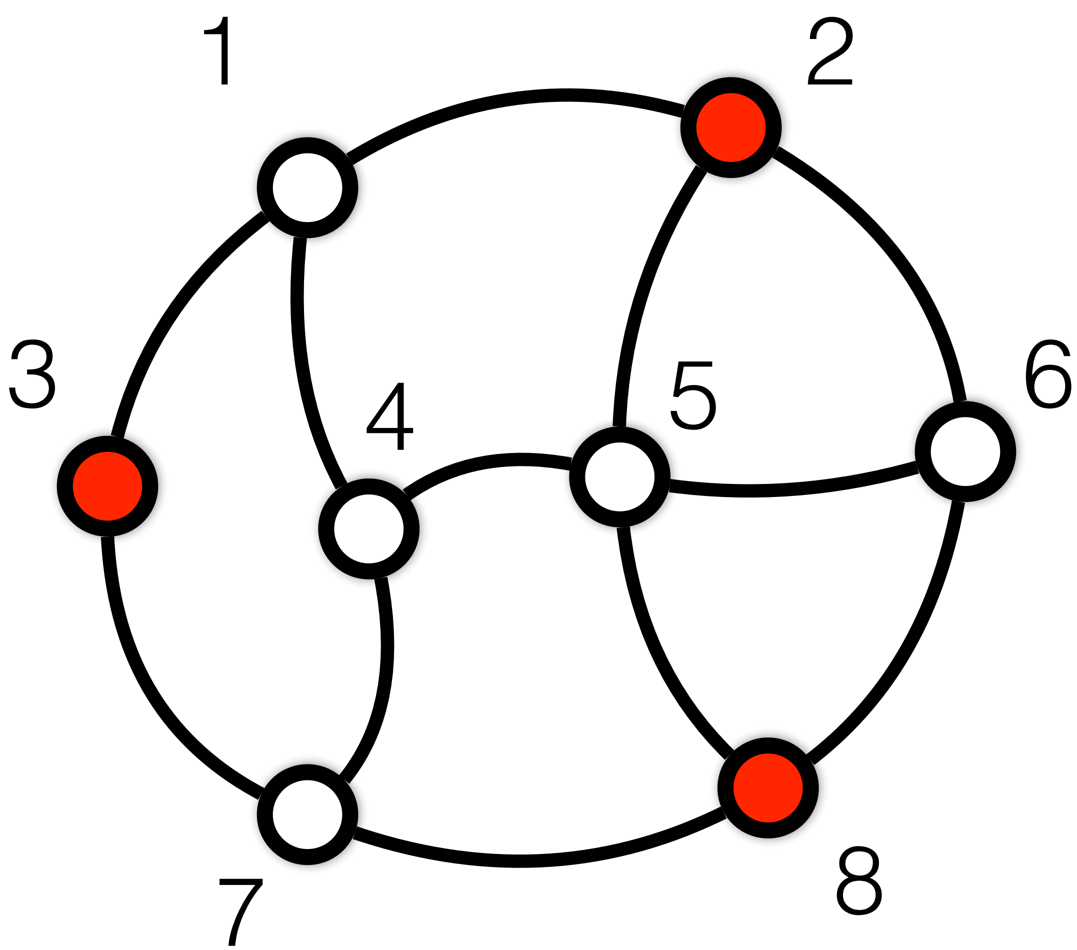
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0								
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



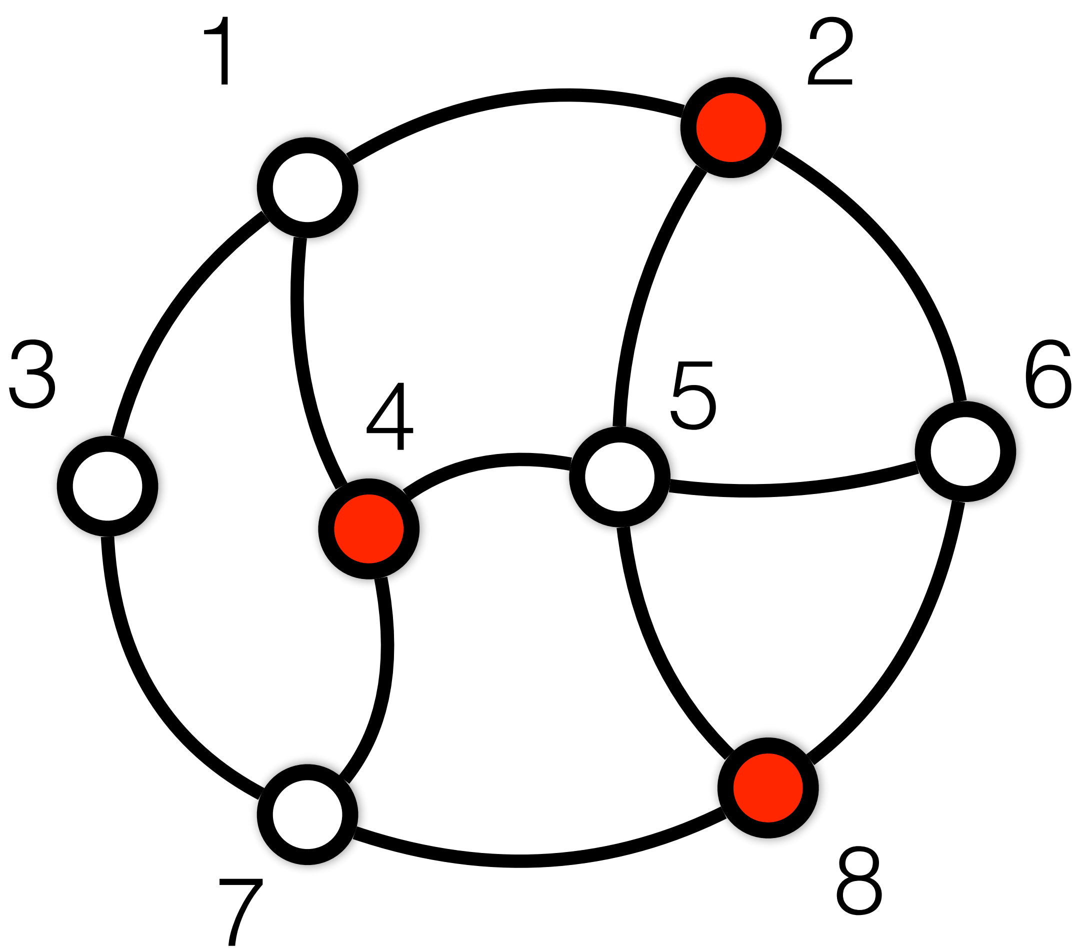
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1								
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



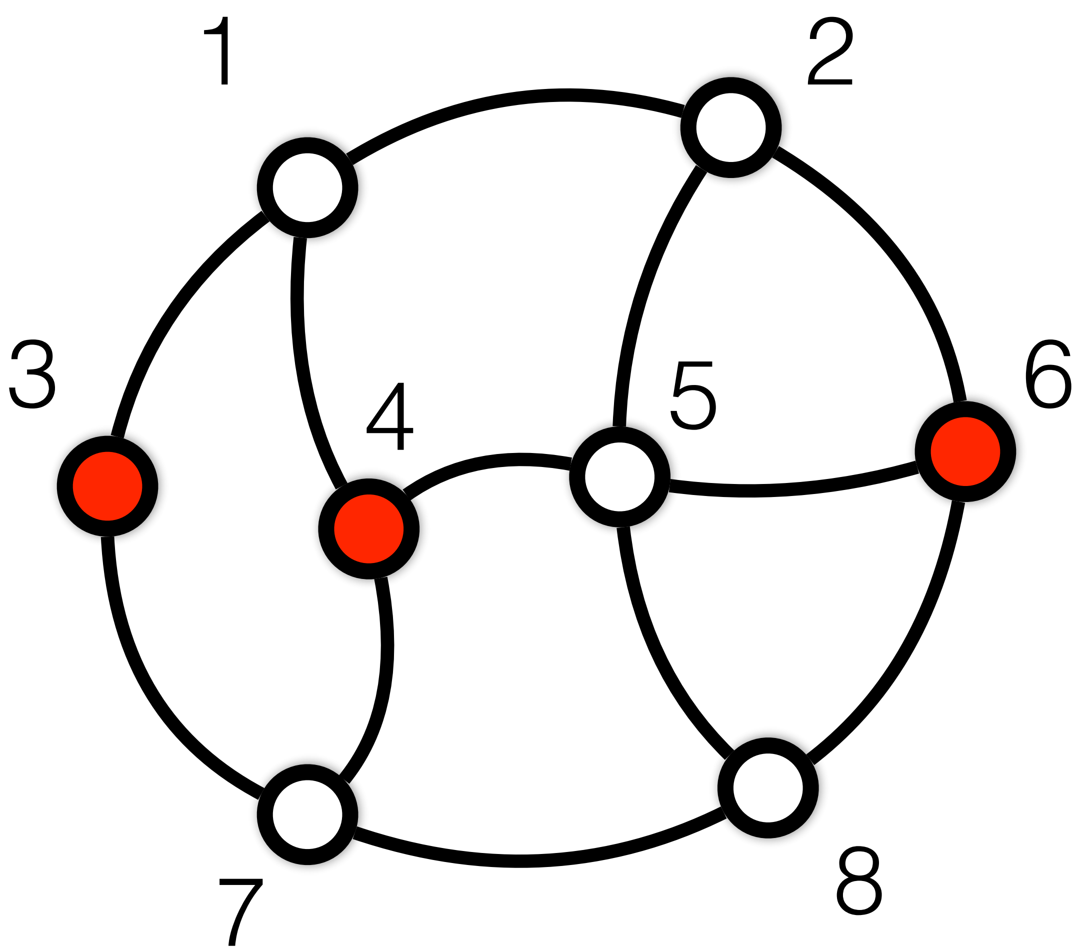
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	0	0	0								
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



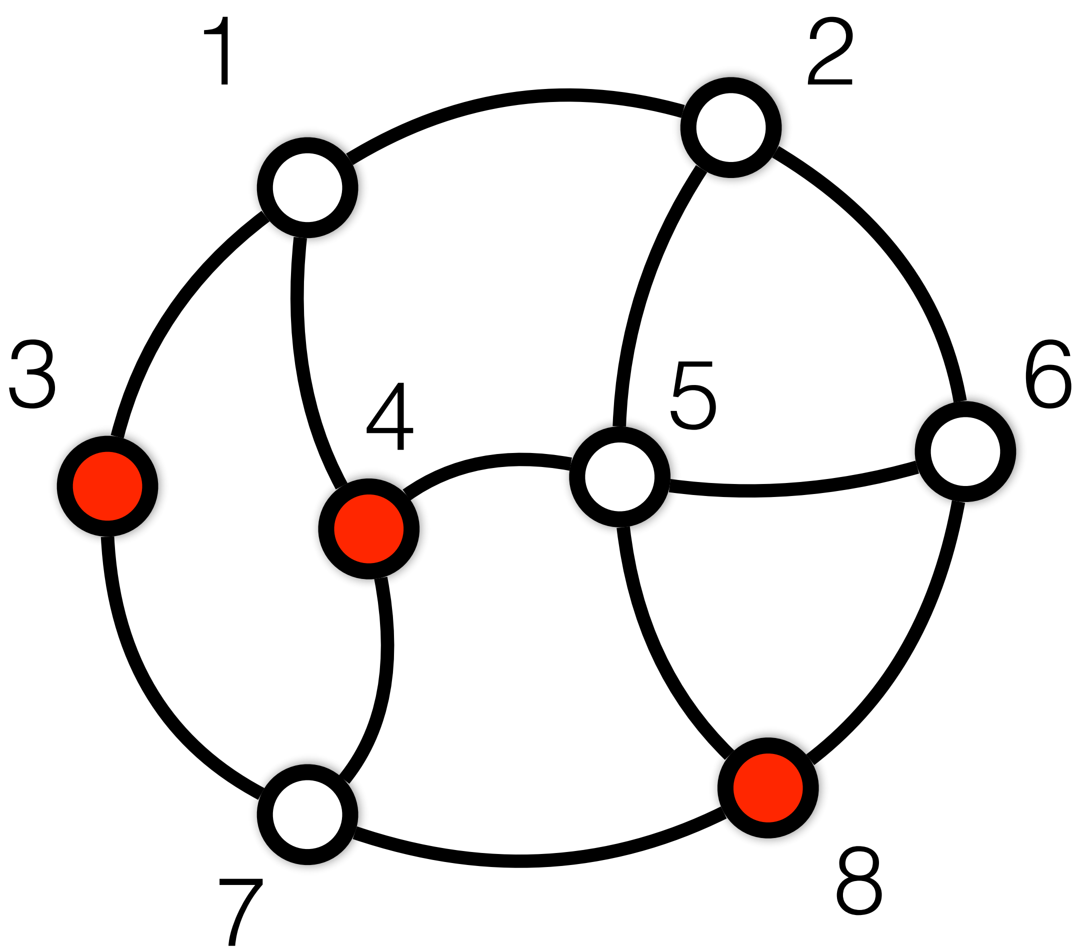
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	0	0	0								
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



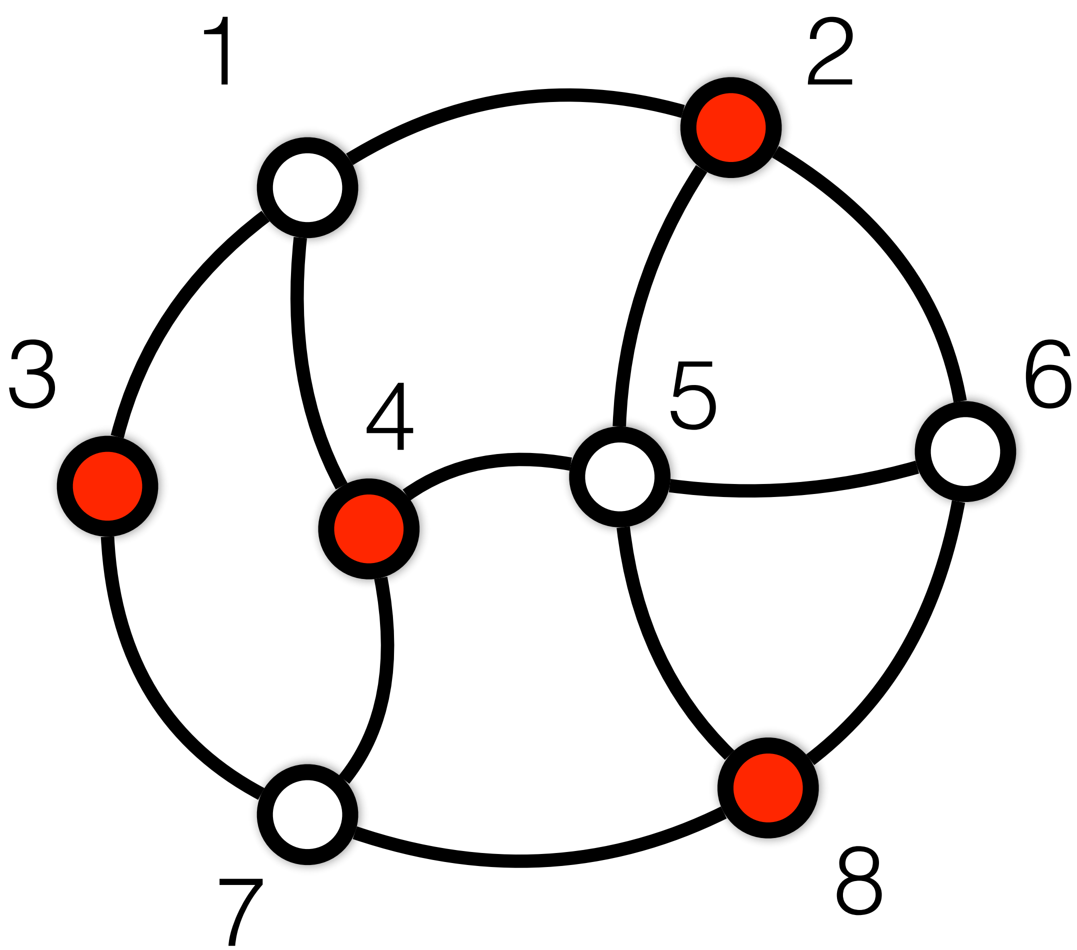
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0								
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



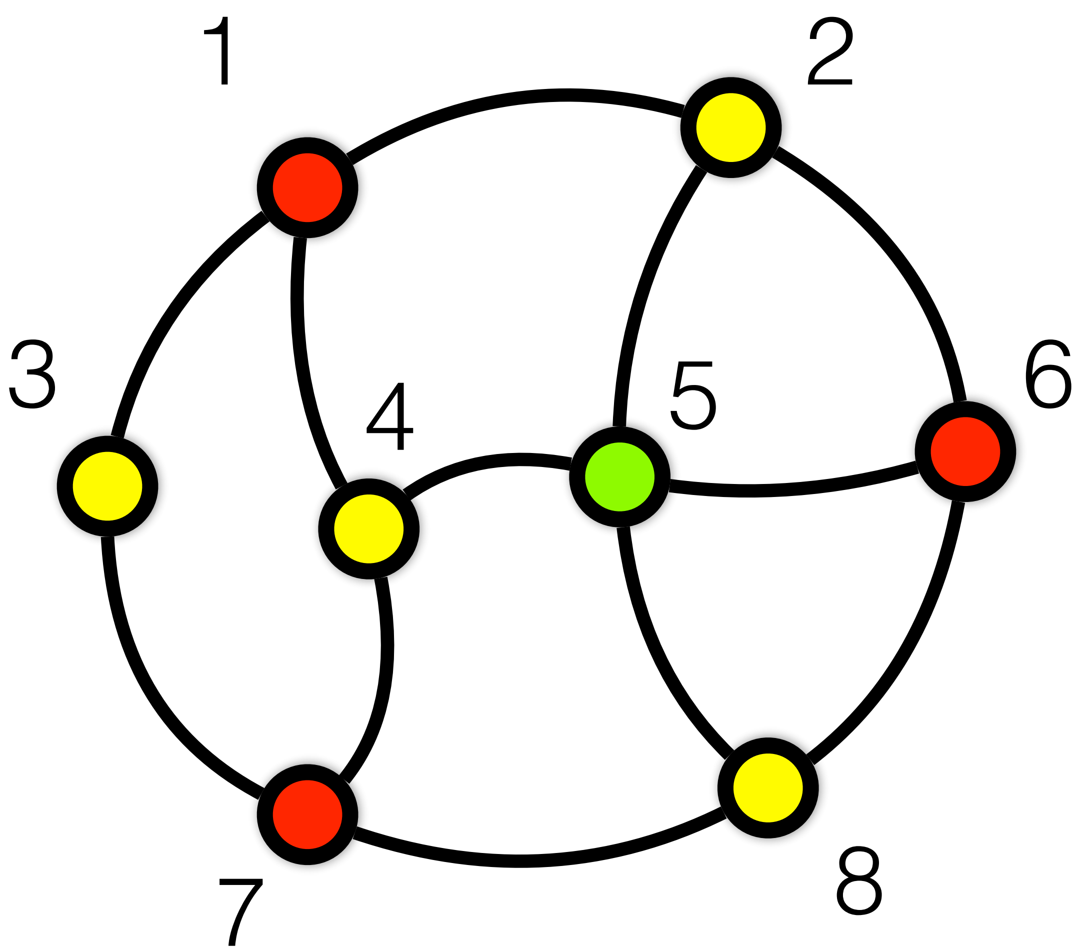
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1
0	1	0	0	0	0	0	1								

All Candidate Colorclasses



1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1
0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	1

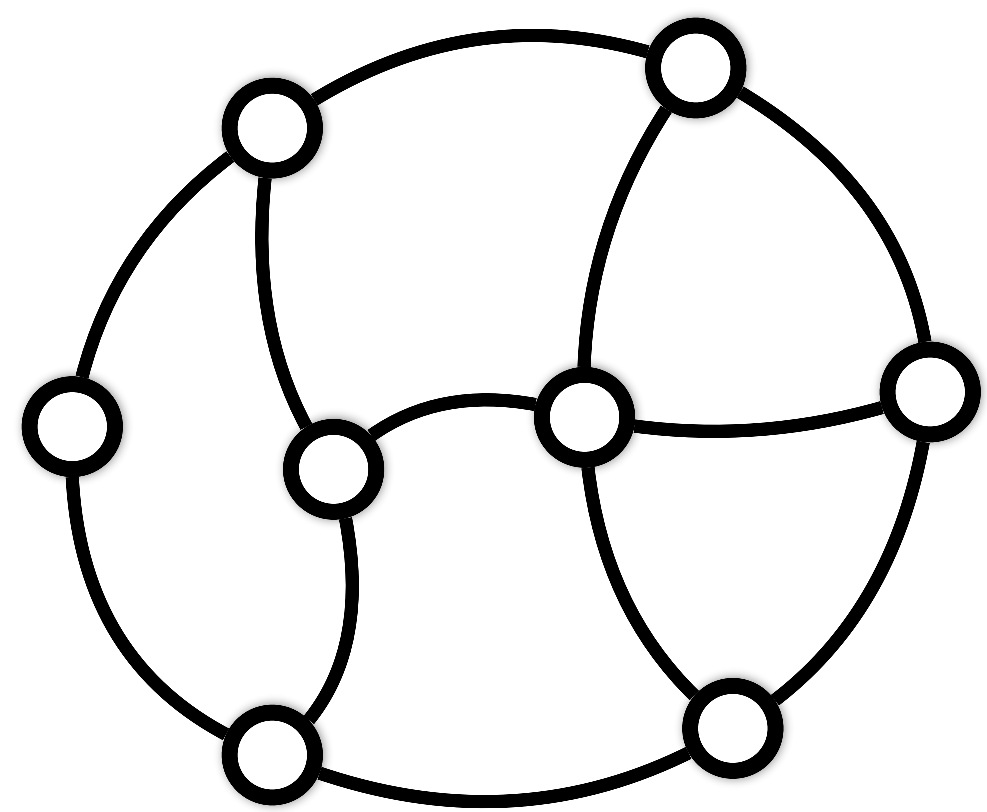
A Coloring is a Set of Disjoint Colorclasses



1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	**0**	**0**	**0**	**1**	**0**	**0**	**0**	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**
1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1
0	1	0	0	0	0	0	1	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**

Dynamic Programming across Vertex Subsets

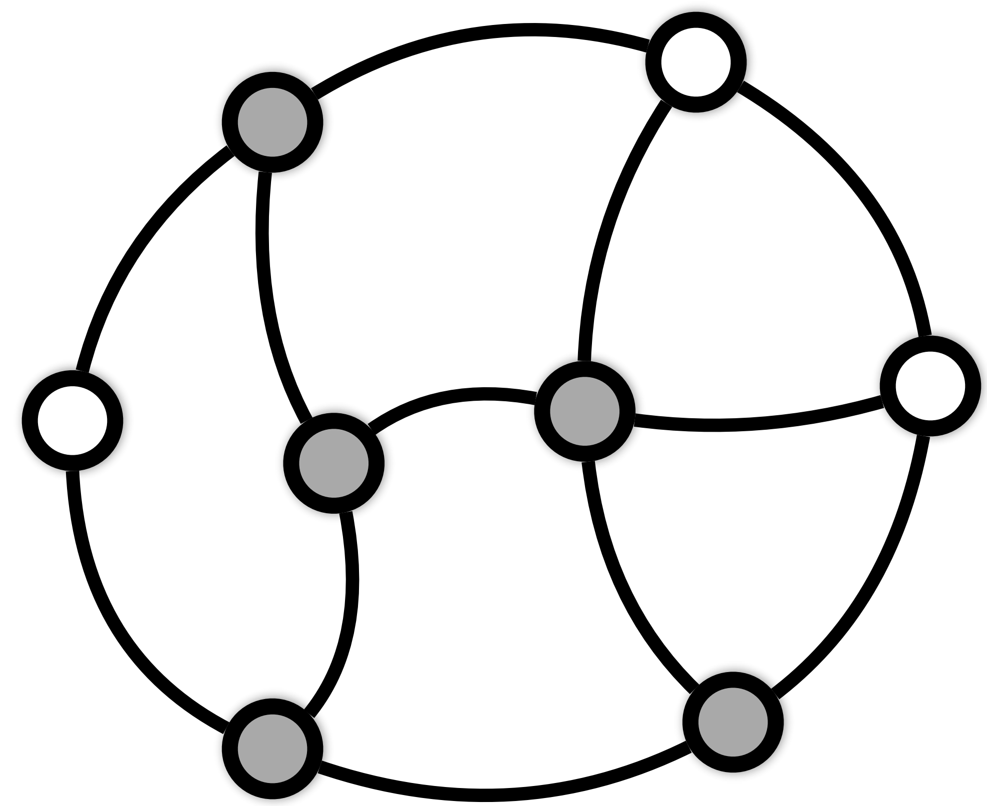
- For a vertex subset X, define d(X) as the smallest number of colors needed in a proper coloring of G[X], the graph *induced* by X.



The graph G.

Dynamic Programming across Vertex Subsets

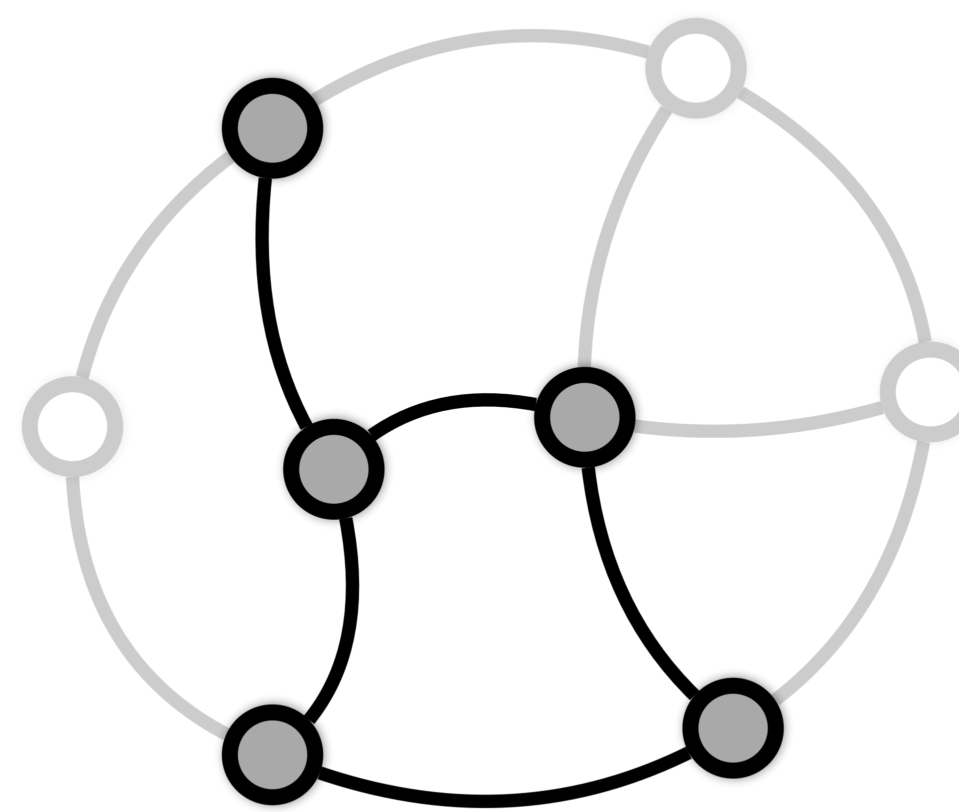
- For a vertex subset X, define d(X) as the smallest number of colors needed in a proper coloring of G[X], the graph *induced* by X.



X is the grey vertices.

Dynamic Programming across Vertex Subsets

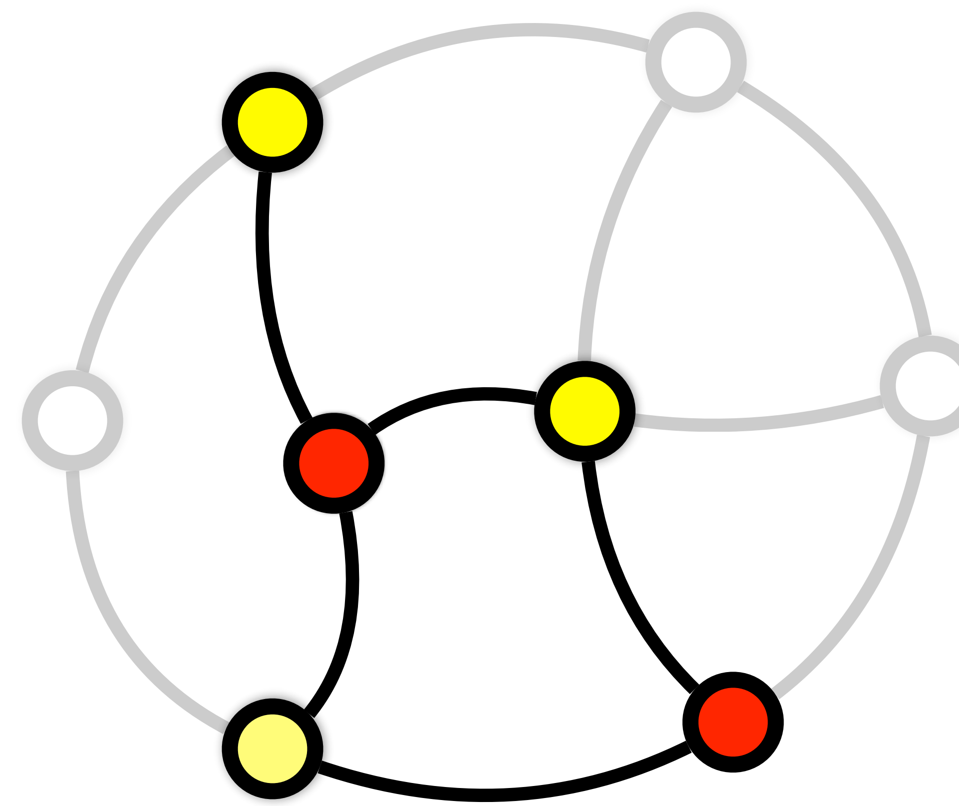
- For a vertex subset X, define d(X) as the smallest number of colors needed in a proper coloring of G[X], the graph *induced* by X.



The induced graph G[X] is
X and all edges between vertices
In X.

Dynamic Programming across Vertex Subsets

- For a vertex subset X, define d(X) as the smallest number of colors needed in a proper coloring of G[X], the graph *induced* by X.



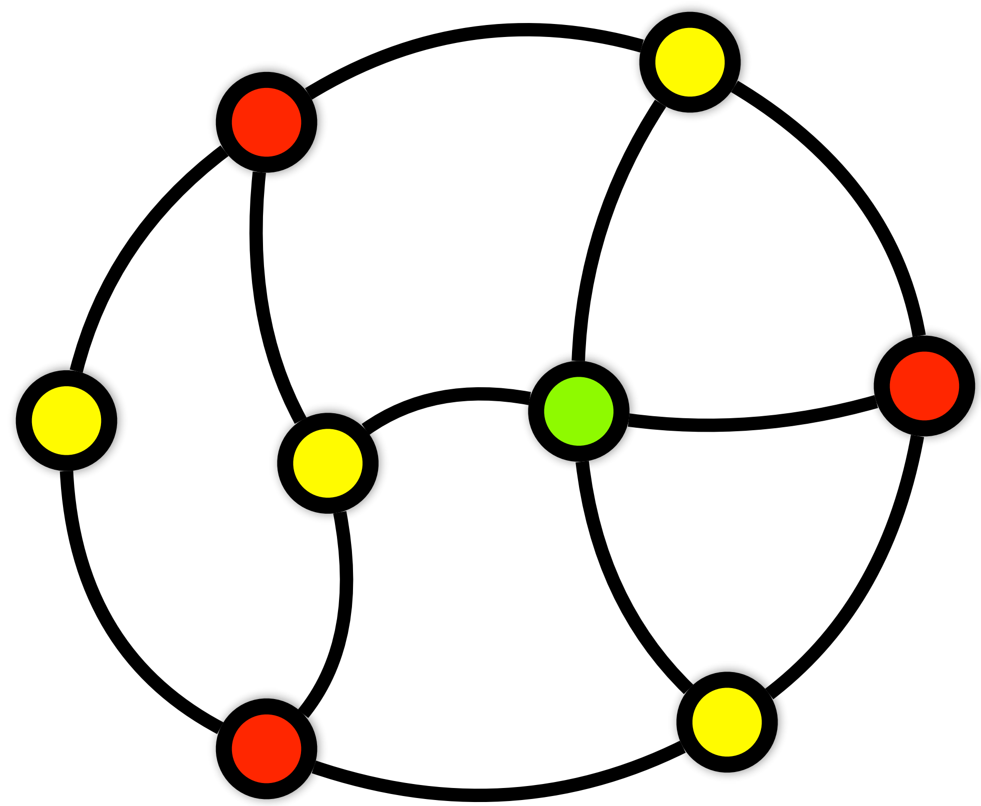
d(X)=2.

Dynamic Programming across Vertex Subsets

- d(0)=0.
- d(X)=min$_Y$ d(X-Y)+1.

  Y is a colorclass

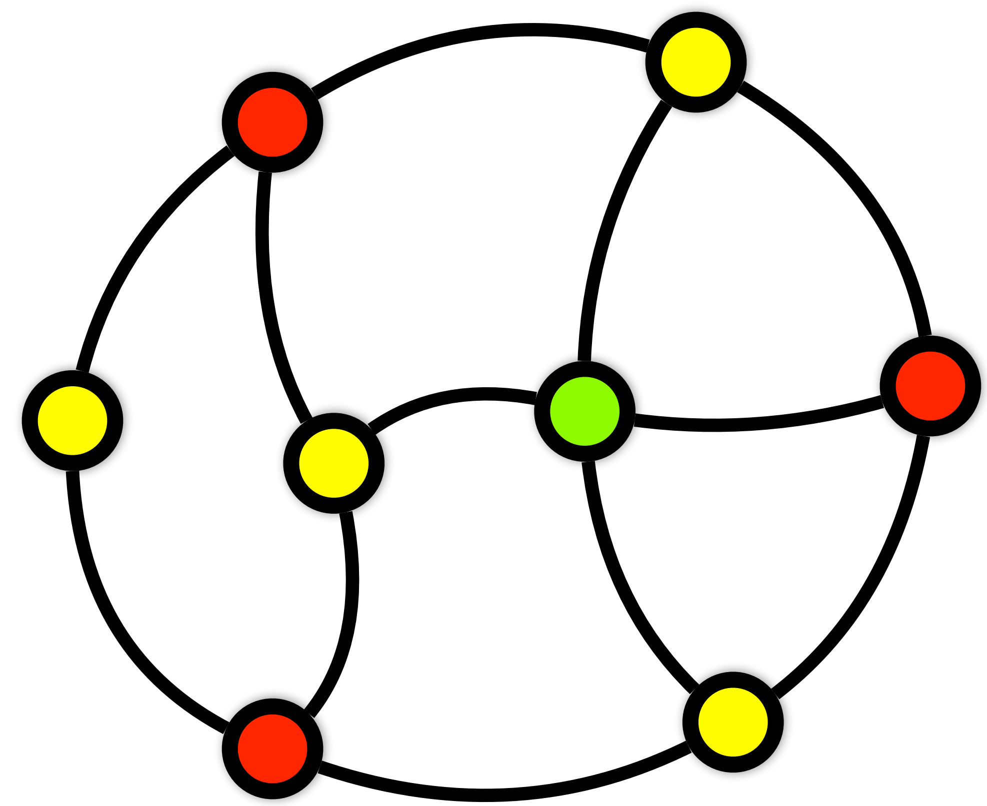
  candidate in G[X]



Dynamic Programming across Vertex Subsets

- Clearly,

  $d(V) = \chi(G)$.



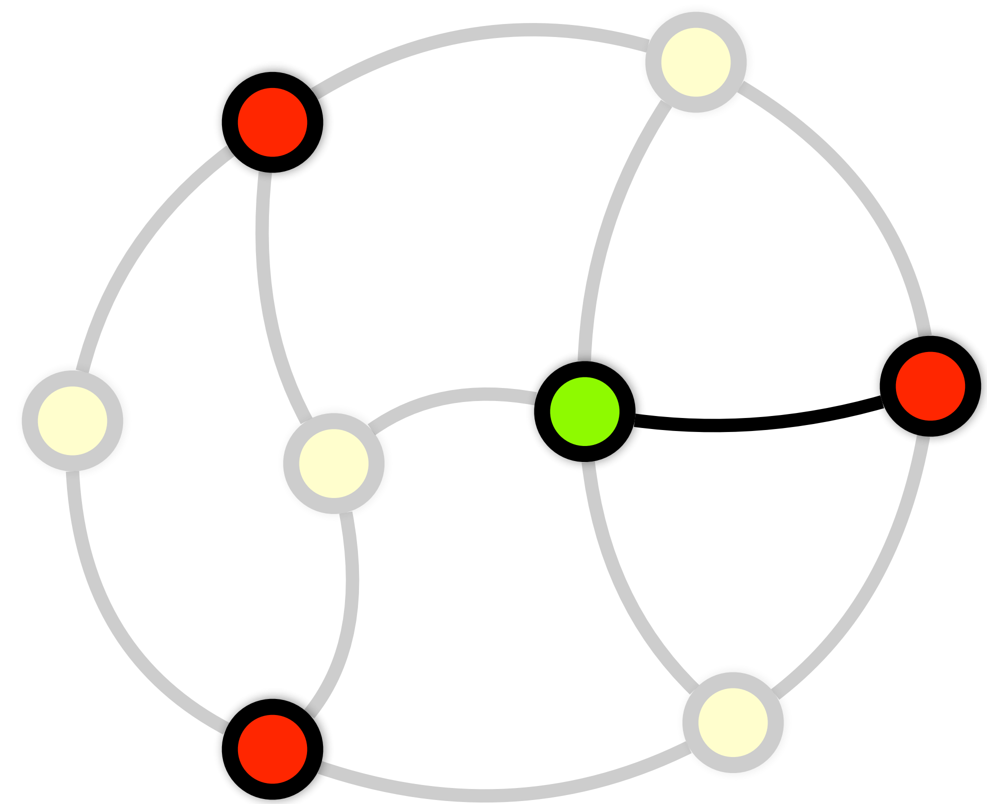
$d(V)=3$.

Dynamic Programming across Vertex Subsets

- d(0)=0.
- d(X)=min$_Y$ d(X-Y)+1.

  Y is a colorclass

  candidate in G[X]



X=V, Y=Yellow colorclass

Running Time Analysis

$$t(n) \leq \sum_{X \subseteq V} \sum_{Y \subseteq X} 1 = \sum_{X \subseteq V} 2^{|X|} = \sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n.$$

Overestimate of
candidate colorclasses.

Improved Bounds for the Dynamic Programming Approach

- [Lawler 1976] $O(2.44^n)$ by looping over maximal independent sets as colorclasses.
- [Byskov 2003] $O(2.40^n)$ by more careful analysis.

2^npoly(n) time algorithm for chromatic number

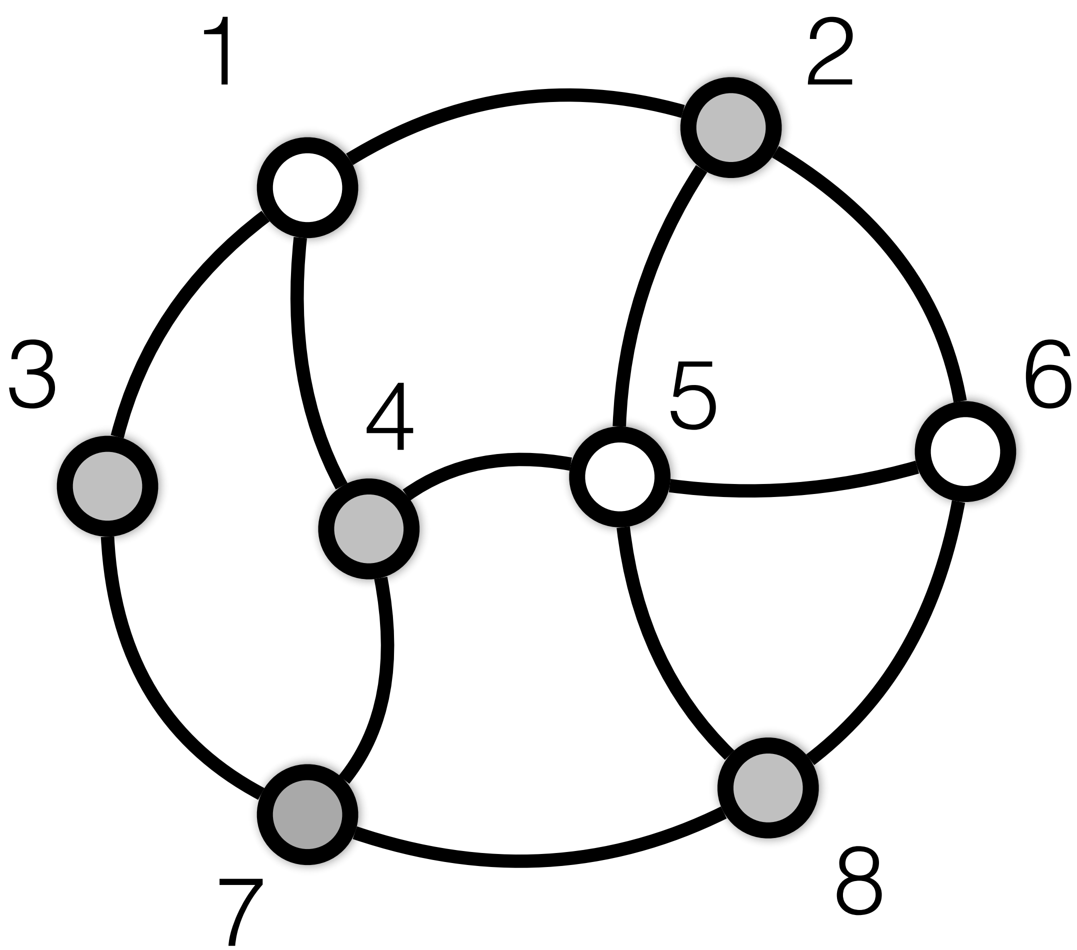
[B., Husfeldt, and Koivisto 2006]

- Use inclusion-exclusion summation to count k-colorings.
- Use the fast zeta transform (a variant of the Fast Fourier transform) to efficiently count candidate colorclasses in every induced subgraph G[X] at once.

Subset Induced Colorclasses

- Let $a(X)$ for X a subset of the vertices V be the number of candidate color classes in the induced graph $G[X]$.

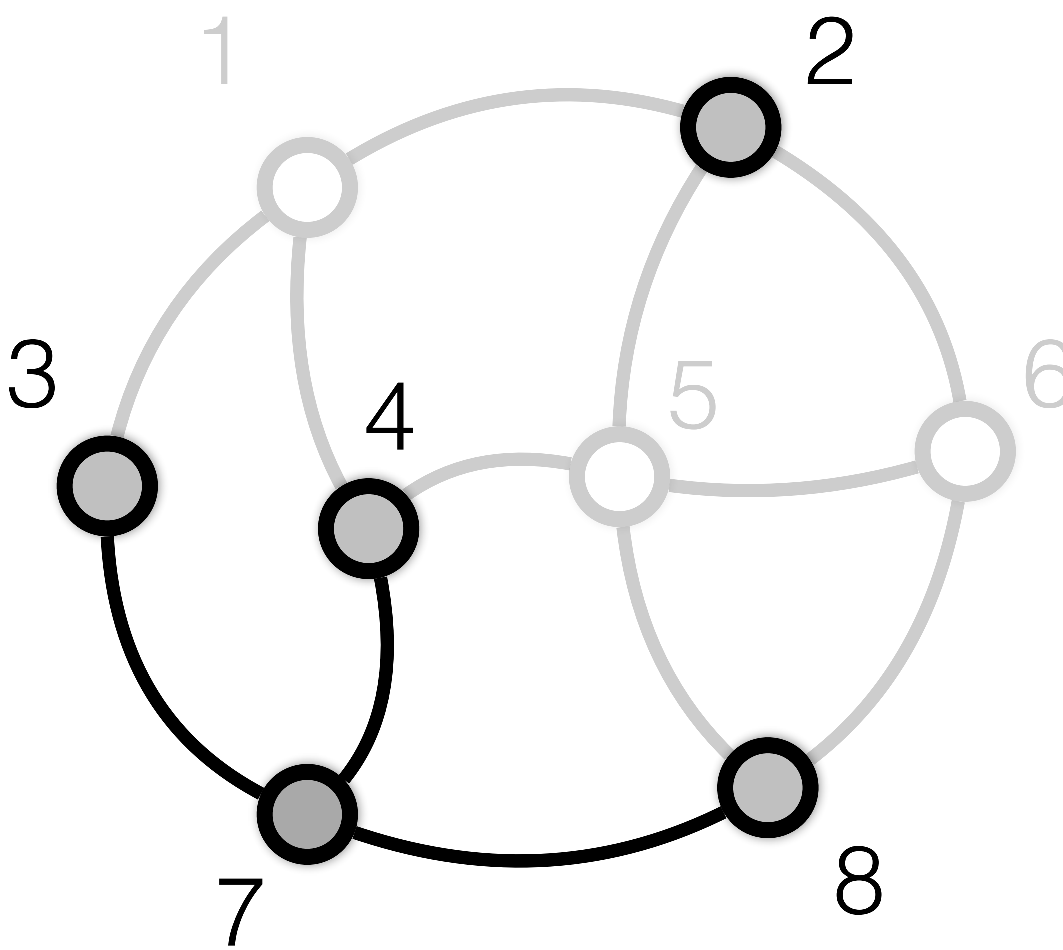
Subset Induced Colorclasses



Subset X

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1
0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	1

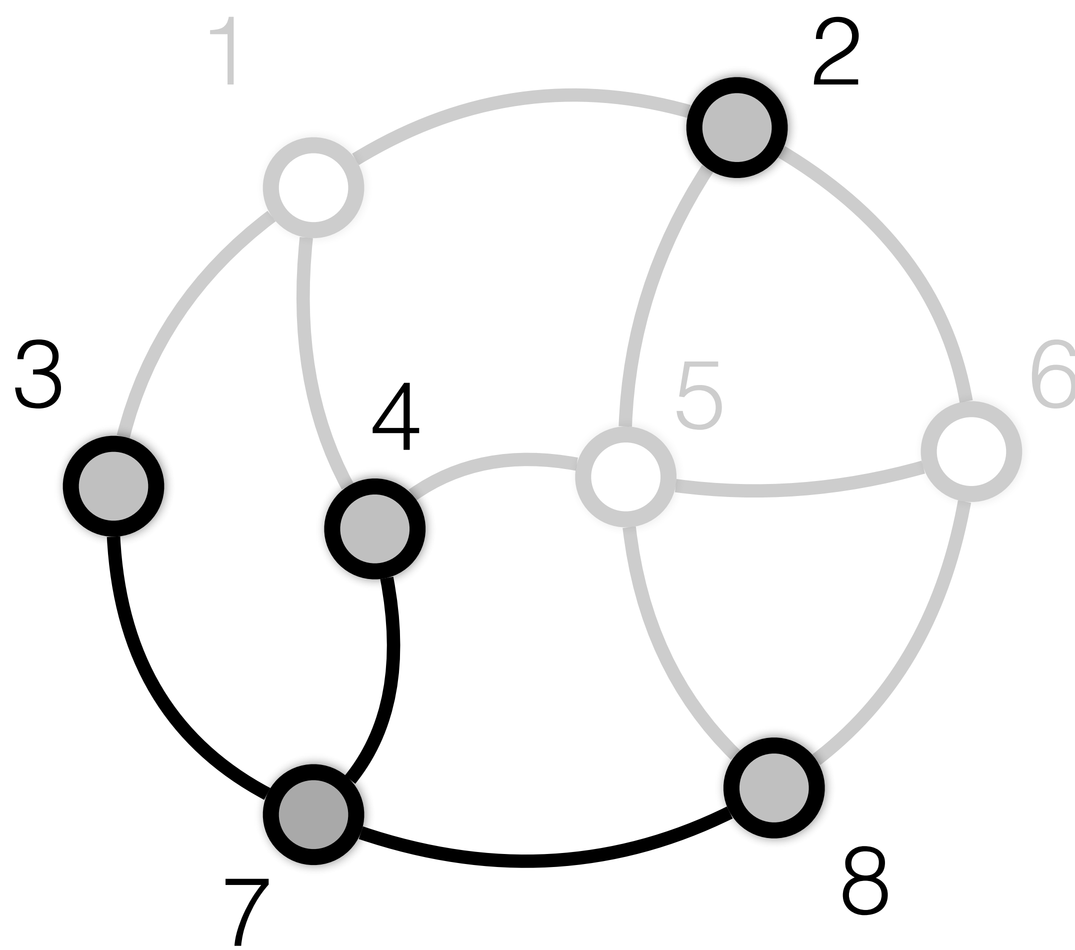
Subset Induced Colorclasses



G[X]

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1
0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	1

Subset Induced Colorclasses

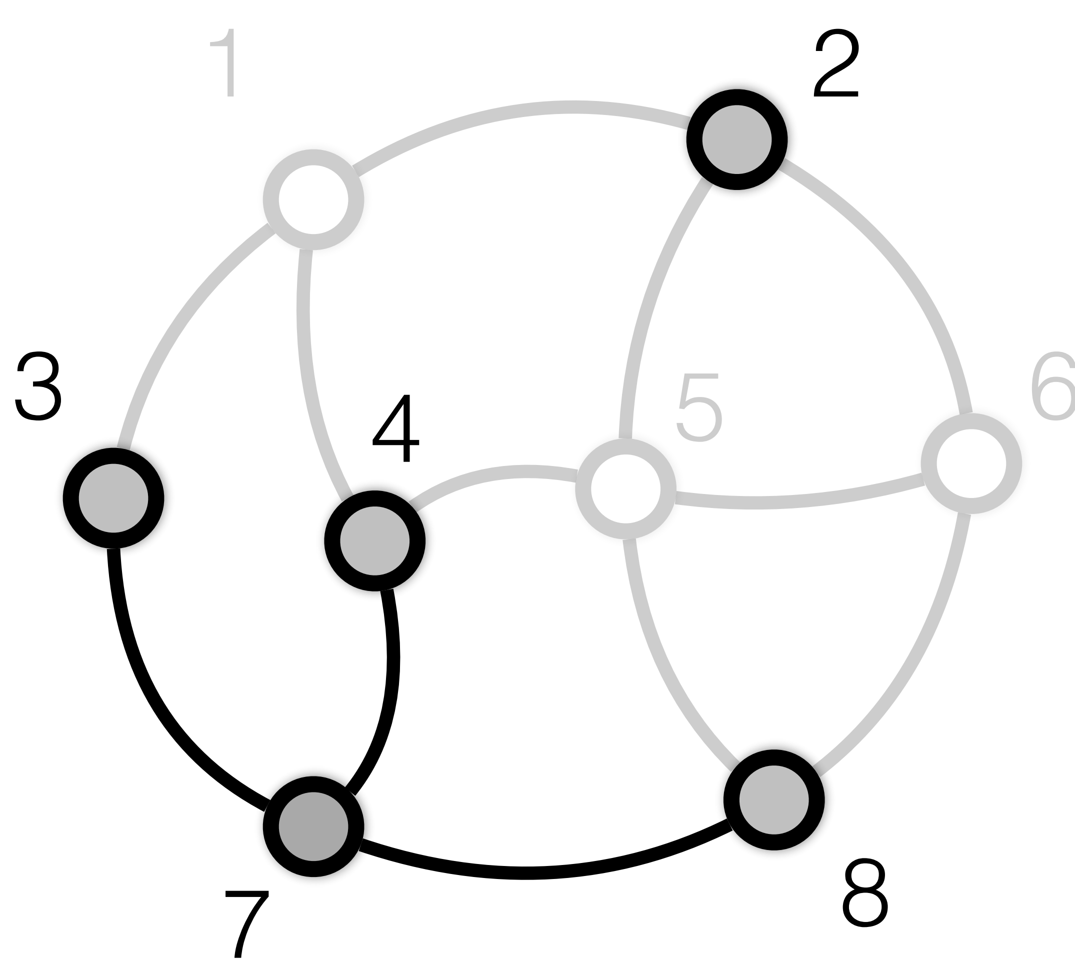


G[X]

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**
0	**1**	**0**	**0**	**0**	**0**	**0**	**0**	0	0	1	0	1	0	0	0
0	**0**	**1**	**0**	**0**	**0**	**0**	**0**	0	0	1	0	0	1	0	0
0	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**
0	**0**	**0**	**0**	**0**	**0**	**1**	**0**	0	0	0	0	1	0	1	0
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**
1	0	0	0	0	0	0	1	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**
0	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**
0	**1**	**0**	**1**	**0**	**0**	**0**	**0**	0	0	1	1	0	1	0	0
0	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**1**
0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**

Subset Induced Colorclasses

a(X)=17



G[X]

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**
0	**1**	**0**	**0**	**0**	**0**	**0**	**0**	0	0	1	0	1	0	0	0
0	**0**	**1**	**0**	**0**	**0**	**0**	**0**	0	0	1	0	0	1	0	0
0	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**
0	**0**	**0**	**0**	**0**	**0**	**1**	**0**	0	0	0	0	1	0	1	0
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**
1	0	0	0	0	0	0	1	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**
0	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**
0	**1**	**0**	**1**	**0**	**0**	**0**	**0**	0	0	1	1	0	1	0	0
0	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**1**
0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**

Inclusion-Exclusion

$$p(k) = \sum_{X \subseteq V} (-1)^{|V-X|} a(X)^k.$$

- p(k) is zero if there is no k-coloring,
- p(k) is non-zero if there are k-colorings.

Meaning of Powers of a(X)

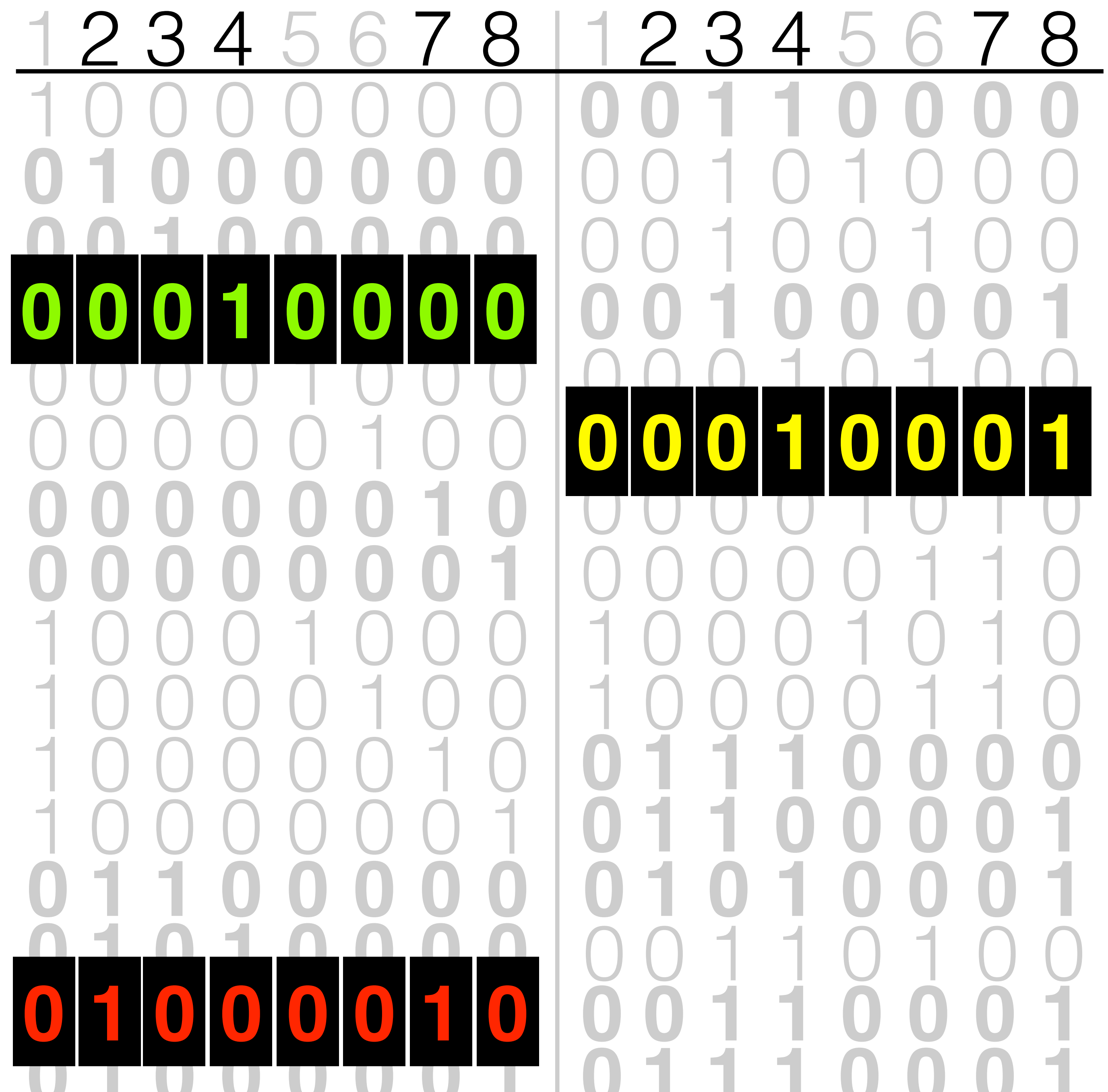
- $a(X)^k$ counts the number of ways to pick k color classes (with repetition) in G[X].

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**
0	**1**	**0**	**0**	**0**	**0**	**0**	**0**	0	0	1	0	1	0	0	0
0	**0**	**1**	**0**	**0**	**0**	**0**	**0**	0	0	1	0	0	1	0	0
0	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**
0	**0**	**0**	**0**	**0**	**0**	**1**	**0**	0	0	0	0	1	0	1	0
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**
1	0	0	0	0	0	0	1	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**
0	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**
0	**1**	**0**	**1**	**0**	**0**	**0**	**0**	0	0	1	1	0	1	0	0
0	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**1**
0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**

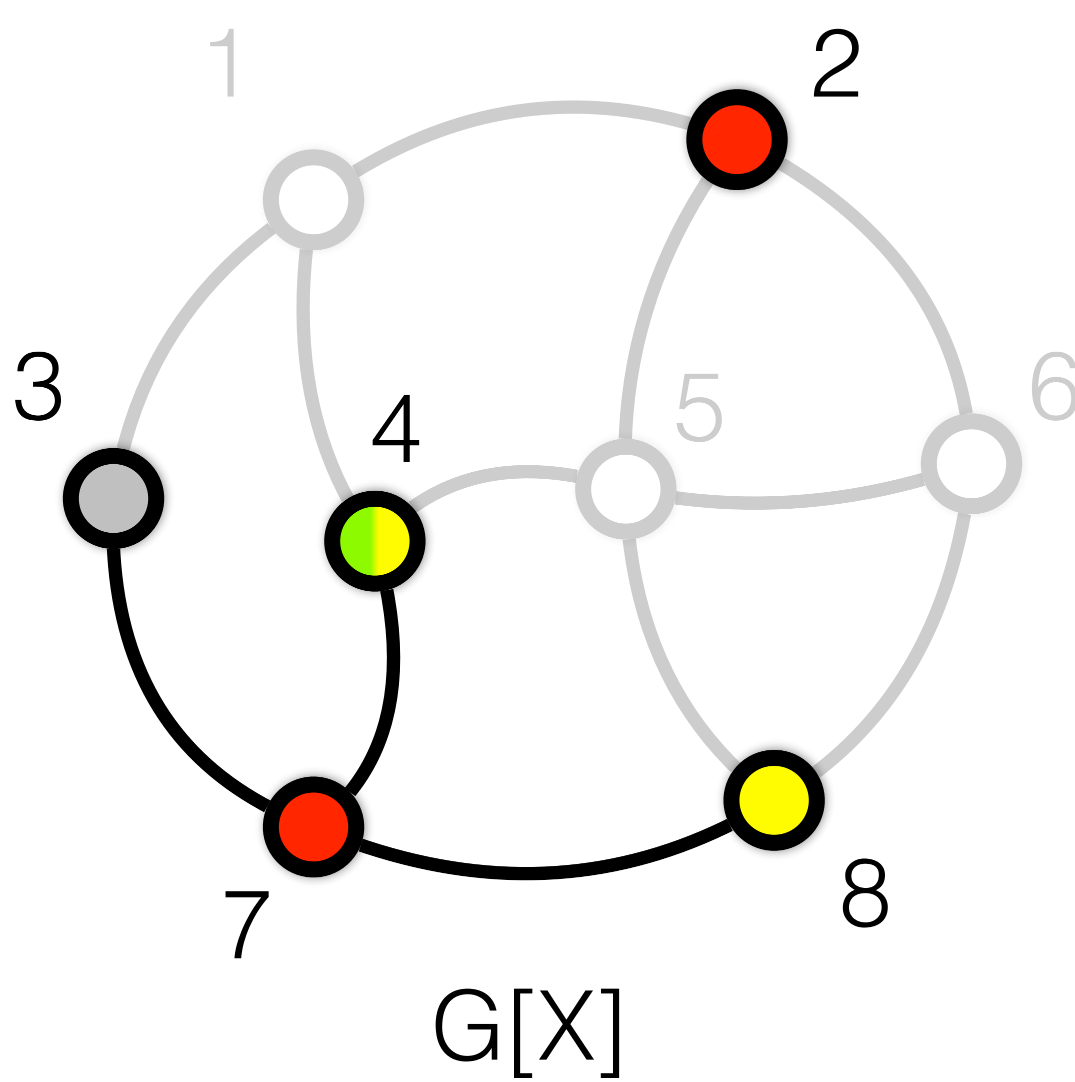
Meaning of Powers of a(X)

k=3

- $a(X)^k$ counts the number of ways to pick k color classes (with repetition) in G[X].

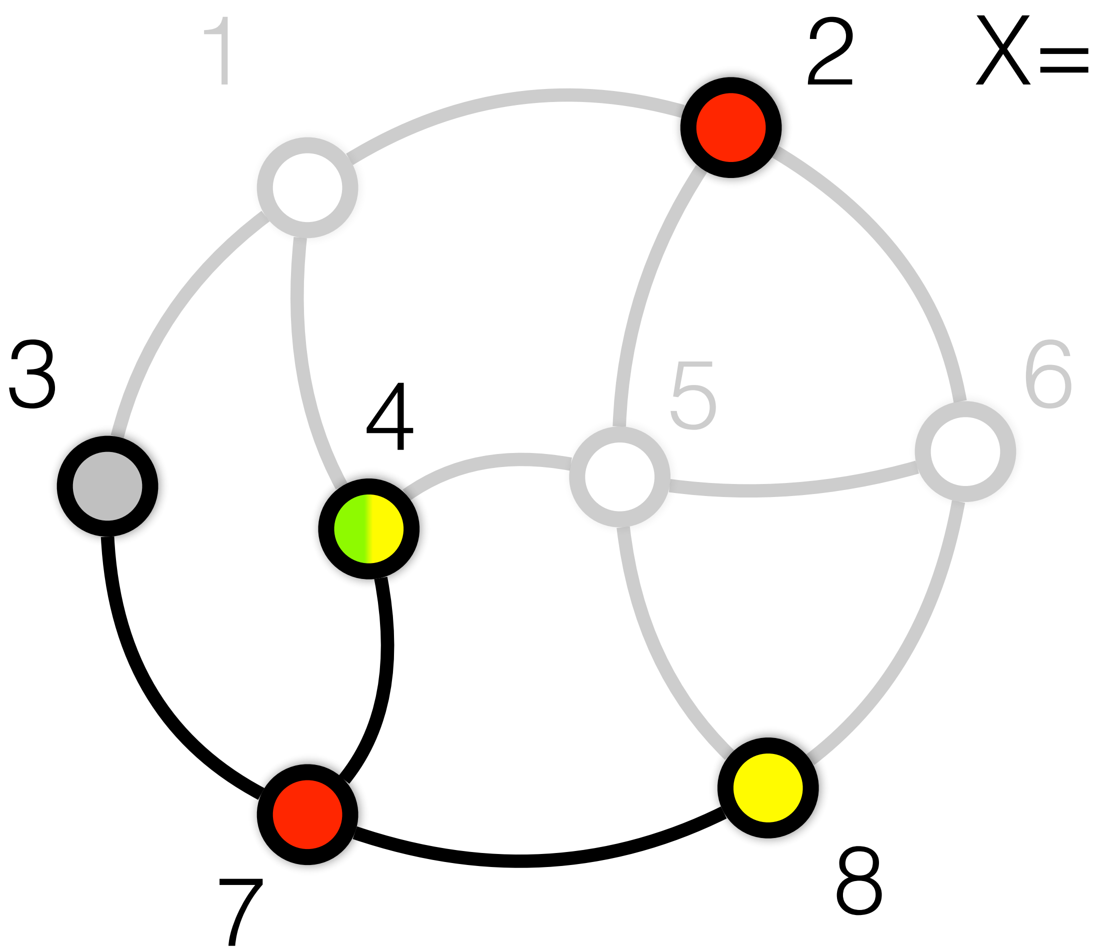


Meaning of Powers of a(X)



1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1
0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	1

A k-tuple of Colorclasses will be Counted in Many G[X]'s



G[X]

X= 1 2 3 4 5 6 7 8

0 0 0 1 0 0 0 0

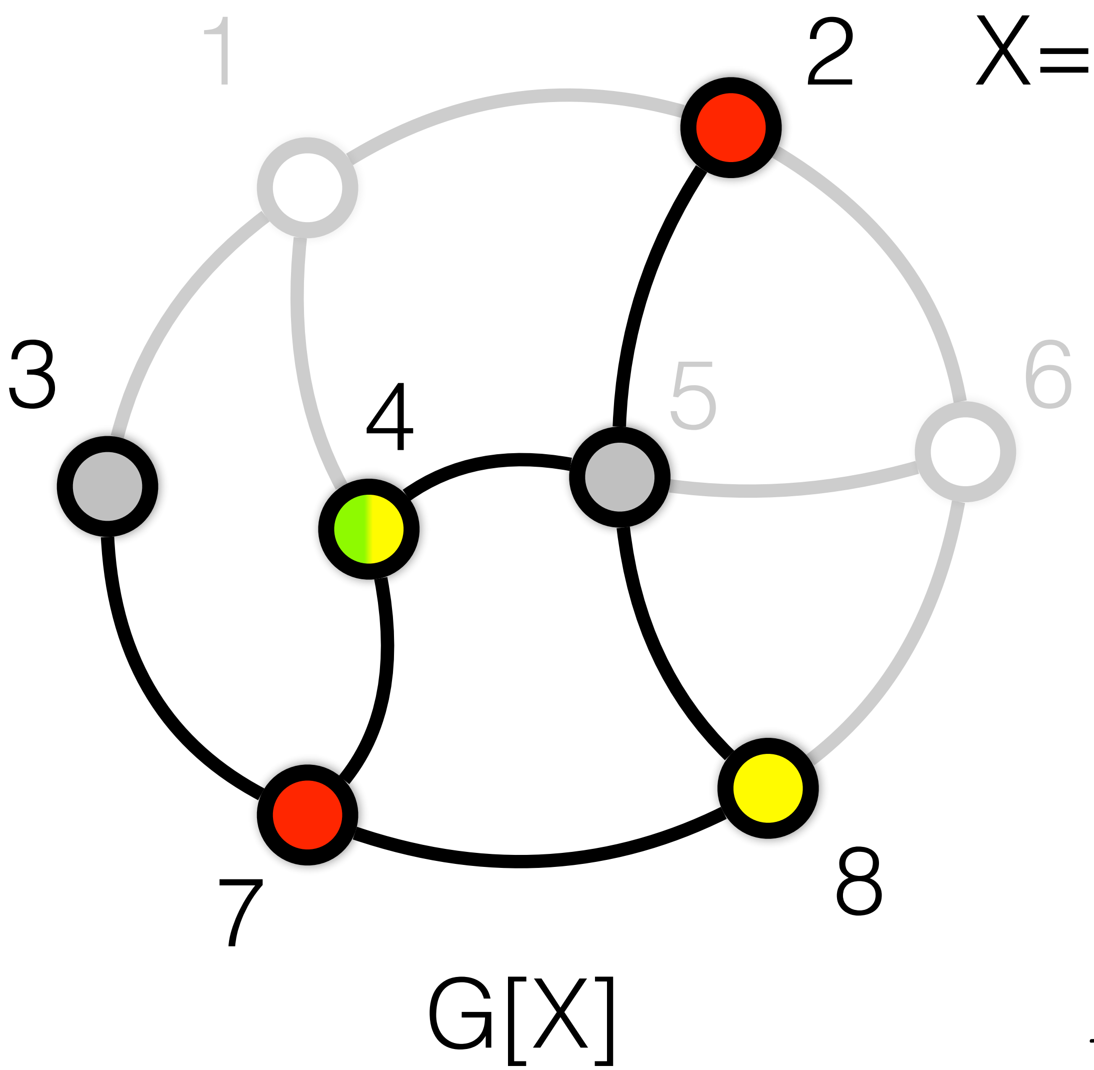
0 1 0 0 0 0 1 0

0 0 0 1 0 0 0 1

Sign=$(-1)^{8-3}$=-1

$$p(k) = \sum_{X \subseteq V} (-1)^{|V-X|} a(X)^k.$$

A k-tuple of Colorclasses will be Counted in Many G[X]'s



X= 1 2 3 4 5 6 7 8

0 0 0 1 0 0 0 0

0 1 0 0 0 0 1 0

0 0 0 1 0 0 0 1

Sign=$(-1)^{8-4}$=+1

$$p(k) = \sum_{X \subseteq V} (-1)^{|V-X|} a(X)^k.$$

K-tuples of Colorclasses

$$p(k) = \sum_{X \subseteq V} (-1)^{|V-X|} a(X)^k.$$

- Will be counted $2^{\text{\#of uncolored vertices}}$ times, but equally many times with sign factor -1 as +1. Hence, they will cancel each other in the sum unless all vertices are colored.

Fast Zeta Transform

$$a(X) = \sum_{Y \subseteq X} [Y \text{ is a candidate colorclass}].$$

- [Yates 1937] A table containing a(X) for all subsets X of V can be computed in $O(n2^n)$ time.

BHK'06 χ(G)-Algorithm

- Compute by Yates's algorithm

$$a(X) = \sum_{Y \subseteq X} [Y \text{ is a candidate colorclass}].$$

2^npoly(n) time.

- For k=1:n, evaluate

$$p(k) = \sum_{X \subseteq V} (-1)^{|V-X|} a(X)^k.$$

2^npoly(n) time.

until p(k)≠0, then return k.