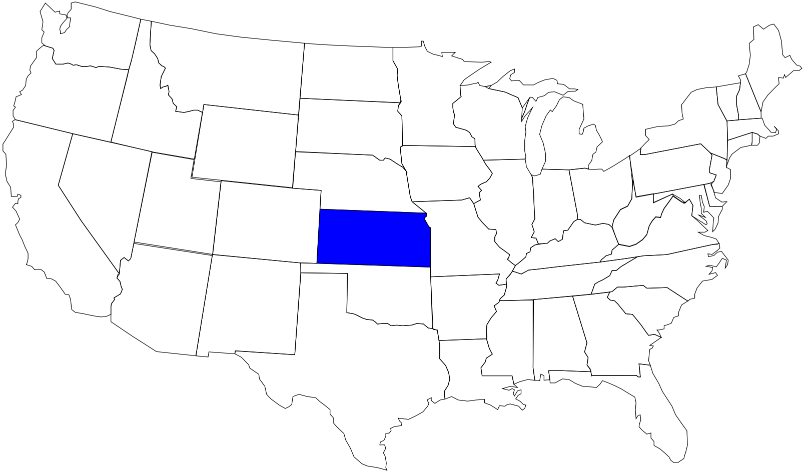
Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.

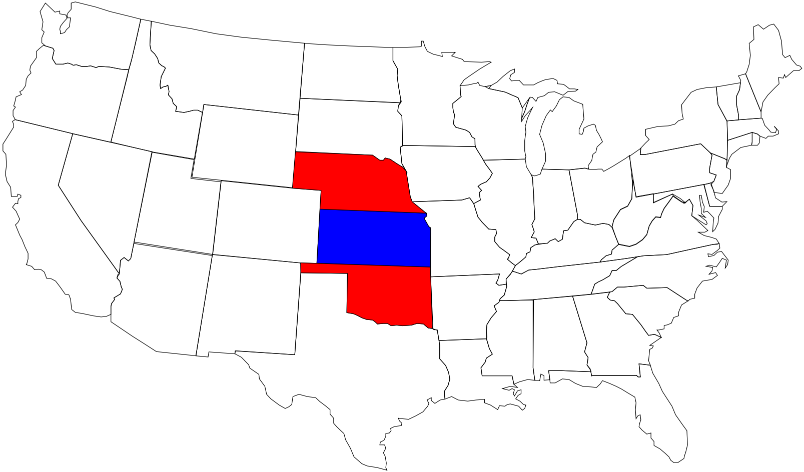
Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.



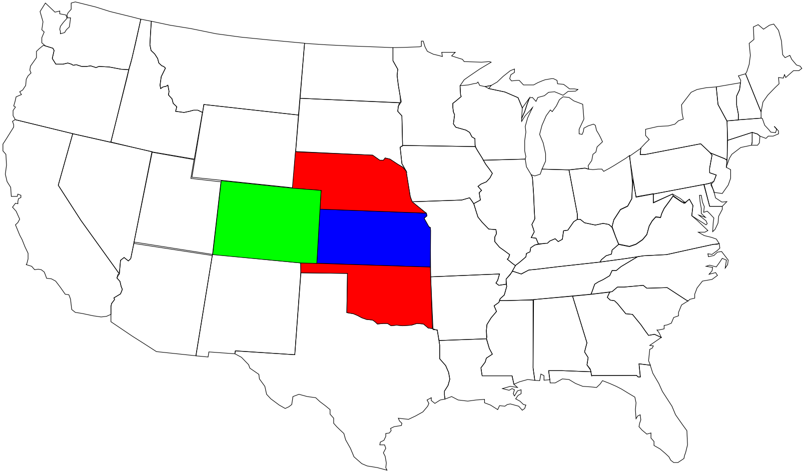
Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.



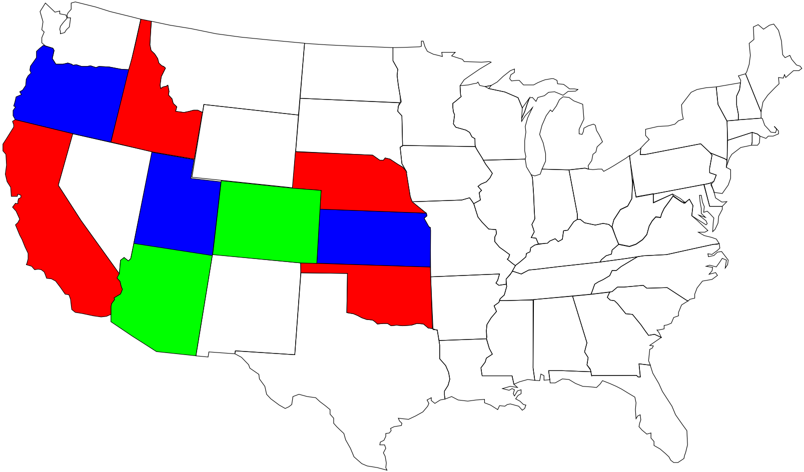
Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.



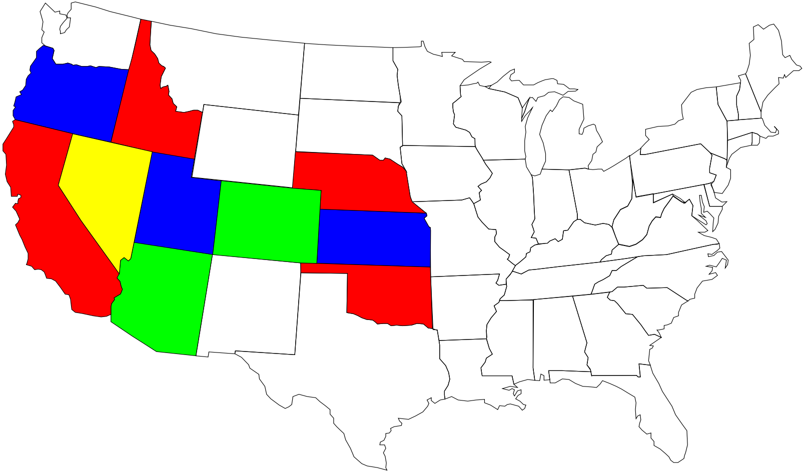
Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.



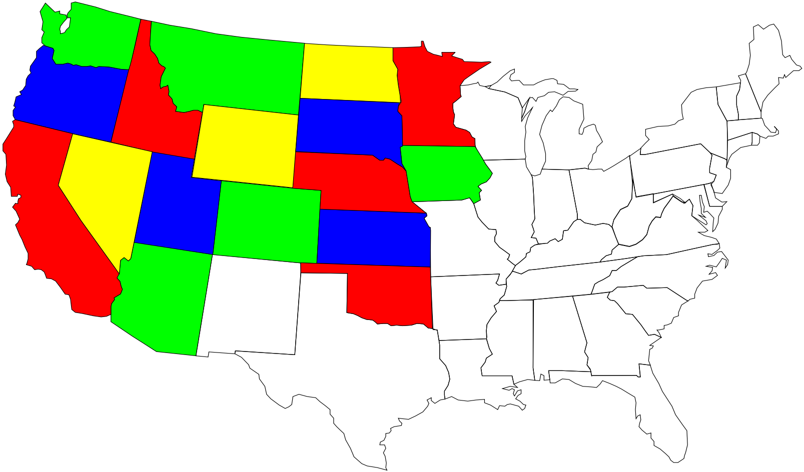
Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.



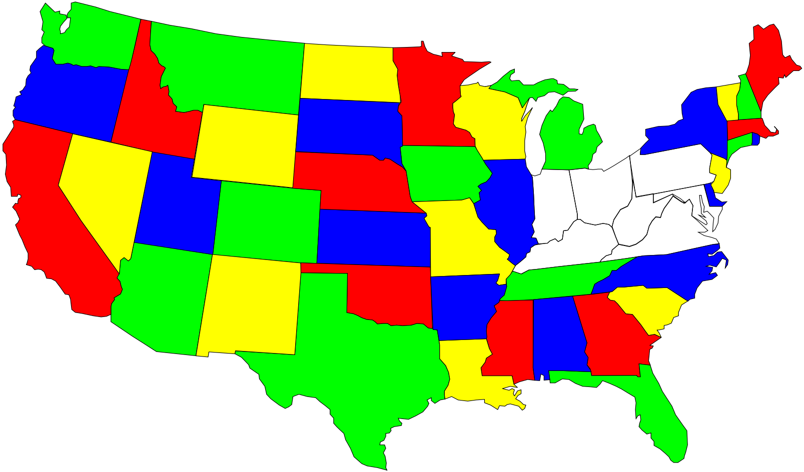
Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.



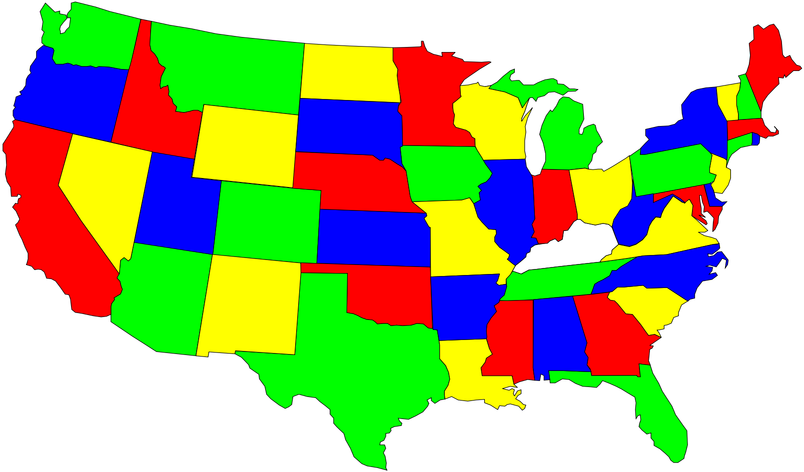
Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.



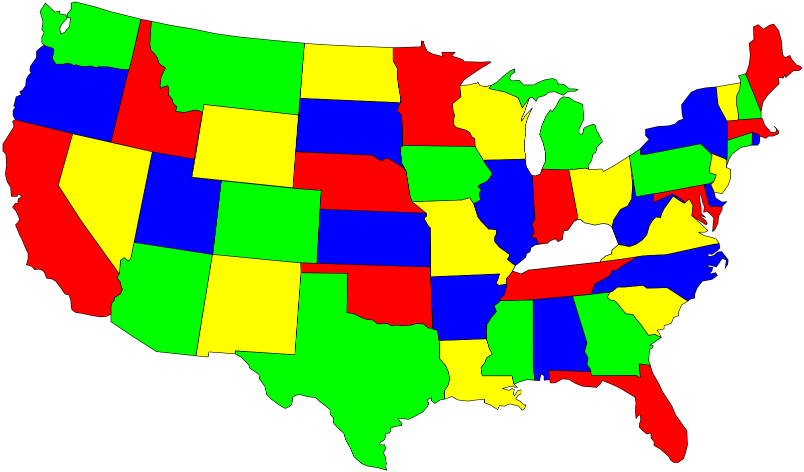
Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.



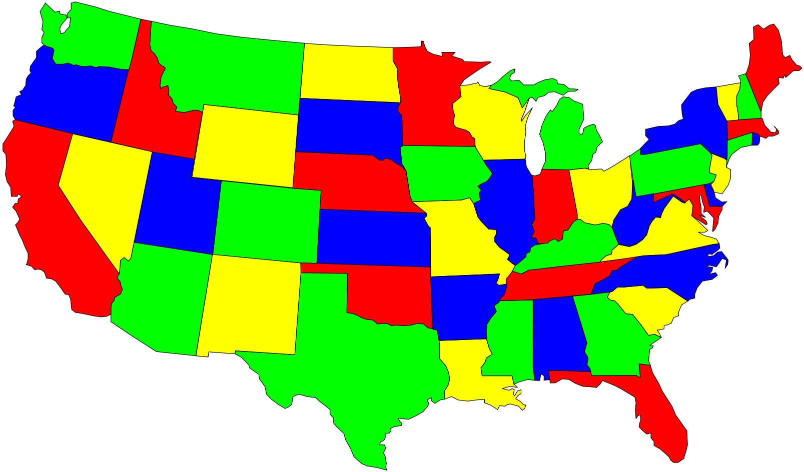
Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.



Map Coloring

Problem #1: Color a map of the USA so that adjoining states are always colored differently.



A Seating Problem

Problem #2: Eight cousins are at a family reunion. Some of them don't like each other.

A Seating Problem

Problem #2: Eight cousins are at a family reunion. Some of them don't like each other.

Cousin	Doesn't like
Alice	Bob, Chloe, Fred, Horace
Bob	Alice, Dave, Gina, Horace
Chloe	Alice
Dave	Bob, Fred, Horace
Edith	Horace
Fred	Alice, Dave
Gina	Bob, Horace
Horace	Alice, Bob, Dave, Edith, Fred, Gina

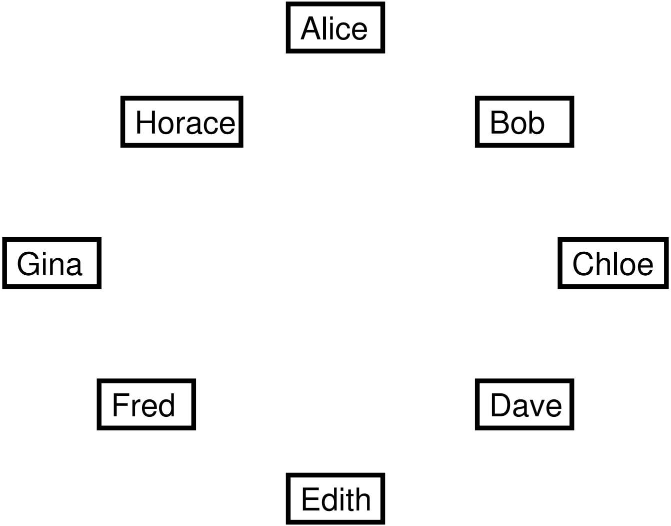
A Seating Problem

Problem #2: Eight cousins are at a family reunion. Some of them don't like each other.

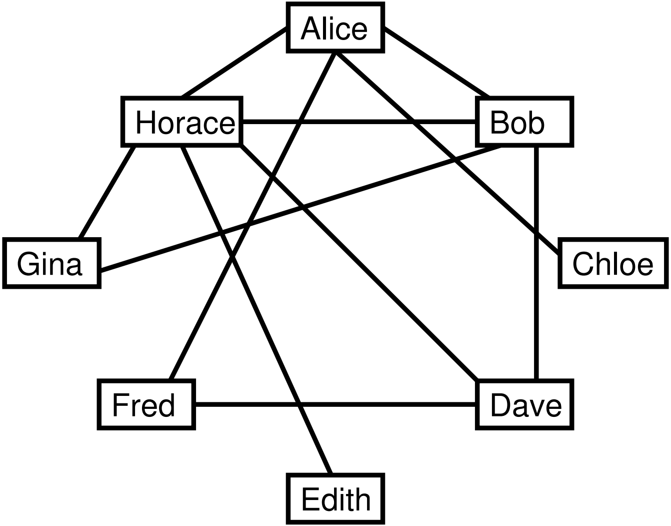
Cousin	Doesn't like
Alice	Bob, Chloe, Fred, Horace
Bob	Alice, Dave, Gina, Horace
Chloe	Alice
Dave	Bob, Fred, Horace
Edith	Horace
Fred	Alice, Dave
Gina	Bob, Horace
Horace	Alice, Bob, Dave, Edith, Fred, Gina

How many tables are necessary so that no one has to sit at a table with someone s/he doesn't like?

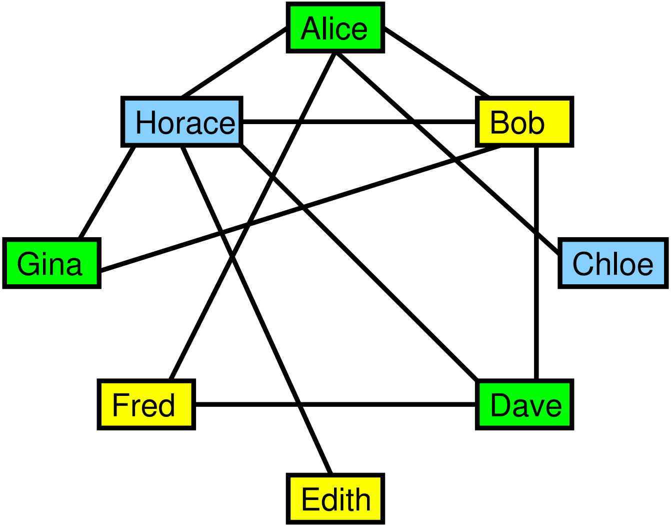
A Seating Problem



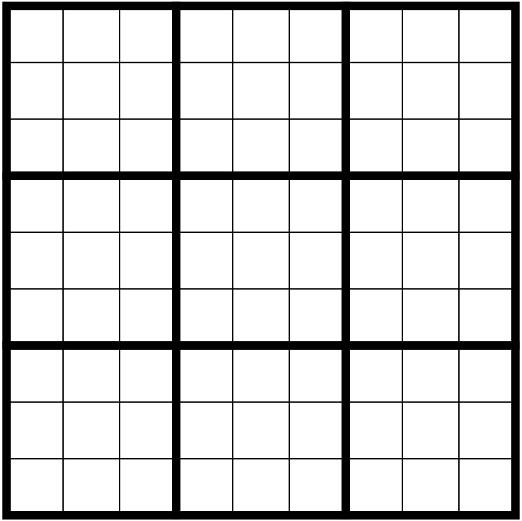
A Seating Problem



A Seating Problem



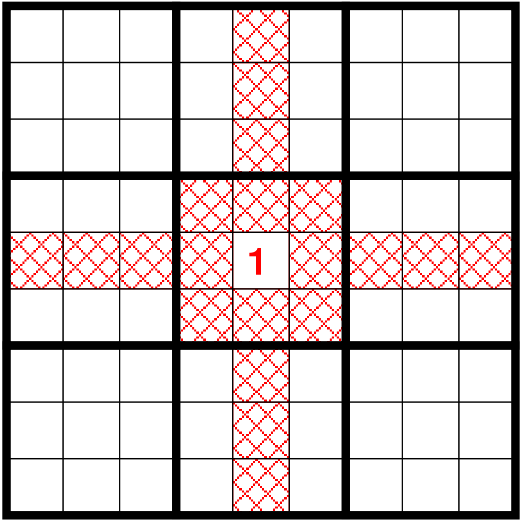
Sudoku



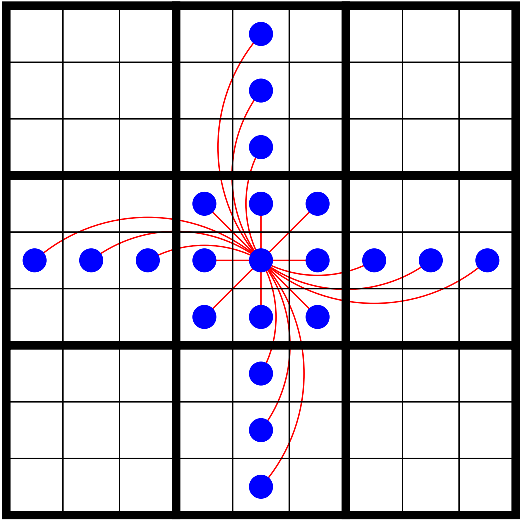
Sudoku

8	9	7	1	2	3	4	6	5
5	3	2	7	6	4	1	9	8
4	6	1	8	5	9	2	3	7
9	8	3	4	7	6	5	2	1
6	2	4	5	1	8	9	7	3
1	7	5	9	3	2	8	4	6
3	1	6	2	4	5	7	8	9
7	4	9	6	8	1	3	5	2
2	5	8	3	9	7	6	1	4

Sudoku



Sudoku



Sudoku

8	9	7	1	2	3	4	6	5
5	3	2	7	6	4	1	9	8
4	6	1	8	5	9	2	3	7
9	8	3	4	7	6	5	2	1
6	2	4	5	1	8	9	7	3
1	7	5	9	3	2	8	4	6
3	1	6	2	4	5	7	8	9
7	4	9	6	8	1	3	5	2
2	5	8	3	9	7	6	1	4

Graph Coloring

Suppose that G is a graph and k is a positive integer.

Definition: A k-**coloring** of G is a coloring of the vertices of G using k colors, such that adjacent vertices are colored with different colors.

Graph Coloring

Suppose that G is a graph and k is a positive integer.

Definition: A k-**coloring** of G is a coloring of the vertices of G using k colors, such that adjacent vertices are colored with different colors.

Definition: The **chromatic number** of G is the smallest number k for which a k-coloring of G is possible. (Notation: $\chi(G)$.)

Graph Coloring

Suppose that G is a graph and k is a positive integer.

Definition: A k-**coloring** of G is a coloring of the vertices of G using k colors, such that adjacent vertices are colored with different colors.

Definition: The **chromatic number** of G is the smallest number k for which a k-coloring of G is possible. (Notation: $\chi(G)$.)

Definition: A coloring of G is **optimal** if it uses exactly $\chi(G)$ colors (and no more).

Graph Coloring

- What are the chromatic numbers of special graphs that we know (e.g., complete graphs, circuits, trees, ...)
- How can we calculate the chromatic number of an arbitrary graph?
- How can we construct an optimal coloring?

Graph Coloring

- In a complete graph K_N, each vertex has to receive a different color.

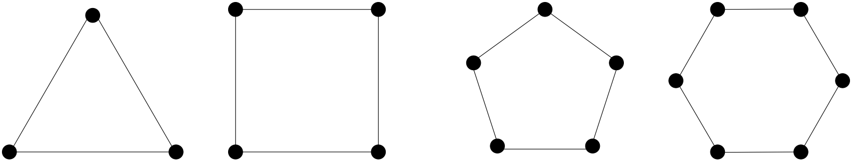
$$\boxed{\chi(K_N) = N.}$$

Graph Coloring

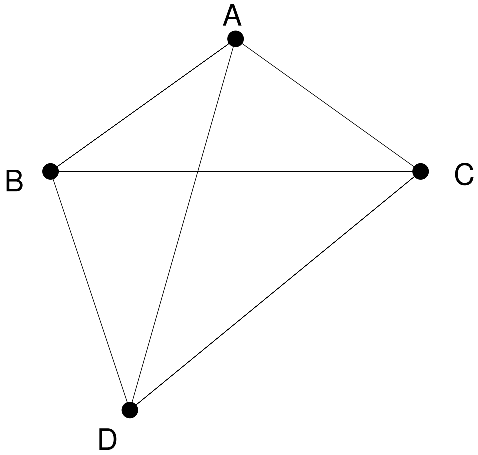
- In a complete graph K_N, each vertex has to receive a different color.

$$\boxed{\chi(K_N) = N.}$$

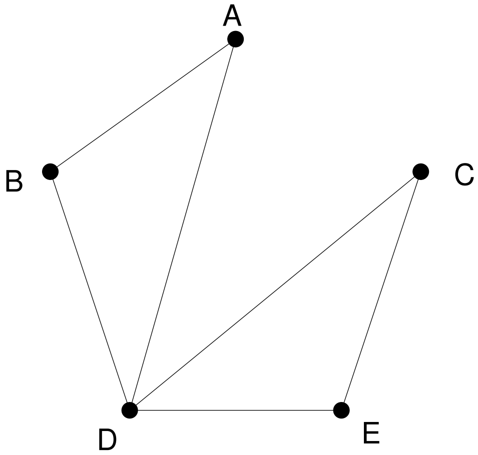
- In a circuit C_N...



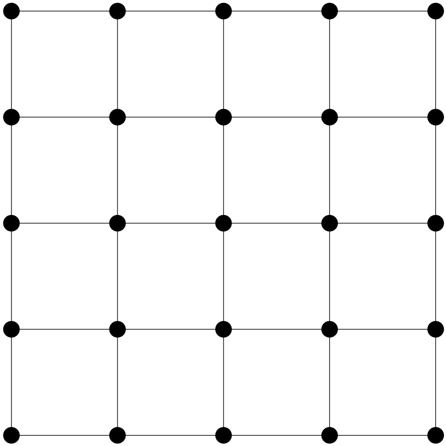
Graph Coloring Examples



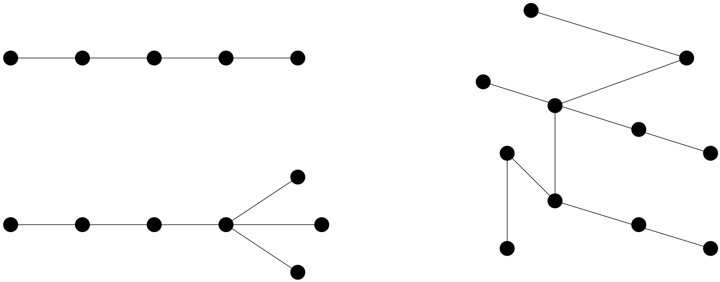
Graph Coloring Examples



Graph Coloring Examples



Graph Coloring Examples



A Graph Coloring Algorithm

What are we doing when we color a graph?

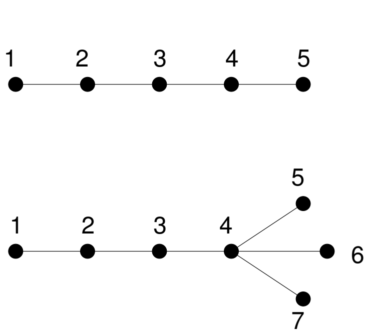
1. List the vertices in order: v_1, v_2, ..., v_n.
2. Number the colors 1, 2, ..., k.
3. Color the vertices in that order. Assign each vertex the smallest possible color.

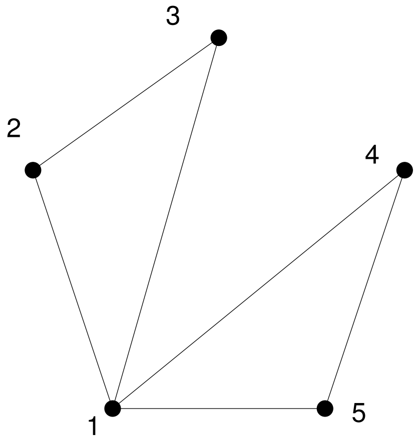
(E.g., if vertex v_6 has neighbors that have already been colored 1 and 2, but has no neighbors that have been colored 3, then assign color 3 to v_6.)

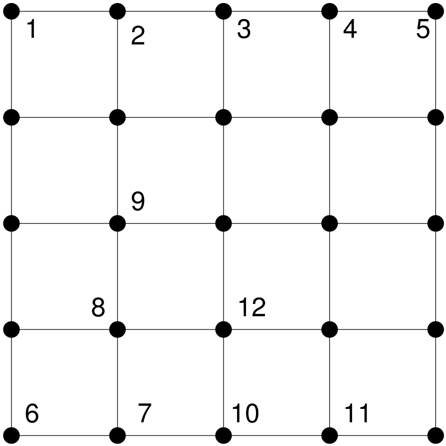
A Graph Coloring Algorithm

Question: Does this algorithm always produce an optimal coloring?

That is, does coloring a graph G in this way always use **only** $\chi(G)$ colors, never more than that?


1
2
3
4
5
1
2
3
4
5
6
7
1
2
3
4
5
6
7
8
9
10
11


3
2
4
1
5


1
2
3
4
5
9
8
12
6
7
10
11

Oops.

The Bad News

- This algorithm is **efficient** but **not optimal** — the number of colors required might be depends on the order in which the vertices are chosen.

The Bad News

- This algorithm is **efficient** but **not optimal** — the number of colors required might be depends on the order in which the vertices are chosen.
- There is always some order that produces an optimal coloring; however, it is very hard to know in advance what order to use.

The Bad News

- This algorithm is **efficient** but **not optimal** — the number of colors required might be depends on the order in which the vertices are chosen.
- There is always some order that produces an optimal coloring; however, it is very hard to know in advance what order to use.
- Finding an optimal coloring is theoretically **just as hard as the Traveling Salesman Problem** — there is no efficient algorithm known.

The Good News

1. List the vertices in order: v_1, v_2, ..., v_n.
2. Number the colors 1, 2, ..., k.
3. Color the vertices in that order. Assign each vertex the smallest possible color.

Observation: If vertex v_i has degree d, then at least one of the colors 1, 2, ..., $d + 1$ is available when it is v_i's turn to get colored.

That means we can say at least something about $\chi(G)$.

If every vertex in a graph G has degree Δ or less, then $\chi(G) \leq \Delta + 1$.

Actually, we can do even better than that!

Brooks' Theorem: If every vertex in a graph G has degree Δ or less, then $\chi(G) \leq \Delta$, unless G is a complete graph or a cycle of odd length.