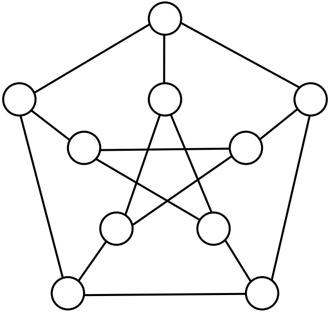
Lecture outline

Graph coloring

- Examples
- Applications
- Algorithms

Graph coloring

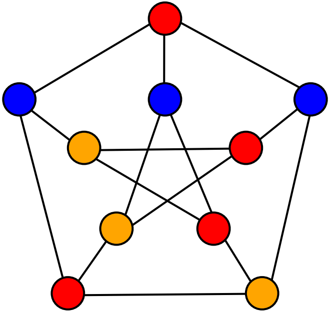
Adjacent nodes must have different colors.



How many colors do we need?

Graph coloring

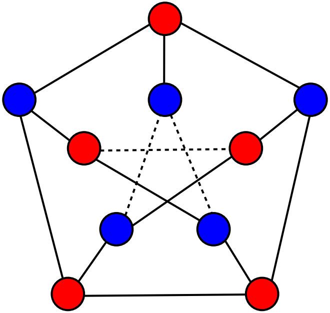
Neighbors must have different colors



How many colors do we need?

Graph coloring

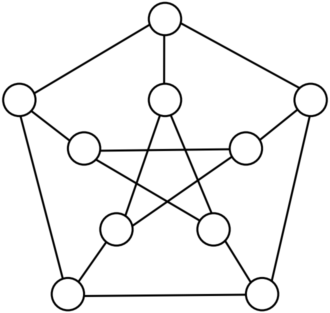
Neighbors must have different colors



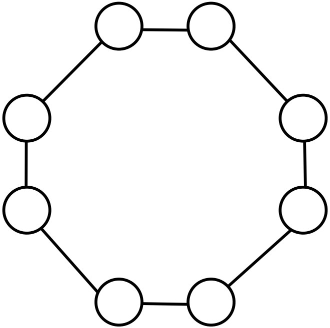
Not valid

Definitions

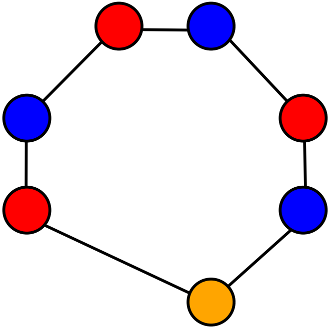
- **k-Colorable**: can be colored with k colors
- **chromatic number:** min number of colors needed



Cycle graph

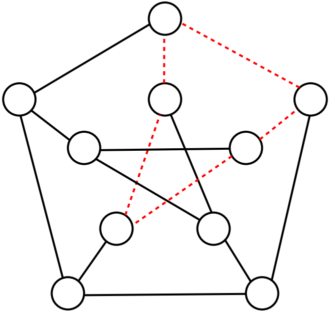


Cycle graph



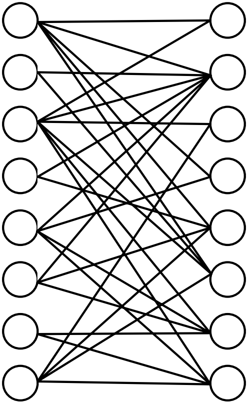
Odd-length cycles require 3 colors

Cycle graph



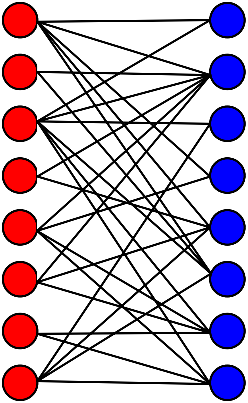
No way to color with only 2 colors

Bipartite graph

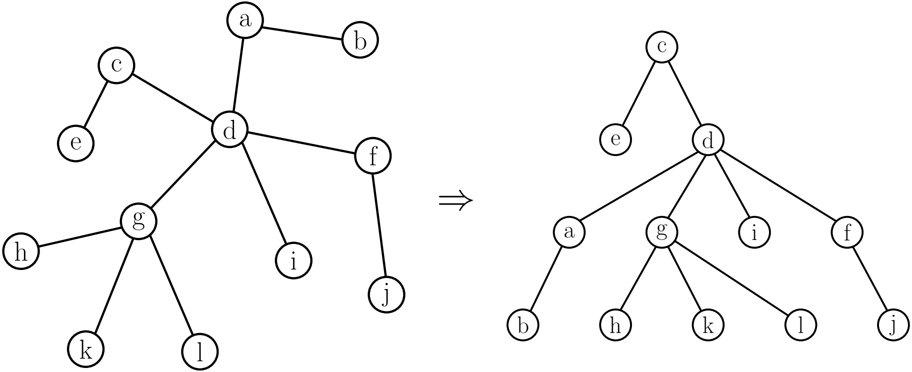


Bipartite graph

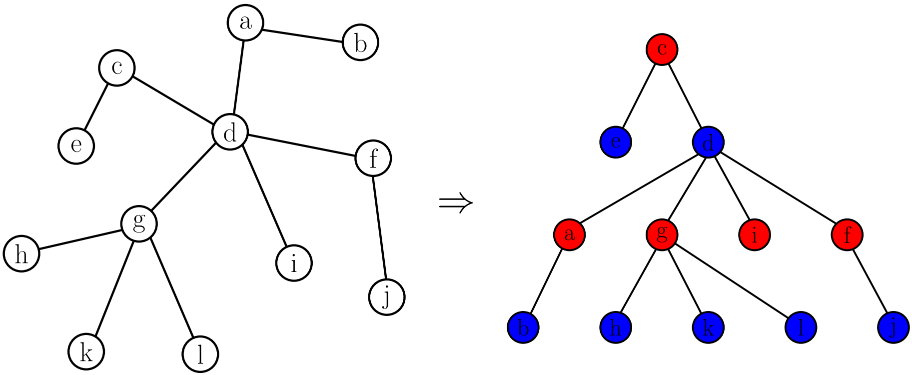
Bipartite $\Leftrightarrow$ 2-colorable



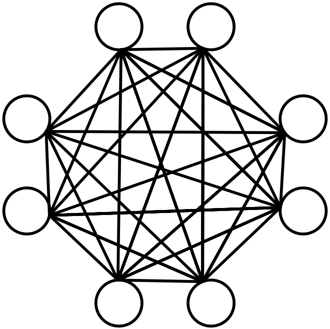
Tree



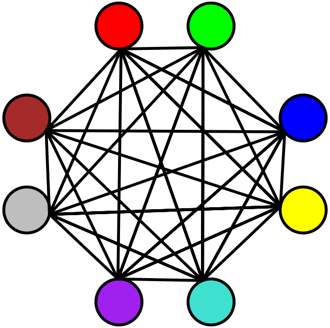
Tree



Clique

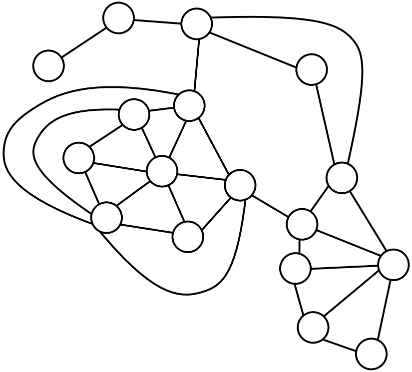


Clique



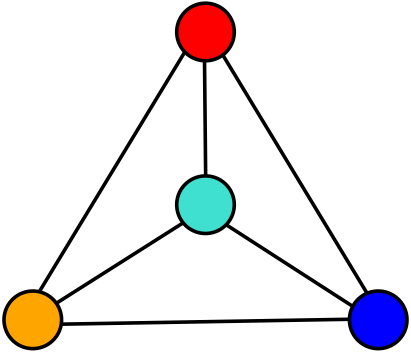
If G contains a Clique of size k, then it requires at least k colors

Planar graph



Planar graph

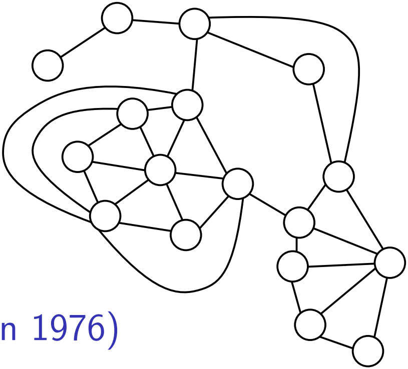
exam-like question: draw the smallest planar graph with chromatic number 4



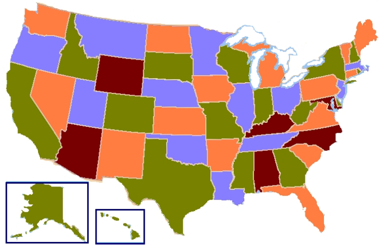
Planar graph coloring

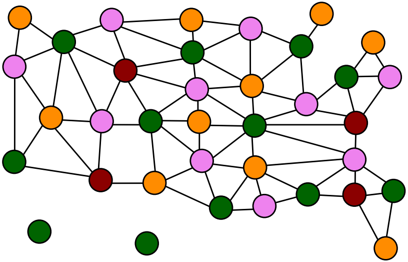
- Some planar graphs require 4 colors
- Any planar graph is 5-colorable (1890)
- Can **every** planar graph be colored with 4 colors?

(Stated in 1852, solved in 1976)



map coloring





map coloring ⇔ planar graph coloring

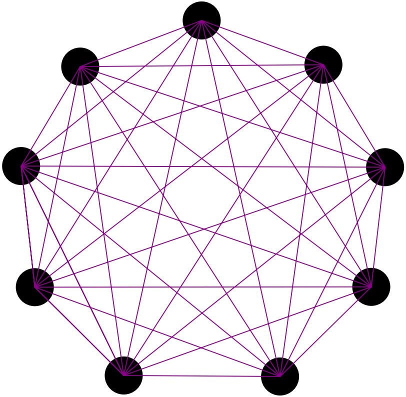
4 color theorem

Every planar graph (and map) can be colored with 4 colors.

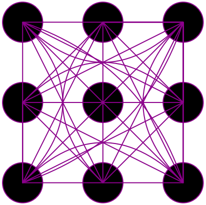
Graph coloring

- Examples
- **Applications**
  - Register allocation
  - Logistics problems
  - Scheduling problems
  - Puzzles
- Algorithms

Applications

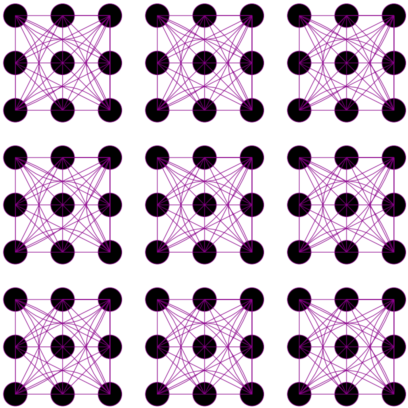


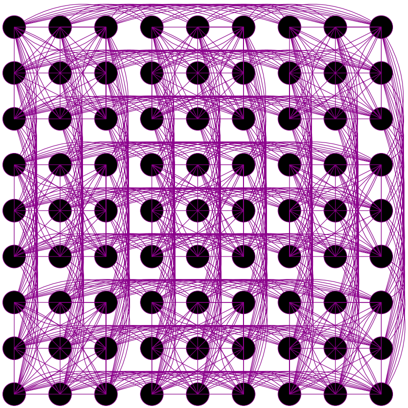
K_9



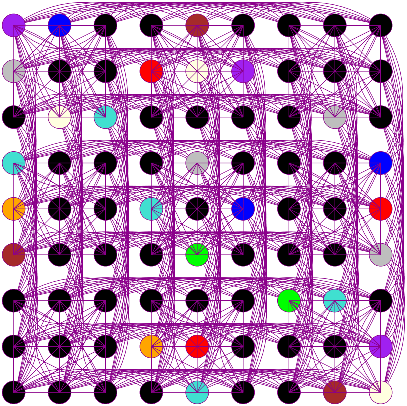
K_9

Applications





Sudoku



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

Register allocation

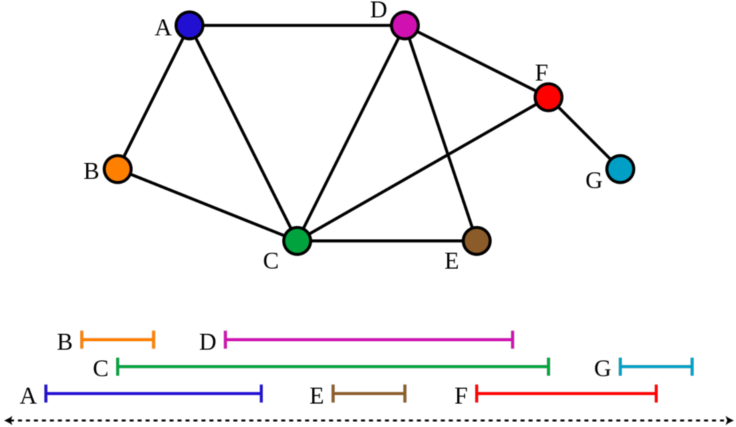
- Limited registers (16 in a x86-64 CPU)
- Goal: avoid storing local variables in main memory
- If a variable X is put in a register, other variables can't use that register until the last use of X
- Need to decide which variable goes to which register

```
for (i=0; i<n; i++){
    ...
}
for (j=0; j<n; j++){
    ...
}
```

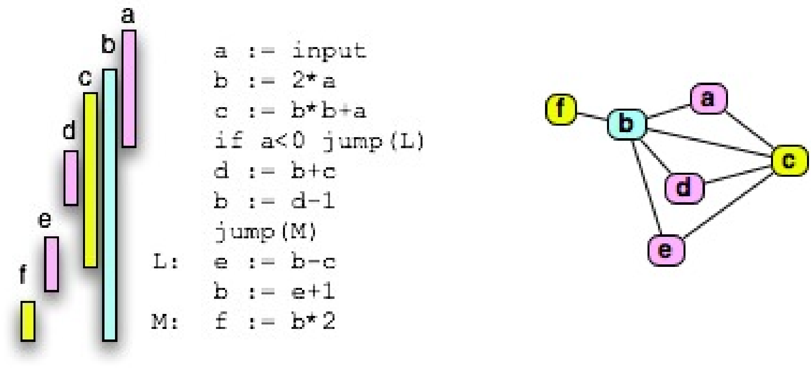
`i` and `j` can go in the same register, but `i` and `n` can't

Interval graph

- Nodes: intervals
- Edges: overlapping intervals



Register allocation



same color = same register

Source: scienceblogs.com/goodmath/2007/06/28/graph-coloring-algorithms-1/

Logistics problems

Grouping things into as few groups as possible

- Nodes: things
- Edges: things that can't be together
- Each color is a group

Examples: register allocation, chemical storage, ...

Scheduling problems

Doing a set of tasks in the shortest time

- Nodes: tasks
- Edges: tasks that can't be done at the same time
- Each color is a time slot

Examples: exam scheduling, ...

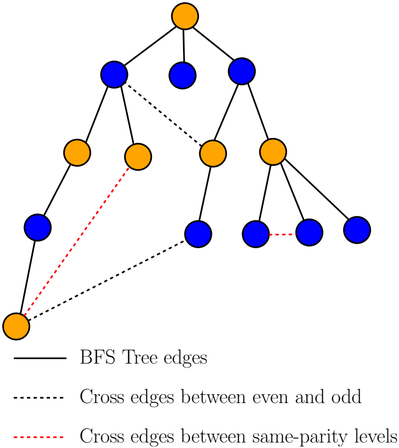
Graph coloring

- Examples
- Applications
- **Algorithms**
  - Can this graph be colored with k colors?
  - Find an optimal coloring
  - Or suboptimal coloring more quickly

2-colorability

Is a graph 2-colorable? (is a graph bipartite?)

- Run BFS
- Color nodes by level in the BFS tree
- If there are cross edges between same-parity levelss $\Rightarrow$ not 2-colorable



3-colorability

Is a graph 3-colorable?

- different colorings with n nodes and 3 colors:

$$3^n \qquad \text{or} \qquad n^3 \qquad ?$$

In general: n nodes and k colors $\Rightarrow$ k^n different colorings

Exponential!

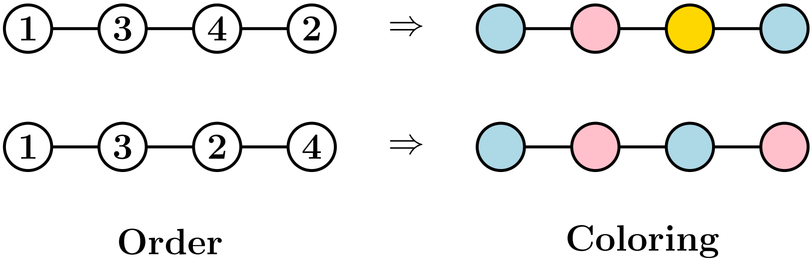
- Next lecture: 3-colorability is NP-complete
  (even for planar graphs)

Greedy coloring

Algorithm:

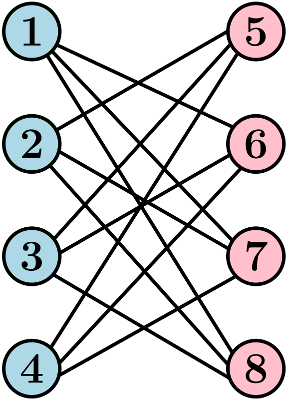
1. Consider the nodes in some order
2. Give each node in this order the first color not used by its neighbors

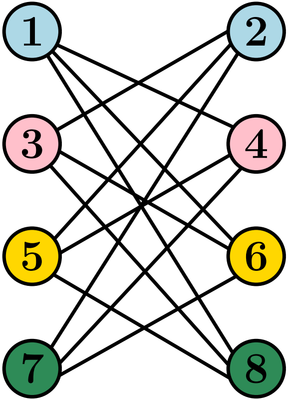
- Not optimal: depends on order



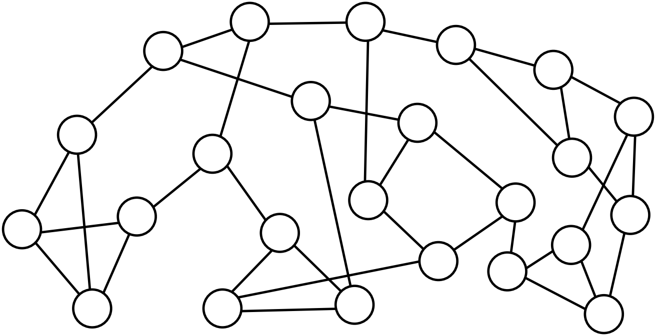
Greedy coloring

It can be really bad





k-regular graph



How many colors are needed?

Greedy upper bound

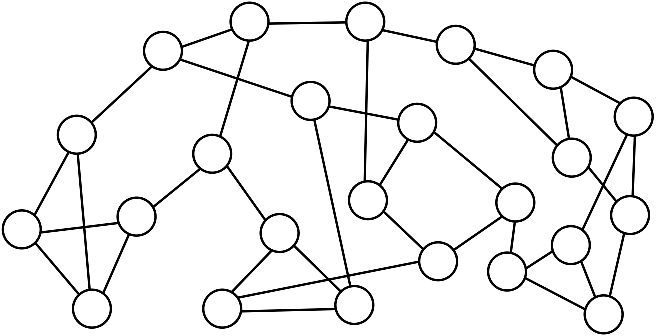
Algorithm:

1. Consider the nodes in some order
2. Give each node in this order the first color not used by its neighbors

If the max degree of any node is r, the graph is $(r+1)$-colorable

Proof: let u be any node. When it's time to color u, even if all its neighbors are already colored, they can use at most r colors, so $r+1$ colors suffice to color u.

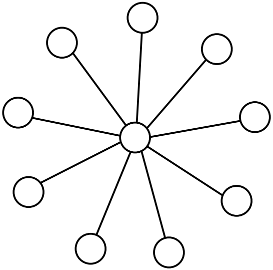
k-regular graph



Greedy coloring $\Rightarrow$ (k+1)-colorable

Greedy upper bound

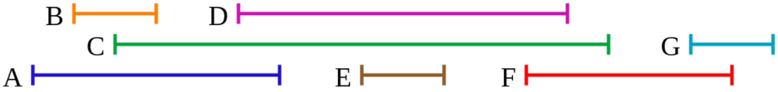
If the max degree of any node is r,
the graph is $(r+1)$-colorable



Can still be a bad upper bound

Greedy for interval graphs

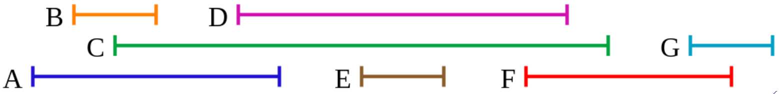
- Suppose the biggest clique has size k
- The chromatic number must be at least k
- If nodes are sorted by starting point, greedy coloring finds a k-coloring
- Therefore, it is **optimal**



Greedy for interval graphs

If nodes are sorted by starting point, greedy coloring finds a k-coloring. **Proof:**

1. Let $I = (I_s, I_e)$ be any interval
2. Any neighbor of I must end after I_s
3. Any already-colored neighbor of I must start before I_s
4. (2. and 3.) $\Rightarrow$ I and the already-colored neighbors of I intersect at I_s
5. If the max clique size is k, there are at most $k - 1$ already-colored neighbors of I, and I can use color k



Overview

Graph coloring

- Examples
  - Bipartite graphs
  - Cliques
  - Planar graphs
- Applications
  - Sudoku
  - Register allocation
  - Logistics problems
  - Scheduling problems
- Algorithms
  - 2-coloring
  - Greedy coloring

Thanks!