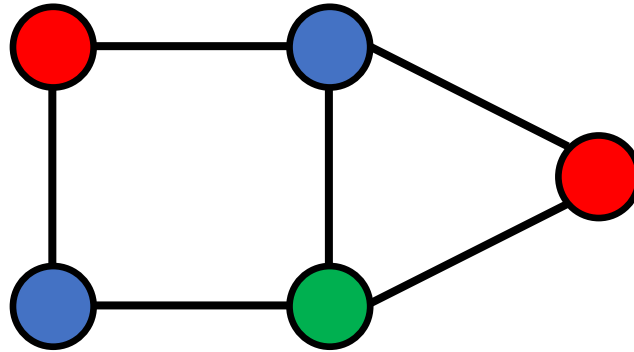


Randomized Algorithms

CSCI 432

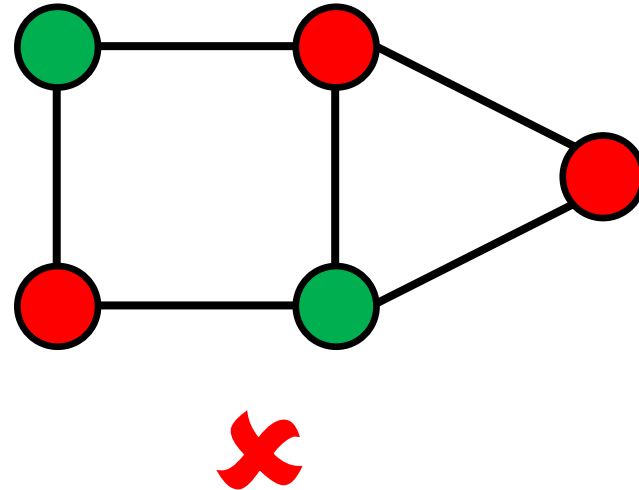
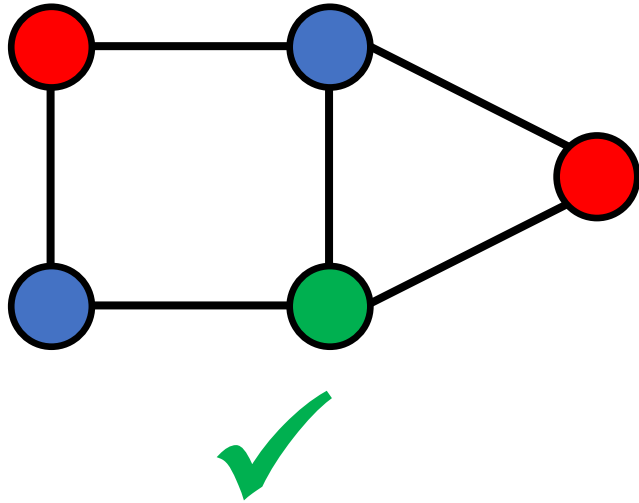
Graph Coloring

Family of graph problems that involve coloring vertices (or edges) subject to various constraints.



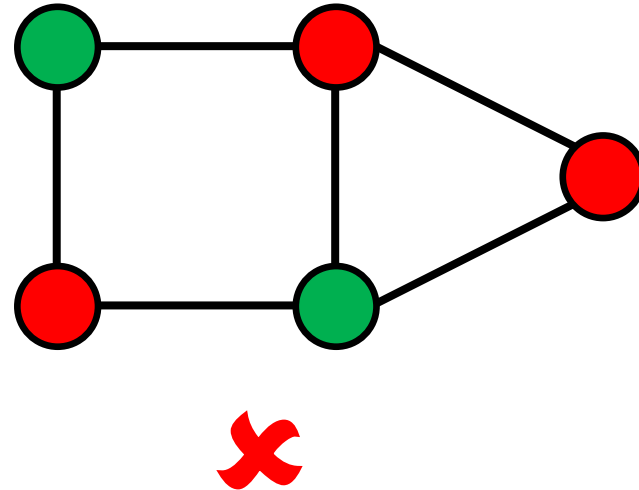
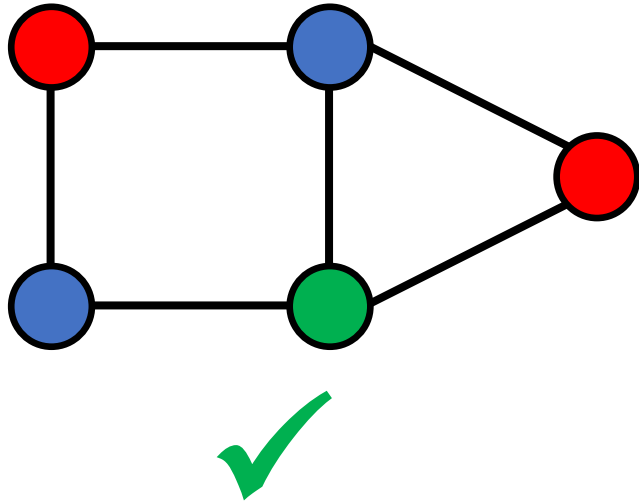
Graph Coloring

Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



Graph Coloring

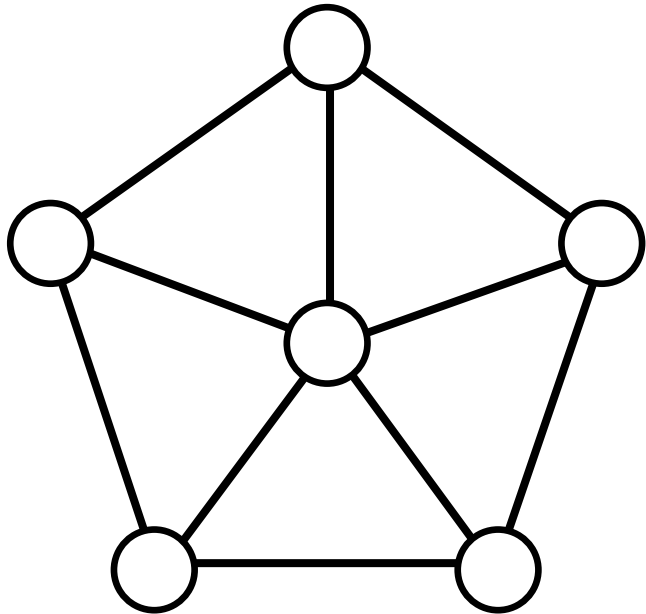
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



NP-Hard

Graph Coloring

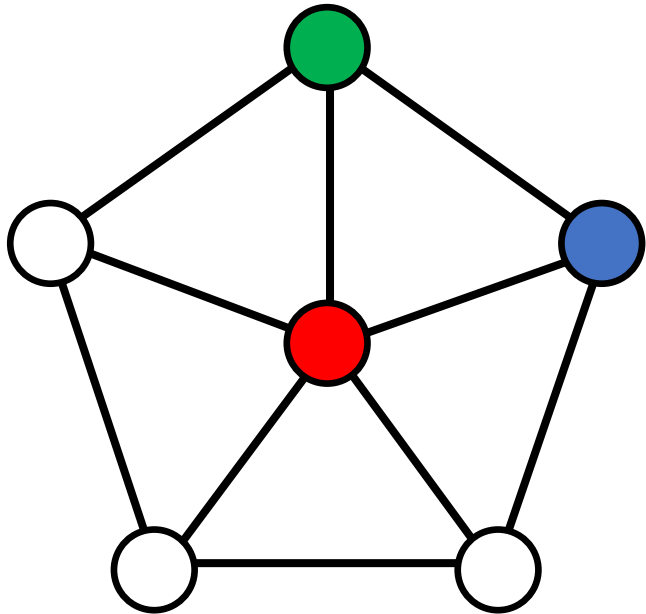
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

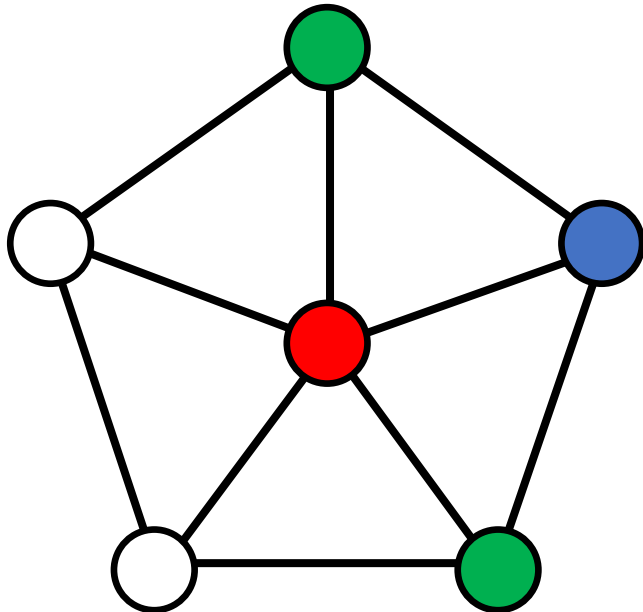
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

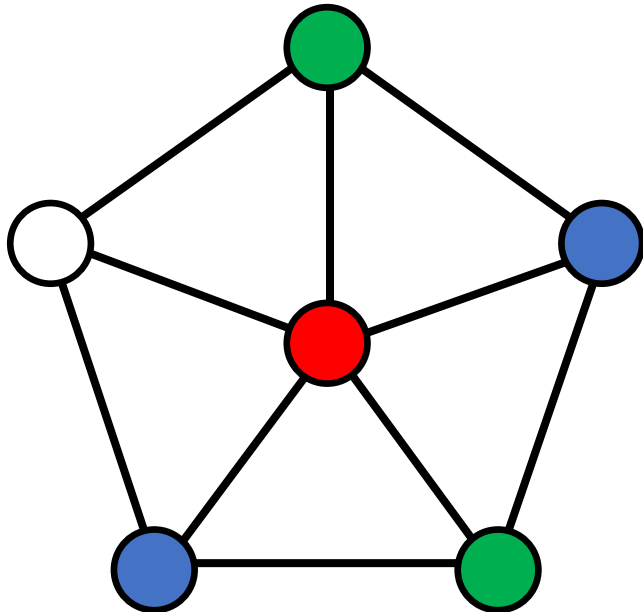
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

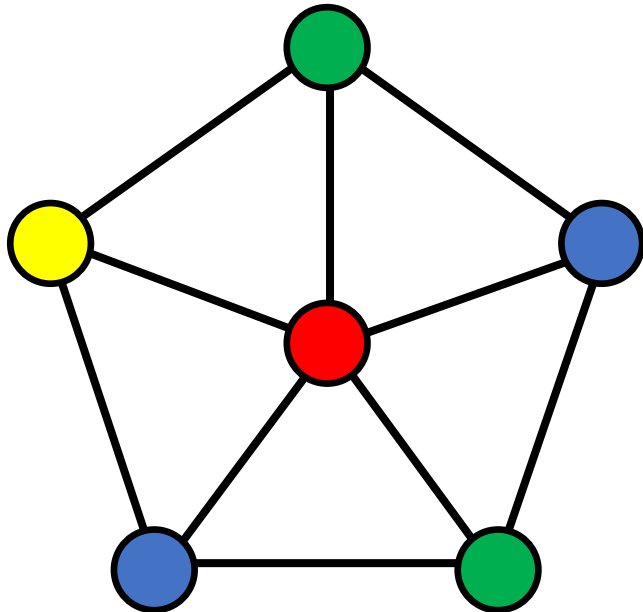
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

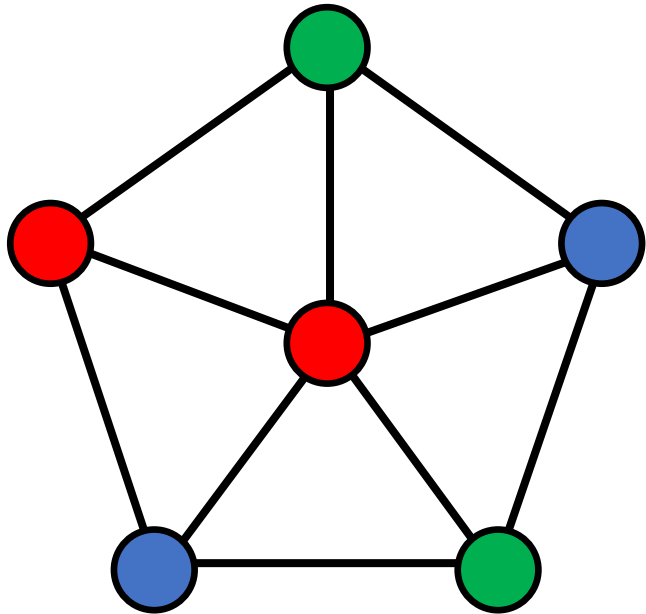
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



How many colors are needed to color this graph?

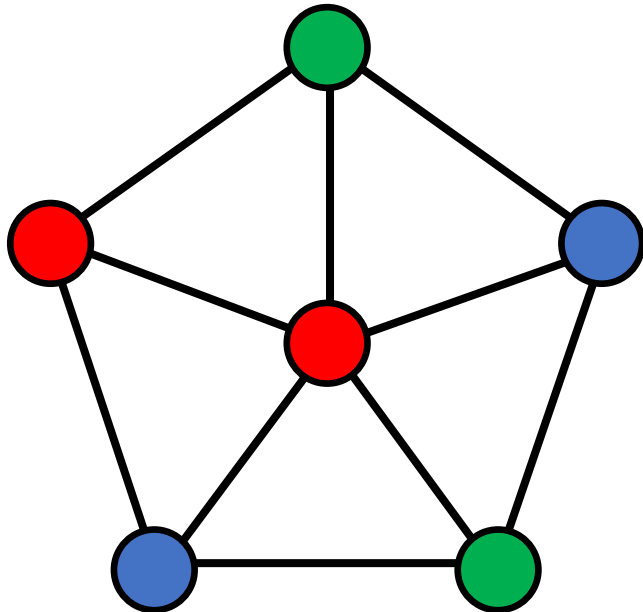
Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Max 3-Coloring

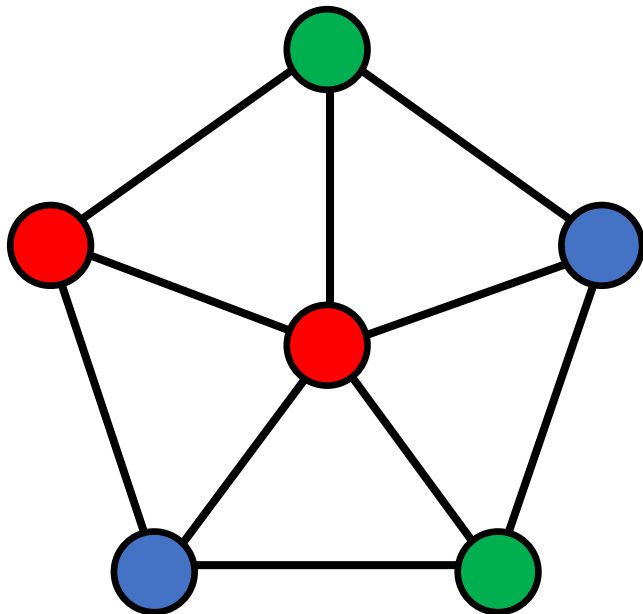
An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm?

Max 3-Coloring

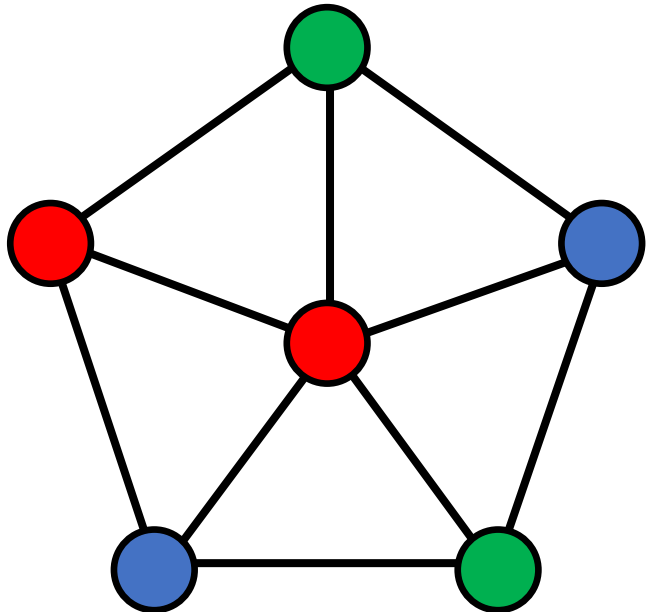
An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm: For each vertex, randomly assign it one of the three colors with probability $1/3$.

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.

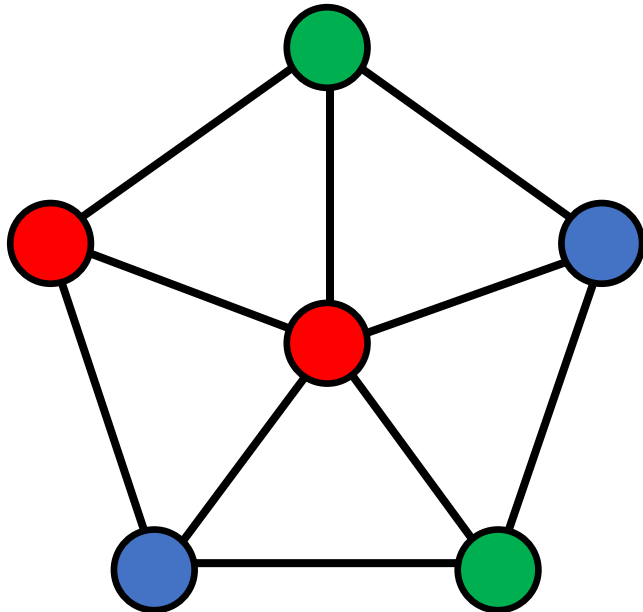


Algorithm: For each vertex, randomly assign it one of the three colors with probability $1/3$.

Running time?

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.

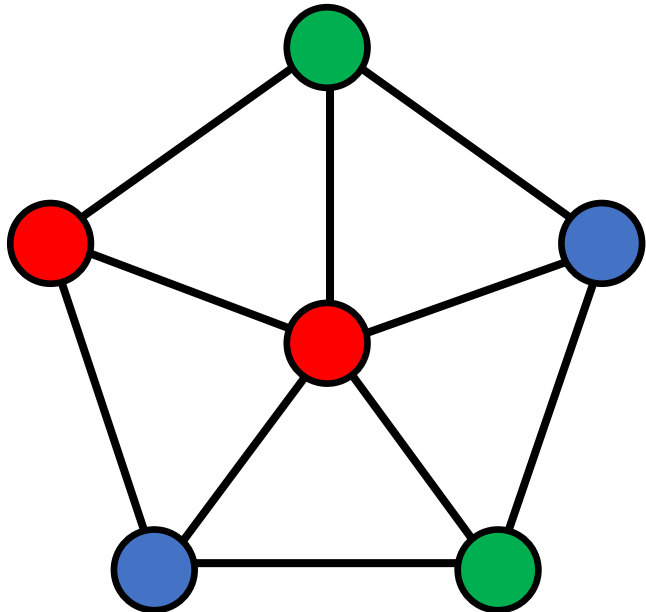


Algorithm: For each vertex, randomly assign it one of the three colors with probability $1/3$.

Running time? $O(n)$

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



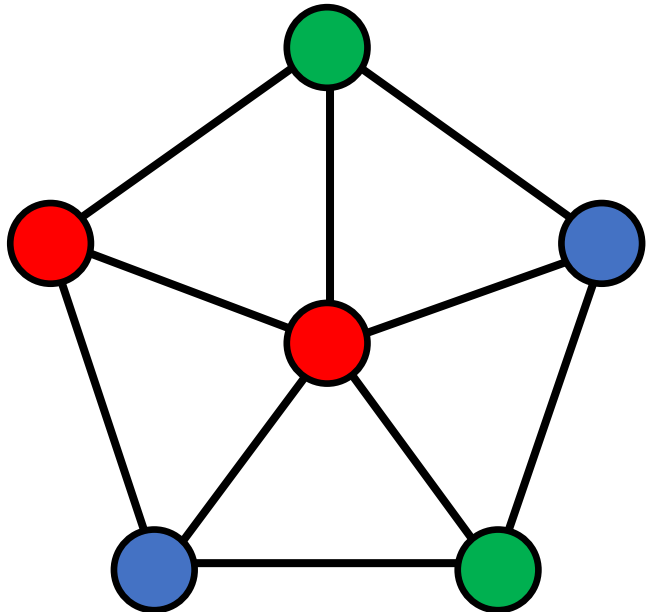
Algorithm: For each vertex, randomly assign it one of the three colors with probability $1/3$.

Running time? $O(n)$

Valid?

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



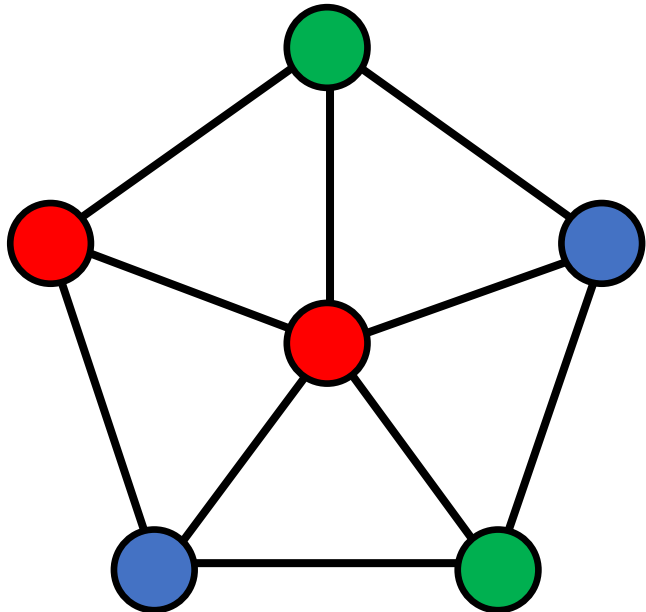
Algorithm: For each vertex, randomly assign it one of the three colors with probability $1/3$.

Running time? $O(n)$

Valid? Yes. Colors all vertices with ≤ 3 colors.

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm: For each vertex, randomly assign it one of the three colors with probability $1/3$.

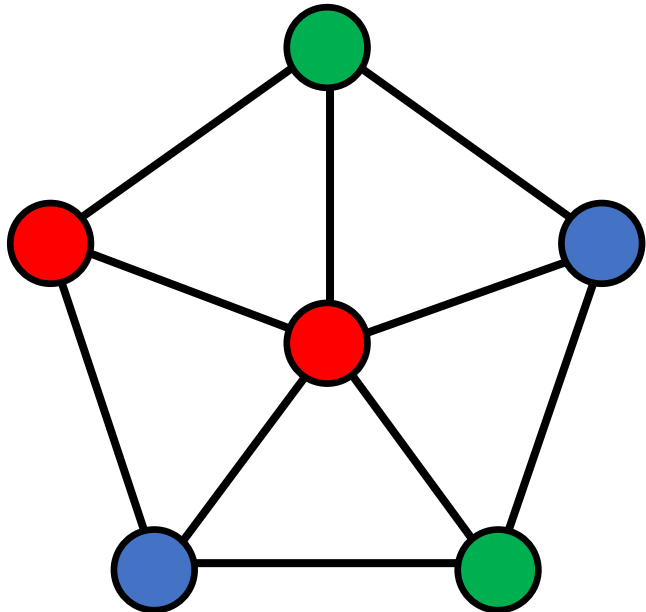
Running time? $O(n)$

Valid? Yes. Colors all vertices with ≤ 3 colors.

Optimal?

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm: For each vertex, randomly assign it one of the three colors with probability $1/3$.

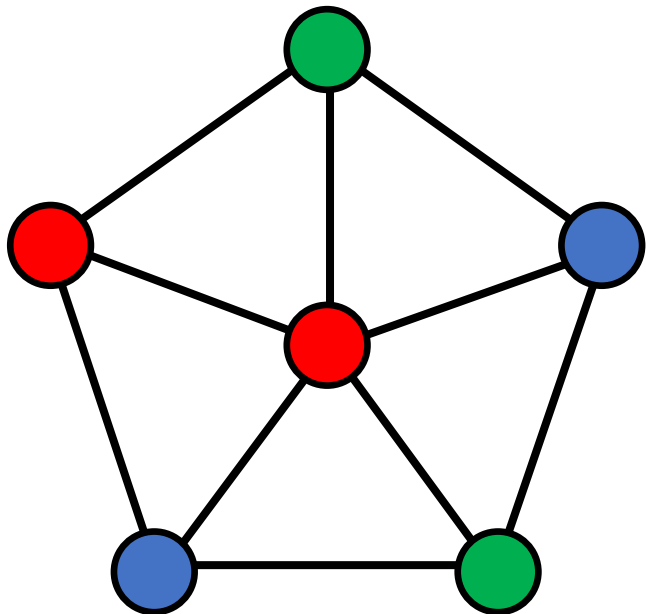
Running time? $O(n)$

Valid? Yes. Colors all vertices with ≤ 3 colors.

Optimal? Probably not...

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm: For each vertex, randomly assign it one of the three colors with probability $1/3$.

Running time? $O(n)$

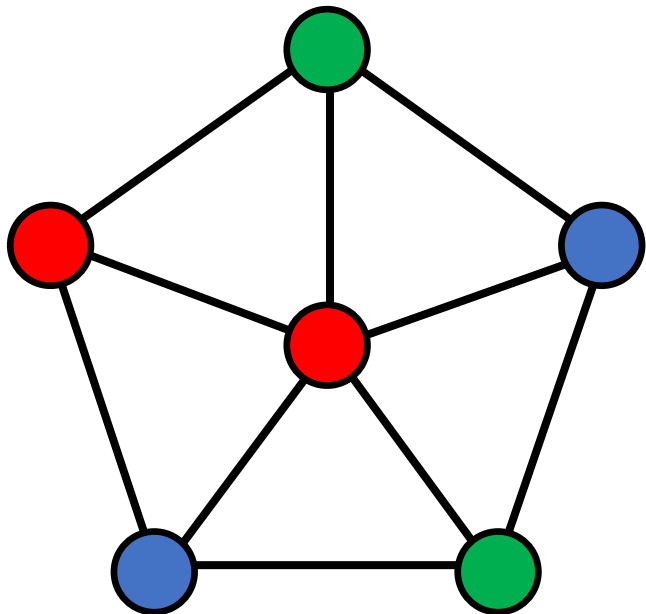
Valid? Yes. Colors all vertices with ≤ 3 colors.

Optimal? Probably not...

Goal: Show this algorithm is expected to satisfy at least $\frac{2}{3}$ of the edges.

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm: For each vertex, randomly assign it one of the three colors with probability $1/3$.

Running time? $O(n)$

Valid? Yes. Colors all vertices with ≤ 3 colors.

Optimal? Probably not...

Goal: Show this algorithm is **expected** to satisfy at least $\frac{2}{3}$ of the edges.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

What is the probability that a single edge is satisfied?

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

What is the probability that a single edge is satisfied?

$$\text{Probability of some outcome} = \frac{\# \text{ events we care about}}{\# \text{ all possible events}}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

What is the probability that a single edge is satisfied?

All possibilities:

$(b,b), (b,g), (b,r), (g,b), (g,g), (g,r), (r,b), (r,g), (r,r)$

$$\text{Probability of some outcome} = \frac{\# \text{ events we care about}}{\# \text{ all possible events}} = \frac{?}{9}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

What is the probability that a single edge is satisfied?

All possibilities:

 $(b,b), (b,g), (b,r), (g,b), (g,g), (g,r), (r,b), (r,g), (r,r)$

$$\text{Probability of some outcome} = \frac{\# \text{ events we care about}}{\# \text{ all possible events}} = \frac{6}{9}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

What is the probability that a single edge is satisfied?

All possibilities:

 $(b,b), (b,g), (b,r), (g,b), (g,g), (g,r), (r,b), (r,g), (r,r)$

$$\text{Probability of some outcome} = \frac{\# \text{ events we care about}}{\# \text{ all possible events}} = \frac{6}{9} = \frac{2}{3}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define X_e

Random Variable: A variable whose value is determined by the outcome of a random process.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

Random Variable: A variable whose value is determined by the outcome of a random process.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

What is $\sum_e X_e$?

Random Variable: A variable whose value is determined by the outcome of a random process.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

Random Variable: A variable whose value is determined by the outcome of a random process.

What is $\sum_e X_e$?

The number of edges satisfied by the algorithm.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

Random Variable: A variable whose value is determined by the outcome of a random process.

What is $\sum_e X_e$?

The number of edges satisfied by the algorithm.

$\sum_e X_e$ changes for each run of the algorithm, depending on the specific values of each X_e .

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

Expected Value: The average value of a random variable over many repetitions of the random process.

What is $\sum_e X_e$?

The number of edges satisfied by the algorithm.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

Expected Value: The average value of a random variable over many repetitions of the random process.

What is $\sum_e X_e$?

The number of edges satisfied by the algorithm.

What is $E[\sum_e X_e]$?

The most likely number of edges satisfied by the algorithm.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

How many edges we expect to satisfy $= E[\sum_e X_e]$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

How many edges we expect to satisfy $= E[\sum_e X_e] = \sum_e E[X_e]$

Linearity of Expectation:

$$E[X + Y] = E[X] + E[Y]$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

Expected Value: The average value of a random variable over many repetitions of the random process.

How many edges we expect to satisfy $= E[\sum_e X_e] = \sum_e E[X_e]$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

Expected Value: The average value of a random variable over many repetitions of the random process.

$E[X_e] = ?$

How many edges we expect to satisfy $= E[\sum_e X_e] = \sum_e E[X_e]$

The expected value of a random variable ($E[Y]$) is the sum of every possible outcome (y_k) times the probability of that outcome (p_k):

$$E[Y] = \sum_k y_k p_k$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

How many edges we expect to satisfy $= E[\sum_e X_e] = \sum_e E[X_e]$

The expected value of a random variable ($E[Y]$) is the sum of every possible outcome (y_k) times the probability of that outcome (p_k):

$$E[Y] = \sum_k y_k p_k$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \text{How many edges we expect to satisfy} &= E[\sum_e X_e] = \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} \end{aligned}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \text{How many edges we expect to satisfy} &= E[\sum_e X_e] = \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, & \text{if } e \text{ is satisfied} \\ 0, & \text{if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1 \left(\frac{2}{3} \right) + 0 \left(\frac{1}{3} \right) = \frac{2}{3}$$

$$\begin{aligned} \text{How many edges we expect to satisfy} &= E[\sum_e X_e] = \sum_e E[X_e] \\ &= \sum_e \frac{2}{3} = \frac{2}{3} |E| \end{aligned}$$

$$\Rightarrow E[ALG] = \frac{2}{3} |E|$$

Amplification?

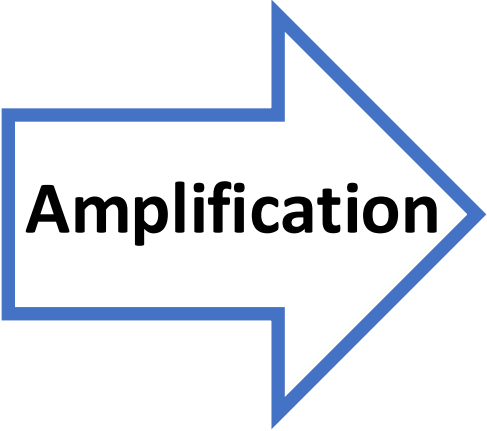
Running Time: $O(n)$

Performance: $E[ALG] = \frac{2}{3} |E|$

Freivalds'

Running Time: $O(n^2)$

Performance: Correct $\geq 50\%$
of the time



Amplification

Running Time: Polynomial
Performance: Almost
guaranteed

Amplification?

Running Time: $O(n)$

Performance: $E[ALG] = \frac{2}{3} |E|$

What is the probability we get the optimal answer?

Amplification?

Running Time: $O(n)$

Performance: $E[ALG] = \frac{2}{3} |E|$

What is the probability we get the optimal answer?

Worst case: Graph is 3-colorable (i.e., All edges are satisfiable). Then the probability we randomly select that coloring is: $\left(\frac{1}{3}\right)^n$

Amplification?

Running Time: $O(n)$

Performance: $E[ALG] = \frac{2}{3} |E|$

What is the probability we get the optimal answer?

Worst case: Graph is 3-colorable (i.e., All edges are satisfiable). Then the probability we randomly select that coloring is: $\left(\frac{1}{3}\right)^n$

Probability we find optimal coloring $\geq \frac{1}{3^n} \Rightarrow$ Probability we fail is $\leq 1 - \frac{1}{3^n}$

Amplification?

Running Time: $O(n)$

Performance: $E[ALG] = \frac{2}{3} |E|$

What is the probability we get the optimal answer?

Worst case: Graph is 3-colorable (i.e., All edges are satisfiable). Then the probability we randomly select that coloring is: $\left(\frac{1}{3}\right)^n$

Probability we find optimal coloring $\geq \frac{1}{3^n} \Rightarrow$ Probability we fail is $\leq 1 - \frac{1}{3^n}$

Let's run it multiple (k) times. Probability they all fail $\leq \left(1 - \frac{1}{3^n}\right)^k$

Amplification?

Running Time: $O(n)$

Performance: $E[ALG] = \frac{2}{3} |E|$

What is the probability we get the optimal answer?

Worst case: Graph is 3-colorable (i.e., All edges are satisfiable). Then the probability we randomly select that coloring is: $\left(\frac{1}{3}\right)^n$

Probability we find optimal coloring $\geq \frac{1}{3^n} \Rightarrow$ Probability we fail is $\leq 1 - \frac{1}{3^n}$

Let's run it multiple (k) times. Probability they all fail $\leq \left(1 - \frac{1}{3^n}\right)^k$

$\Rightarrow$ Probability at least one succeeds $\geq 1 - \left(1 - \frac{1}{3^n}\right)^k$

Amplification?

Running Time: $O(n)$

Performance: $E[ALG] = \frac{2}{3} |E|$

What is the probability we get the optimal answer?

Worst case: Graph is 3-colorable (i.e., All edges are satisfiable). Then the probability we randomly select that coloring is: $\left(\frac{1}{3}\right)^n$

Probability we find optimal coloring $\geq \frac{1}{3^n} \Rightarrow$ Probability we fail is $\leq 1 - \frac{1}{3^n}$

Let's run it multiple (k) times. Probability they all fail $\leq \left(1 - \frac{1}{3^n}\right)^k$

$\Rightarrow$ Probability at least one succeeds $\geq 1 - \left(1 - \frac{1}{3^n}\right)^k$

To get "With High Probability", set $\left(1 - \frac{1}{3^n}\right)^k \leq \frac{1}{n}$ and solve for k .

An event occurs **with high probability** if its probability goes to 1 as n goes to infinity.

Amplification?

Running Time: $O(n)$

Performance: $E[ALG] = \frac{2}{3} |E|$

What is the probability we get the optimal answer?

Worst case: Graph is 3-colorable (i.e., All edges are satisfiable). Then the probability we randomly select that coloring is: $\left(\frac{1}{3}\right)^n$

Probability we find optimal coloring $\geq \frac{1}{3^n} \Rightarrow$ Probability we fail is $\leq 1 - \frac{1}{3^n}$

Let's run it multiple (k) times. Probability they all fail $\leq \left(1 - \frac{1}{3^n}\right)^k$

$\Rightarrow$ Probability at least one succeeds $\geq 1 - \left(1 - \frac{1}{3^n}\right)^k$

To get "With High Probability", set $\left(1 - \frac{1}{3^n}\right)^k \leq \frac{1}{n}$ and solve for k .

$$\Rightarrow k \geq 3^n \ln n$$

Amplification?

Running Time: $O(n)$

Performance: $E[ALG] = \frac{2}{3} |E|$

What is the probability we get the optimal answer?

Worst case: Graph is 3-colorable (i.e., All edges are satisfiable). Then the probability we randomly select that coloring is: $\left(\frac{1}{3}\right)^n$

Probability we find optimal coloring $\geq \frac{1}{3^n} \Rightarrow$ Probability we fail is $\leq 1 - \frac{1}{3^n}$

Let's run it multiple (k) times. Probability they all fail $\leq \left(1 - \frac{1}{3^n}\right)^k$

$\Rightarrow$ Probability at least one succeeds $\geq 1 - \left(1 - \frac{1}{3^n}\right)^k$

To get "With High Probability", set $\left(1 - \frac{1}{3^n}\right)^k \leq \frac{1}{n}$ and solve for k .

$$\Rightarrow k \geq 3^n \ln n$$

No amount of polynomial amplification will rescue us.