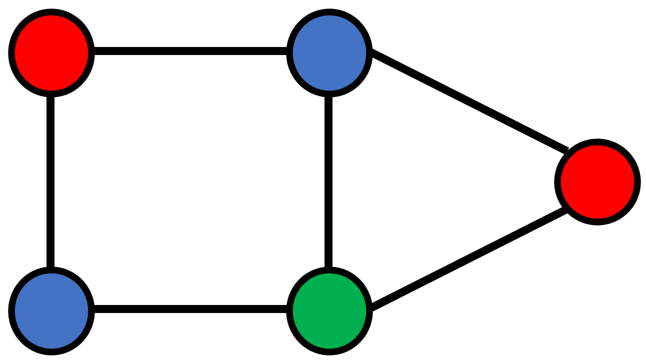
Randomized Algorithms

CSCI 432

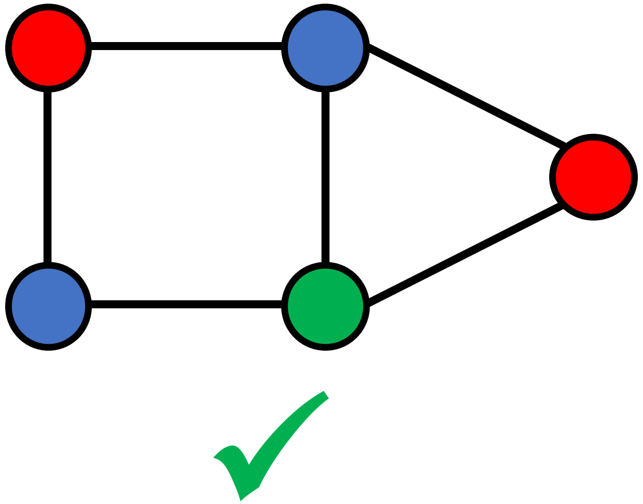
Graph Coloring

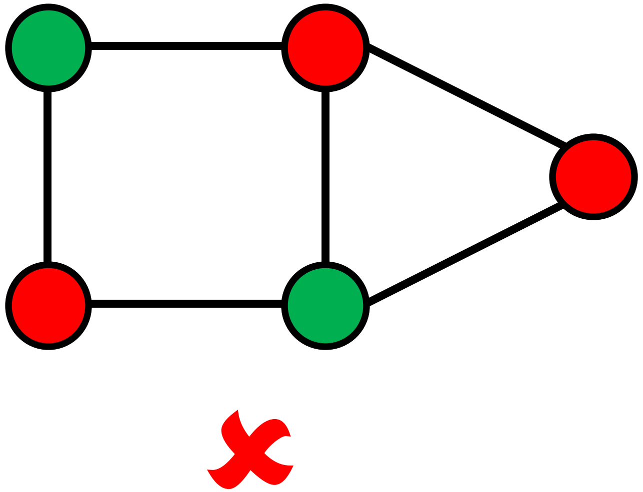
Family of graph problems that involve coloring vertices (or edges) subject to various constraints.



Graph Coloring

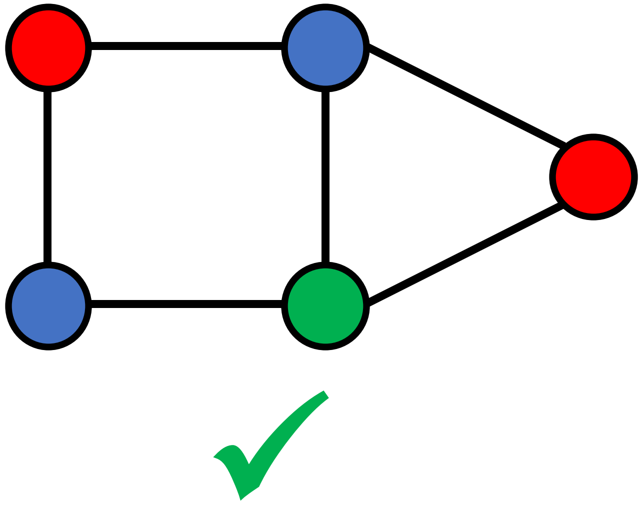
<u>Minimum Vertex Coloring</u>: Color the vertices of a graph using the smallest number of colors such adjacent vertices are different colors.

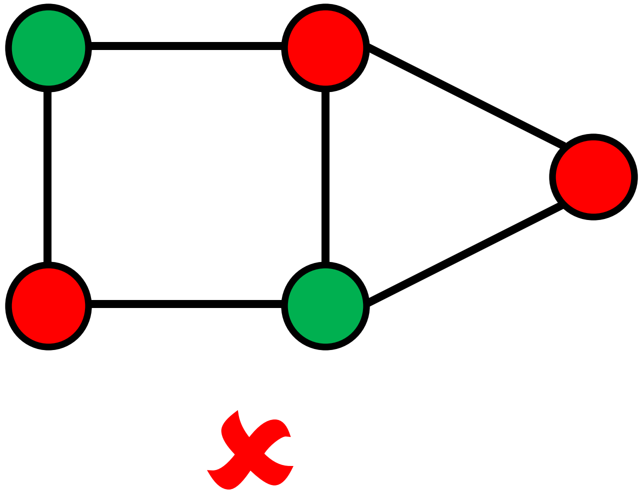




Graph Coloring

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

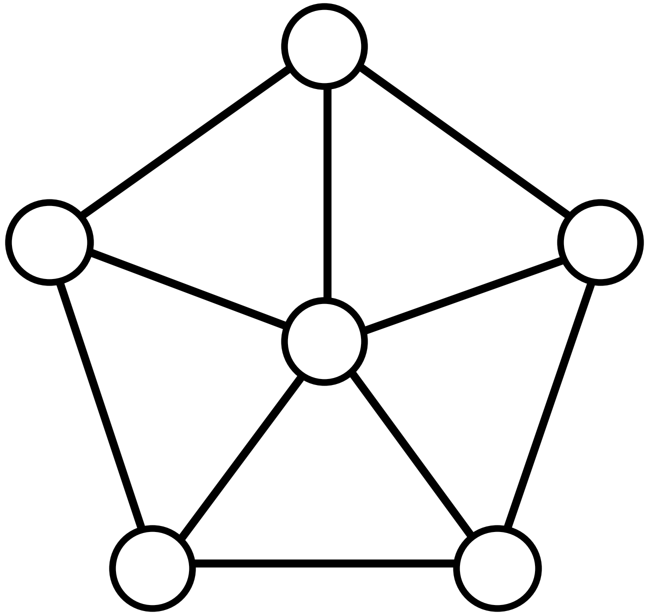




NP-Hard

Graph Coloring

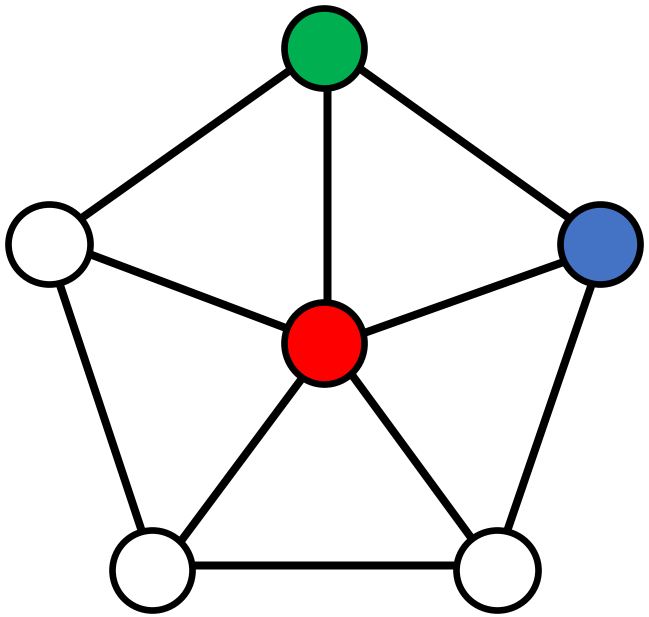
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors such adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

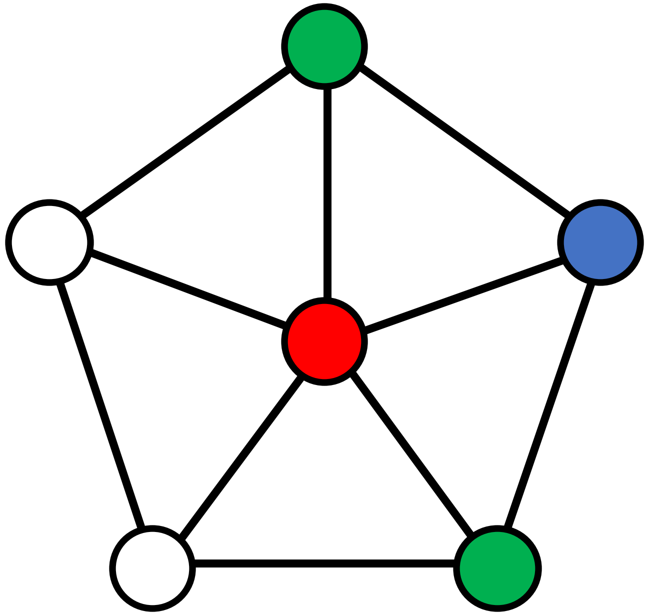
<u>Minimum Vertex Coloring</u>: Color the vertices of a graph using the smallest number of colors such adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

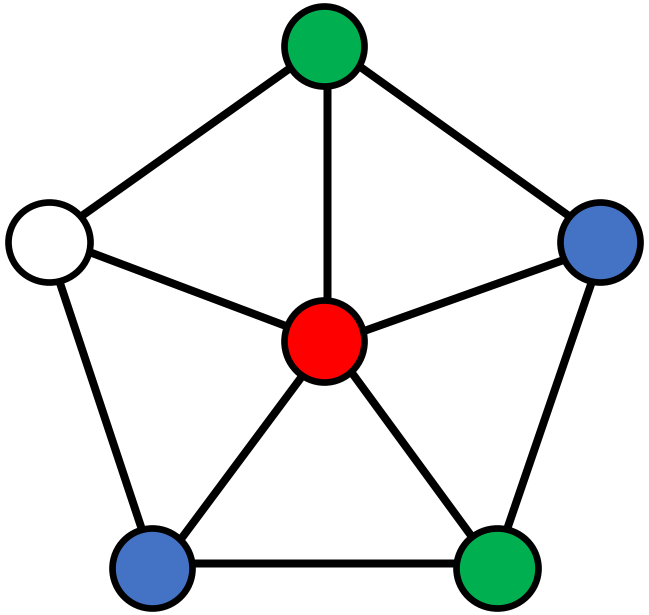
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors such adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

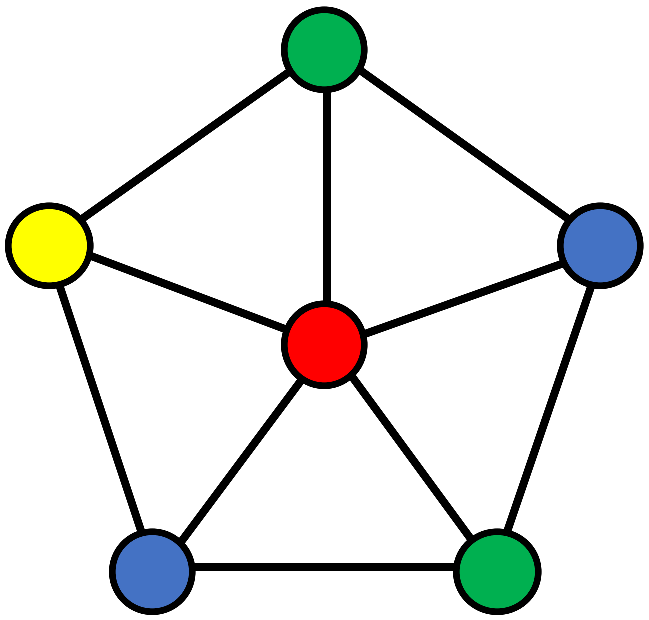
<u>Minimum Vertex Coloring</u>: Color the vertices of a graph using the smallest number of colors such adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

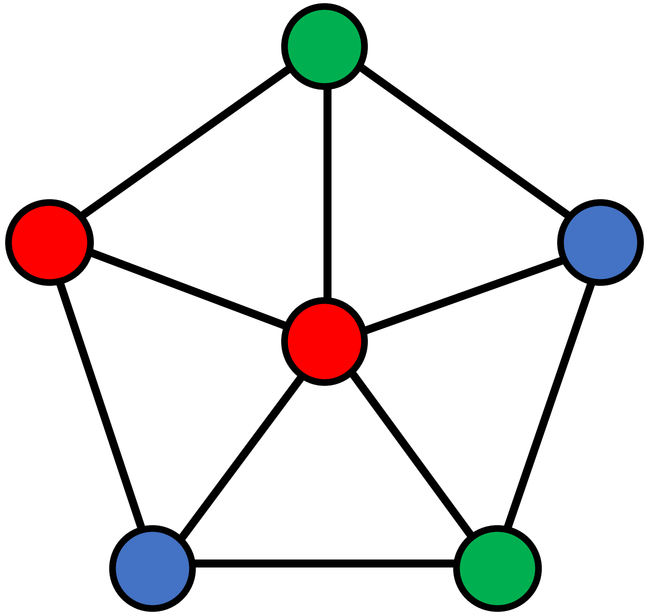
<u>Minimum Vertex Coloring</u>: Color the vertices of a graph using the smallest number of colors such adjacent vertices are different colors.



How many colors are needed to color this graph?

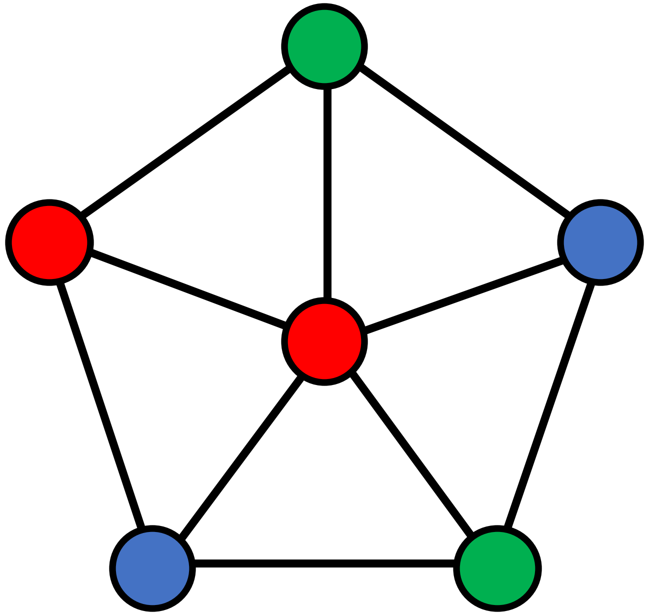
Max 3-Coloring

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

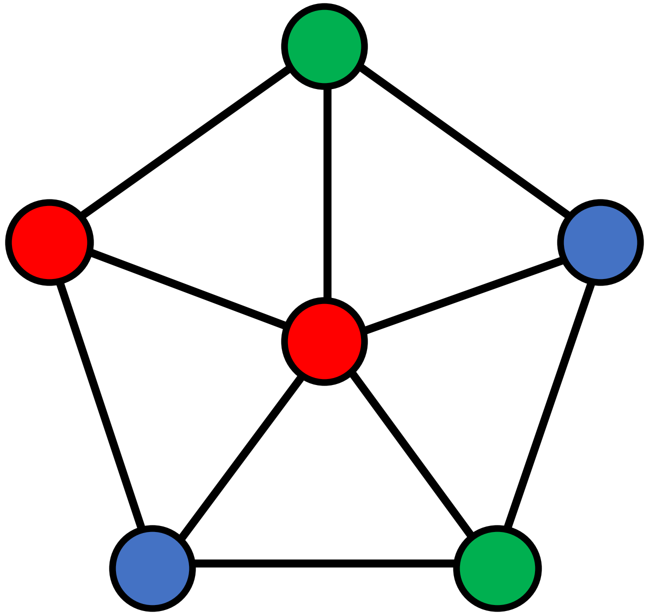
An edge in a graph is <u>satisfied</u> if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm?

Max 3-Coloring

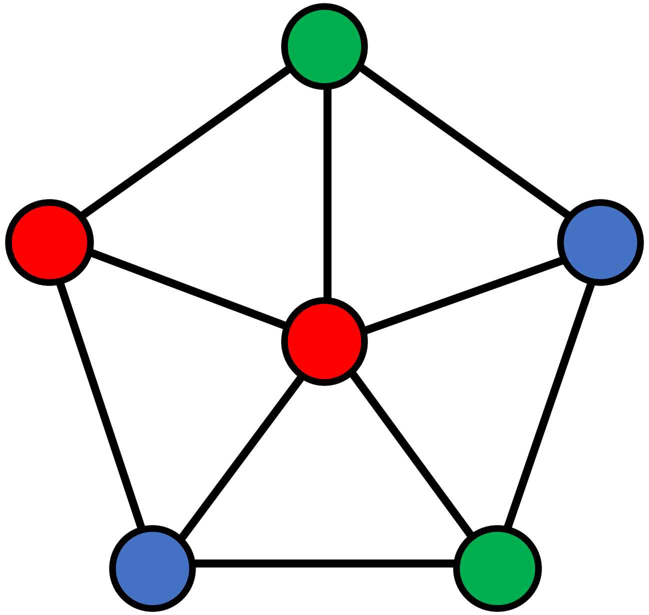
An edge in a graph is <u>satisfied</u> if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



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.

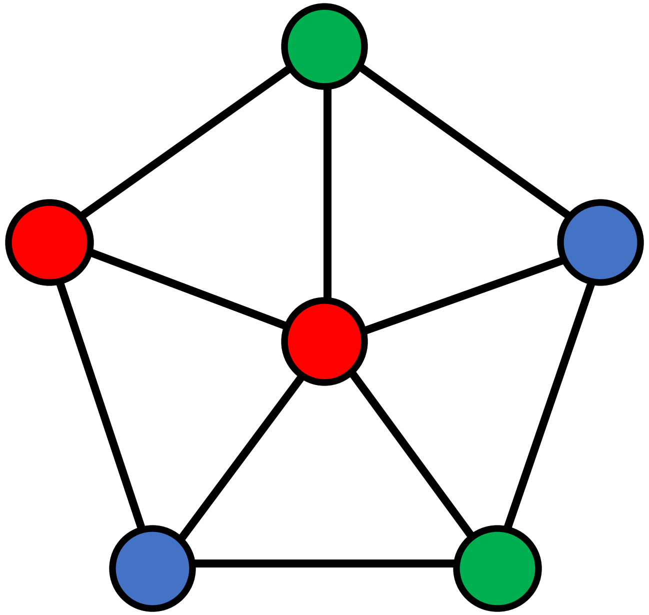


Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Running time?

Max 3-Coloring

An edge in a graph is <u>satisfied</u> 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.

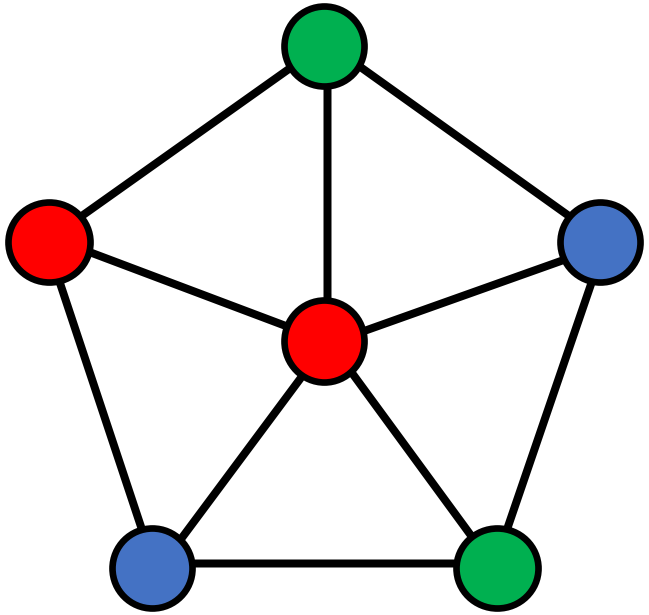


Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

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.

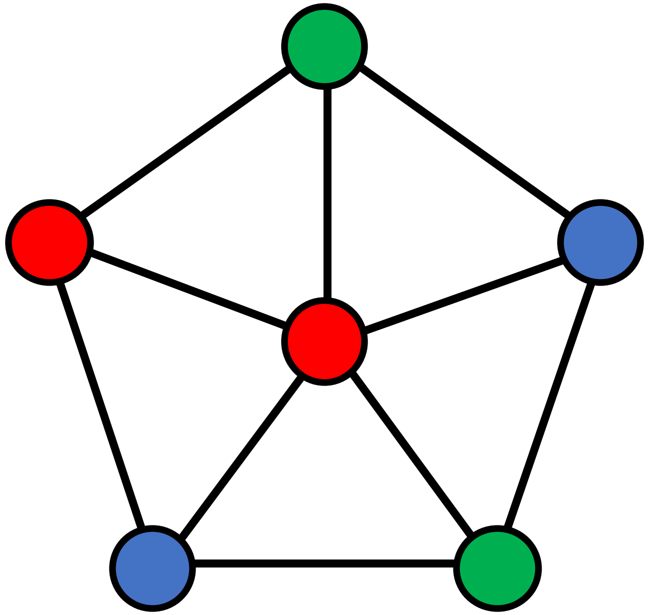


Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

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.

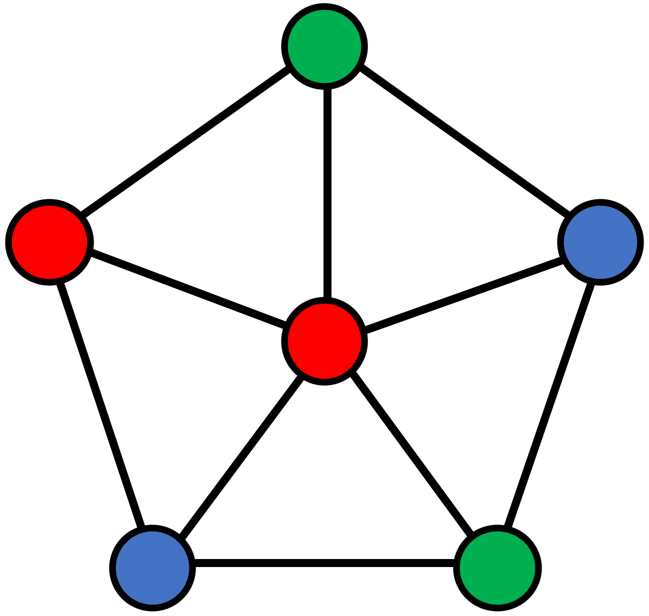


Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

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.

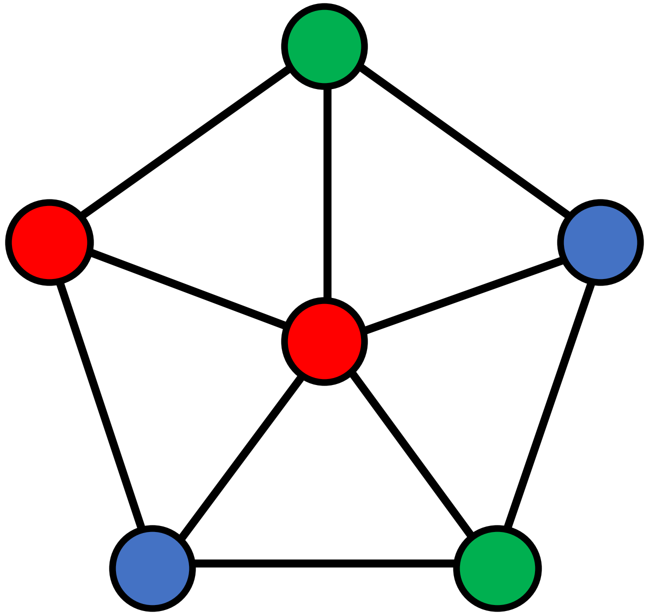


Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

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.

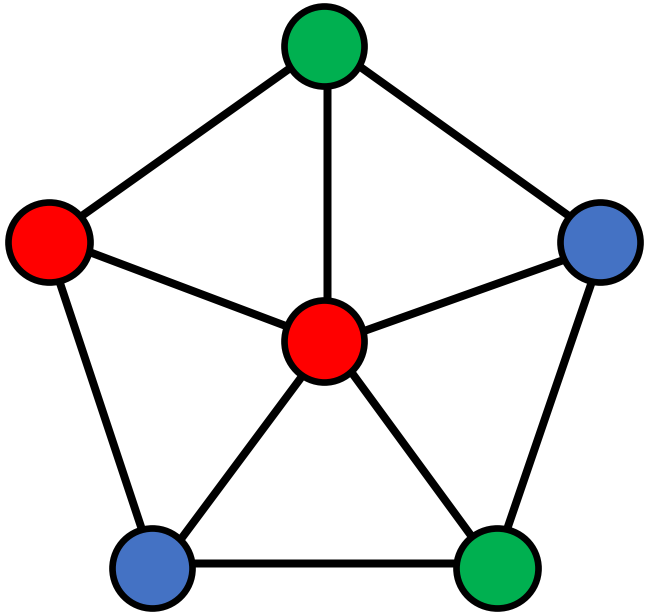


Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Running time? $O(n)$
Valid? Yes. Colors all vertices with ≤ 3 colors.
Optimal? Probably not…

Max 3-Coloring

An edge in a graph is <u>satisfied</u> 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.

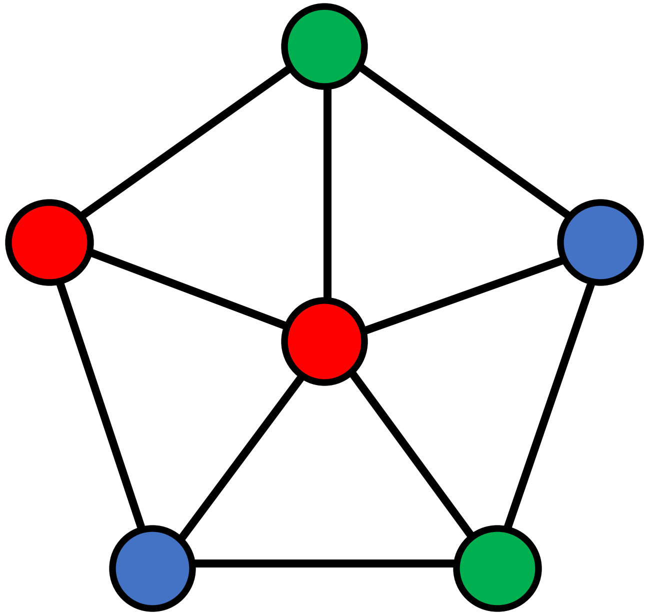
Running time? $O(n)$
Valid? Yes. Colors all vertices with $\leq$ 3 colors.
Optimal? Probably not…

Goal: Show this algorithm is expected to perform as a $\frac{3}{2}$-approximation algorithm.

Max 3-Coloring

Note: if problem is a maximization problem, $ALG \geq \frac{1}{\alpha} OPT$

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**.



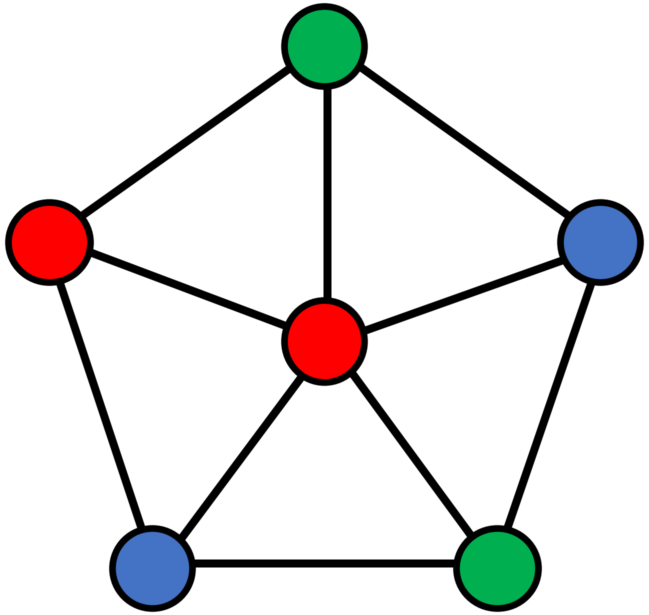
Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Running time? $O(n)$
Valid? Yes. Colors all vertices with $\leq$ 3 colors.
Optimal? Probably not…

Goal: Show this algorithm is expected to perform as a $\frac{3}{2}$-approximation algorithm.

Max 3-Coloring

An edge in a graph is <u>satisfied</u> 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 perform** as a $\frac{3}{2}$-approximation algorithm.

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)

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

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:

$\cancel{(b,b)}$, (b,g), (b,r), (g,b), $\cancel{(g,g)}$, (g,r), (r,b), (r,g), $\cancel{(r,r)}$

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

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)

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

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

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 $\sum_e X_e$?

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 $\sum_e X_e$?

ALG – It's the number of satisfied edges.

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 $\sum_e X_e$?

ALG – It's the number of satisfied edges.

What is $E[\sum_e X_e]$?

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

What is $\sum_e X_e$?

ALG – It's the number of satisfied edges.

What is $E[\sum_e X_e]$?

The most likely value for ALG.

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 $= E[ALG] = E[\sum_e X_e]$

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 $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

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

$E[X_e] =$?

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

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

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

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

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 $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

> 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 $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

$= \sum_e \frac{2}{3}$

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

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

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

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

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

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

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

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

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

Max 3-Coloring

Note: if problem is a maximization problem, $ALG \geq \frac{1}{\alpha} OPT$

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 $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

OPT?

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

Max 3-Coloring

Note: if problem is a maximization problem, $\boldsymbol{ALG \geq \frac{1}{\alpha} OPT}$

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 $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

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

Since $\boldsymbol{OPT \leq |E|}$ (can't have more satisfied edges than there are edges!)

SAT & 3–SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3–SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3–SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3–SAT

$$\phi = (x_1 \lor x_1 \lor x_2) \land (\overline{x_1} \lor \overline{x_2} \lor \overline{x_2}) \land (\overline{x_1} \lor x_2 \lor x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3−SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

Can you set the variables to **true** or **false** so that ϕ evaluates to **true**?

SAT & 3–SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

$x_1 = false$
$x_2 = true$

SAT & 3–SAT

$$\phi = \ (x_1 \lor x_1 \lor x_2) \land (\overline{x_1} \lor \overline{x_2} \lor \overline{x_2}) \land (\overline{x_1} \lor x_2 \lor x_2)$$

$$(F \lor F \lor T) \qquad (T \lor F \lor F) \qquad (T \lor T \lor T)$$

$x_1 = false$
$x_2 = true$

SAT & 3–SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

$$\downarrow \qquad\qquad\qquad \downarrow \qquad\qquad\qquad \downarrow$$

$$(F \vee F \vee T) \qquad (T \vee F \vee F) \qquad (T \vee T \vee T)$$

$$\downarrow \qquad\qquad\qquad \downarrow \qquad\qquad\qquad \downarrow$$

$$T \qquad\qquad\qquad T \qquad\qquad\qquad T$$

$x_1 = false$

$x_2 = true$

SAT & 3–SAT

$$\phi = (x_1 \lor x_1 \lor x_2) \land (\overline{x_1} \lor \overline{x_2} \lor \overline{x_2}) \land (\overline{x_1} \lor x_2 \lor x_2)$$

$$(F \lor F \lor T) \qquad (T \lor F \lor F) \qquad (T \lor T \lor T)$$

$$T \qquad T \qquad T$$

$$T$$

$x_1 = false$
$x_2 = true$

SAT & 3–SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

$$\begin{array}{ccc} \downarrow & \downarrow & \downarrow \\ (F \vee F \vee T) & (T \vee F \vee F) & (T \vee T \vee T) \\ \downarrow & \downarrow & \downarrow \\ T & T & T \\ \searrow & \downarrow & \swarrow \\ & T & \end{array}$$

$x_1 = false$
$x_2 = true$

$SAT = \{\langle\phi\rangle\text{: } \phi$ is a satisfiable formula$\}$
$3SAT = \{\langle\phi\rangle\text{: } \phi$ is a satisfiable formula with 3 variables per clause$\}$

Max 3-SAT

Given a 3-SAT instance, set variable values so that the number of satisfied clauses is maximized.

$$X = \{x_1, x_2, x_3, x_4\}$$

$$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (x_1 \vee x_3 \vee x_4)$$

{true, true, false, false} → true ∧ false ∧ true
{true, false, true, false} → true ∧ true ∧ true

Max 3-SAT

Given a 3-SAT instance, set variable values so that the number of satisfied clauses is maximized.

$$X = \{x_1, x_2, x_3, x_4\}$$

$$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (x_1 \vee x_3 \vee x_4)$$

{true, true, false, false} $\rightarrow$ true $\wedge$ false $\wedge$ true
{true, false, true, false} $\rightarrow$ true $\wedge$ true $\wedge$ true

Algorithm?

Max 3-SAT

Given a 3-SAT instance, set variable values so that the number of satisfied clauses is maximized.

$$X = \{x_1, x_2, x_3, x_4\}$$

$$(x_1 \lor x_2 \lor \overline{x_3}) \land (\overline{x_2} \lor x_3 \lor x_4) \land (x_1 \lor x_3 \lor x_4)$$

{true, true, false, false} $\rightarrow$ true $\land$ false $\land$ true
{true, false, true, false} $\rightarrow$ true $\land$ true $\land$ true

Algorithm: Assign $x_i =$ true with probability 1/2. Otherwise, set it to false.

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

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

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

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

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

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

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

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

$= \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

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

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

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

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

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

$= \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

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

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $=$?

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

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

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

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

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

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied = ?

All possibilities:

(T,T,T), (T,T,F), (T,F,T), (T,F,F)
(F,T,T), (F,T,F), (F,F,T), (F,F,F)

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

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

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

$= \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

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

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

All possibilities:

(T,T,T), (T,T,F), (T,F,T), (T,F,F)
(F,T,T), (F,T,F), (F,F,T), (F,F,F)

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

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

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

$= \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

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

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

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

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

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

$= \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

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

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Probability of **(F,F,F)**?

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

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

\# edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

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

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Probability of **(F,F,F)**?

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$

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

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

\# edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$
$$= \sum_e E[X_e]$$
$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

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

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Probability of **(F,F,F)**?

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$

Probability of anything but that?

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

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

\# edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

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

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Probability of **(F,F,F)**?

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$

Probability of anything but that?

$$1 - \frac{1}{8} = \frac{7}{8}$$

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

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

\# edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

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

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

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

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

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

$= \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

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

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

$E[Z_c] =$?

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

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

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

$= \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

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

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

$E[Z_c] = 1\left(\frac{7}{8}\right) + 0\left(\frac{1}{8}\right) = \frac{7}{8}$

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

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

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

$= \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

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

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

$E[Z_c] = 1\left(\frac{7}{8}\right) + 0\left(\frac{1}{8}\right) = \frac{7}{8}$

clauses we expect to satisfy $= E[ALG] = E[\sum_c Z_c]$

$= \sum_c E[Z_c]$

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

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

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$
$= \sum_e E[X_e]$
$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

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

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

$E[Z_c] = 1\left(\frac{7}{8}\right) + 0\left(\frac{1}{8}\right) = \frac{7}{8}$

clauses we expect to satisfy $= E[ALG] = E[\sum_c Z_c]$
$= \sum_c E[Z_c]$
$= \sum_c \frac{7}{8} = \frac{7}{8}|C|$

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

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

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

$$= \sum_e E[X_e]$$
$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

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

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

$$E[Z_c] = 1\left(\frac{7}{8}\right) + 0\left(\frac{1}{8}\right) = \frac{7}{8}$$

clauses we expect to satisfy $= E[ALG] = E[\sum_c Z_c]$

$$= \sum_c E[Z_c]$$
$$= \sum_c \frac{7}{8} = \frac{7}{8}|C|$$

$$\Rightarrow E[ALG] = \frac{7}{8}|C| \geq \frac{7}{8}\ OPT$$