Three-coloring an arbitrary finite graph in linear time with a very small constant

Miles Smith +1

Typically for three-coloring n represents the number of edges, the big O of the algorithm described here is actually O(v * e) where v is the number of vertices and e is the number of edges.*
***Ok at this point it might not be linear, but it is 100% polynomial with the factor being <= 3. My guess is that for k coloring, it will end up being $O(n^k)$, but that's an issue for future me.*

Notation:

We will define certain properties of the graph, such as the number of vertices and edges of a given graph g to be v and e respectively.
Vertex(ices) v and node(s) n are interchangeable with no distinction.
Edge(s) e and connection(s) $unused$ are interchangeable with no distinction.
The Chromatic number of a graph as defined below is represented with the symbol k

Definitions

Graph

An integer v, representing number of vertices and a list of pairs of integers with length e representing the connection between any two vertices. A vertex cannot connect to itself, and the connections from $n_a \rightarrow n_b$ and $n_b \rightarrow n_a$ are considered the same. This is more commonly known as an undirected graph.

Finite Graph

A graph that has a finite number of nodes.[1]

[1] If, from our definition of a graph, the graph has a finite number of nodes, n, then the number of possible edges, e, said graph can have is represented by the following equation e <= n(n-1)/2.

Connected Graph

A finite graph such that from any given node to any other node there is at least one path that connects the two, potentially by using other nodes as intermediaries. *Consider all future mentioned graphs to be connected[2]and finite graphs.*

Chromatic Number

A $finite\ connected\ graph\ g$ is said to have $chromatic\ number\ k$, if and only if, the minimum length of a given set S, which contains unique elements, needed in order to assign an element of S to each vertex v_n in g such that no two vertices which share an edge have the same element of S assigned to them is of length k.

Fully Connected Graph

A graph such that from any given node there is a connection to every other node in the graph. *A fully connected graph is a connected graph.*

Determinate Graph (3 colorable)

A graph g is said to be determinate (under 3 colors) iff g is 3 colorable and when given the color of any 2 nodes that share an edge, then there is exactly one permutation of the graph that is a valid 3 coloring. A conclusion can be made here that for any determinate graph that is 3-colorable, every node will have at least two neighbors that are directly connected.

True Graph

The fully connected graph with k nodes. *This graph is also determinate.*

Uncolored graph

A graph with an unknown chromatic number.

Base 2 Representation

If a number is represented in base 2, then it will be proceed by $0x$... for example

$$4 = 2^2\ (base\ 10) = 0x100\ (base\ 2)$$

[2] The algorithm I developed can actually detect if the input is a single graph or multiple disconnected graphs and can find if each is three-colorable in the same time.

Hamming Weight

The hamming weight of a number is the number of digits that are equal to one when the number is represented in base 2.

Bitwise operators

As I am a Programmer first and mathematician second, the use of bitwise operators on base 2 numbers is significantly easier for me to describe than sets and operations on sets. I will define some useful bitwise operators below that we will use throughout the paper. These are operators that can be described as functions that perform the same operation on each digit in a base 2 number. The operator names and their symbols are described below.

Not ^

The ^ operator takes in one argument and for every bit changes $1 \rightarrow 0$ and every $0 \rightarrow 1$

Input (base 2)	Input (base 10)	Output (base 2)	Output (base 10)
0x001	1	0x110	6
0x010	2	0x101	5
0x100	4	0x011	3
0x111	7	0x000	0

The ^ operator is its own inverse

$$X = \hat{}(\hat{}X)$$

AND &

The & Operator takes in 2 arguments and produces a 1 in the i^{th}place if both of the arguments have a 1 in the i^{th} place.

X1 (base 2)	X2 (base 2)	X1 (base 10)	X2 (base 10)	Y (base2)	Y (base 10)
0x001	0x010	1	2	0x000	0
0x111	0x101	7	5	0x101	5
0x110	0x011	6	3	0x010	2
0x000	0x111	0	7	0x000	0

OR |

The | operator takes in 2 arguments and produces a 1 in the i^{th} place if either of the arguments have a 1 in the i^{th} place. This will still produce a 1 in the i^{th} place if both arguments have a 1 in the i^{th} place.

Color State

We define nodes as having a color state. That is one of potentially many values combined. The maximum number of possible states is equal to 2^kwhich in our case is 8. A color state is defined by a number in binary with $k = 3$ bits. A node's color state s is said to be **collapsed** if and only if there is some nonnegative integer x that satisfies the following.

$$(base\ 10)\ s\ =\ 2^x\ where\ x\ is\ an\ integer\ and\ x\ \geq\ 0$$

In other words if the Hamming Weight is exactly 1, then the node's color state is said to be collapsed. If the Hamming Weight of a given node is > 1, then we say the node's state is a $superposition\ s$ of all colors that satisfy the following:

$$Color\ \&\ s\ \neq\ 0x000$$

Color Definitions

We will be showing a polynomial time algorithm with 3 colors, which will be defined as such.

$$RED\ =\ 0x001$$
$$GREEN\ =\ 0x010$$
$$BLUE\ =\ 0x100$$

And the following hamming weight = 2 superpositions as follows

$$ANTIRED\ =\ \hat{}RED\ =\ 0x110$$
$$ANTIGREEN\ =\ \hat{}GREEN\ =\ 0x101$$
$$ANTIBLUE\ =\ \hat{}BLUE\ =\ 0x011$$

And the following hamming weight = 3 super position as

$$ANY\ =\ RED\,|\,GREEN\,|\,BLUE\ =\ 0x111$$

And the following constant

$$ERROR\ =\ 0x000$$

A graph is not three-colorable if and only if during our propagation phase and node's color state is $0x000$

Correlations? Connections? Axioms?

1. Values of colors are interchangeable - Given a graph g which chromatic number k, then the graph g' created by replacing every instance of color C_a with color C_b and vice versa also has chromatic number k and is validly colored.
    a. In a determinate graph you must replace every instance of C_a with color C_b
    b. In an indeterminate graph where there are more than one valid three coloring permutation you may replace every instance, or only the ones required to maintain the rules of three colorability.
2. Given a valid coloring of a graph with chromatic number k, every node must have a color that is a member from a set of colors with length k.
3. No two nodes that share an edge may be the same color.
4. The choice of what colors, or more accurately, labels will be in the set of size k of which we assign to nodes does not matter.
    a. Likewise the permutation of the ordering of elements in the set of “colors” to be assigned does not matter.
5. If a graph g has n nodes, then it is n colorable
    a. Note, this does not mean it has chromatic number n. This means $k \ <= \ n$
6. If a fully connected graph g has n nodes, then it’s chromatic number is n and g is determinate.

Theorems

Edge Reduction Theorem

Let $g\ be\ a\ determinate\ graph\ that\ is\ k\ colorable$
Let $S\ be\ the\ set\ of\ unique\ colors\ in\ g$ with $|S|\ =\ k$
Let $n'\ be\ a\ node\ not\ in\ g$
Let $S_0\ be\ the\ set\ of\ unique\ colors\ of\ nodes\ that\ are\ neighbors\ to\ n'$
Add connections from n' to g such that the following is true
$|S|\ =\ \left|S_0\right| + 1$
$S_0\ is\ a\ subset\ of\ S$
Let $count(S_{0x})$be equal to the $number\ of\ nodes\ connected\ to\ n'\ that\ share\ the\ color\ x$

The Edge reduction theorem states that - the graph's deterministic property will not change so long as for every x, $count(S_{0x})\ >\ 0$, and neither will its chromatic number k remains the same.

1st attempt at putting it into words.

If I have a $determinate\ graph\ g$ that is $k\ colorable$ and I want to add a $new\ node\ n'$ that is neighbors with exactly $k\ -\ 1$ unique colors while the resulting $graph\ g'$ remains $k\ colorable$, then will there be any difference if { for each color $C_x\ such\ that\ \ 0\ \leq\ x\ <\ k$ I connect the new node to exactly 1 node for each color C_x except a single color C_0, and exactly 0 connections to nodes with color C_0. } versus { for each color C_x| $0\ \leq\ x\ <\ k$ I connect the new node to more than 1 node which all have color C_x except for a single color C_0, and exactly 0 connections to nodes with color C_0 } - No!

Remember a determinate graph has exactly 1 validly k-colored permutation when given k-1 uniquely colored nodes that are connected to a single node.

Therefore, for any given color that the new node is neighbors with, all except for one of those connections is redundant, and removing them results in a graph that is still determinable, and has the same number of valid k colored permutations = 1 and has the same chromatic number.

This is only true because the entire original graph is determinate!

We expand this to non determinate graphs and state that the graph is still validly colored for k colors, however once removing the connections to a non determinate graph, the chromatic number need not remain the same but can only decrease.

Node Reduction Theorem

Given $graph\ g$ that has $chromatic\ number\ k$ and $number\ of\ nodes\ >\ k$ for any $node\ n$ that has less than k neighbors, if we remove n then the resulting $graph\ g'$ also has $chromatic\ number\ k$.

We show this with the opposite of the Edge Reduction Theorem. If the original graph in the Edge Reduction Theorem has the exact same chromatic number as the produced graph, then the reverse process must be the same(specifically where the number of nodes in the graph is larger than the chromatic number of the graph, if they are equal, then the chromatic number is reduced by removing a node).

Graph Reduction Theorem

1. Given any $graph\ g$ with chromatic number k if there is a $determinate\ sub\ graph$ g_0 of g with $chromatic\ number\ k_0\ and\ number\ of\ nodes\ n_0 > \ k_0$, then there is a $reduced\ graph\ g'$ with $number\ of\ nodes\ \ n' \ =\ n\ -\ n_0\ +\ k_0$ and chromatic number $k'\ =\ k$ that results from replacing g_0 with a $true\ graph\ T$ that has $n_T = k_0\ nodes.$
2. $k = k_0 = k'$ is a valid reduction

Graph Reduction "Proof"

The graph g' is formed by the following operations

1. Create a disconnected true graph T with $n_T\ =\ k_0$ number of nodes.
3. Assign each node to one of k_0 colors.
4. For each connection between a node n_g in g that has a connection to a node n_{g0} in g_0 where n_g is not a node in g_0 remove the connection, then add a connection to the node n_T in T that has the same color as the color of n_{g0} if there is not already a connection between $n_g\ and\ n_T$.
5. Discard the now disconnected subgraph g_0 and recognize that the chromatic number could not have changed, and the number of valid k-colorable permutations of g has not changed.

Graph Coloring via Propagation

Take some uncolored $graph\ g$. How might we determine if it is *three-colorable?* Based on [2] we know that every node must be one of the colors in our set of available colors, which is $S = \{RED,\ GREEN,\ BLUE\}$. So let's try coloring the graph one node at a time. Take some $node\ n_0$ and based on [6] our definitions of colors, and which color a node is doesn't matter at an individual node level, so let's declare $color(n_0) = RED$. Great, we've done one node, now let's take some other $node\ n_x$ that may or may not share an edge with n_0, What should we color this node? Based on the information we have so far, we have no idea if coloring n_x any of our colors would result in a valid graph. So let's be more smart about this.

Let's restart, and this time we will initialize all of the node's to the $color\ state$ ANY. And now, when we color a node, let's perform the & operator on the ^ of our chosen color, on all of the neighbors of the chosen node. So if we take some node n_0 and assign it the color RED then we would say that for every neighbor x of n_0 it's color now becomes

$$[1]\ Equation\ of\ Propagation\ \ n_x = n_{x0}\ \&\ (\hat{}n_0)$$

$$n_x = ANY\ \&\ (\hat{}n_0)$$

$$n_x = ANY\ \&\ ANTIRED$$

$$n_x = ANTIRED = GREEN \mid BLUE$$

Now we have a pretty neat relationship here. We've colored one node and propagated its consequences to all of its neighbors. We also have a clear and easy method for detecting if a graph is validly colored. If at any point during propagation a given node's color state is equal to 0, then the graph can't be validly colored (given our previous choices). Now, let's begin to demonstrate how to only make choices that don't matter.

Based on [1] our first choice is not important, so how can we continue to make choices that represent all possible permutations without actually calculating all possible permutations? If the given state represents the possible colors to collapse a given node to, then consider two nodes $n_1\ and\ n_2$ with $superposition\ state\ s_1\ and\ s_2$ both of which $hammingWeight(s_1) = hammingWeight(s_2) = 2$ and share a connection. We find from [3] that if we collapse one node, then the other collapses as well. We can consider these two nodes as $entangled$. Now consider another node n_x with

$superposition(n_x) \; = \; 0x111$ and has a connection to both n_1 and n_2. What can we tell about this node? Well we see that since it is connected to both of them, it can't be the same color as them and since both n_1 and n_2 must be one of two colors, but not each other, then we must conclude that in order for the graph to be three-colorable that the only valid color for n_x is the same as n_0. If we find any nodes like this we can remove them from the list of nodes in $superposition \; = \; 0x111$ and propagate the consequences to its neighboring nodes and put n_x into the propagated list. Without looking too much into how the practical upper limit is lower, we can search through every node (in the $0x111$ state) and then check if any 2 of it's neighbors share a connection. Assuming a $node\ n_x$ has $y\ neighbors$ the time complexity to find such nodes is $O(y(y - 1)/2) \; \approx O(y^2)$ and if we consider worst case every node is connected to every node, then every node has n neighbors we get $O(n^2)$ as a theoretical maximum time to find one node to collapse.

Note that in three-colorable graphs this is impossible for EVERY node, and so the practical upper bound is much less than O(n^2)

Now consider the fact that we will only continue this process of searching for neighbors if and only if we found at least one node that satisfies those conditions. So there is another theoretical upper limit of performing this operation $n - 2$ times where n is the number of nodes in the graph. So in a theoretical upper limit of $O(n^3)$ time we have found every node of a given color in a determinate graph with $n\ nodes$.

At this point it should be obvious to recognize that this method of propagating colors will correctly color any determinate graph if it is three-colorable.

Expanding to indeterminate graphs

Up until now we've only been working in the realm of graphs that have exactly one valid permutation of colors once two neighboring nodes have been connected. Now we're going to show that any choice we make does not affect if we will be able to three-color a given graph.

First we must realize that an indeterminate graph can but is not required to contain determinate subgraphs with $chromatic\ number \; = \; 3$, and that via the *graph reduction theory* and our algorithm we can accurately color any determinate subgraph.

After we have finished coloring the subgraph, any nodes that are in a superposition of multiple states must have one of those colors given the chosen configuration of the determinate subgraph in our final coloring. The entire graph is colorable if and only if we can finish propagating the colors throughout the graph without running into a conflict based on our "choice" in the subgraph. As we have defined a determinate graph to have exactly one permutation of validly three-coloring, given two neighbor nodes with different colors, then we can conclude that the determinate graph's configuration never needs to be touched propagated again and since we only propagated/visited each node in the sub graph once, this is a linear process to propagate a linear amount of nodes.

We can restrict the Node Reduction Theorem to k=3 and see it obviously holds true.

Just need to wrap it all together now.
Cyclical graphs
Chain graphs
Multidimensional graphs
It works for all of them.

I don't want to lose this stuff, but it's not in a very organized manner.

if our given node does not connect to a determinate subgraph, then we will show in the following ways that choosing the color of any of its neighbors does not affect the colorability of the graph. First, we showed in the previous section that once any node that is contained in a determinate subgraph has been colored, then we can color the entire subgraph before looking at the whole graph and have not reduced the colorability of the graph.

Consider the following determinate graph.

Now consider adding a new node n to the graph via exactly one connection. How many possible options for the color of n? A: 2. Now consider 2 determinate three-colorable graphs that share a single node n. How many possible permutations are there that are still three-colorable? A: 4 times as many as a single determinate graph, which is 12. We find that for an indeterminate graph that contains x number of determinate graphs that share a single point, the number of possible valid three-coloring permutations to be $3(2^x)$.

Find now that for every graph that can be represented as a tree of three-colorable determinate subgraphs such that no nodes in sibling (cousins, ect) subgraphs connect and each subgraph is determinate, then the entire graph is three-colorable with possible permutations dependant on the number of subgraphs (nodes in the subgraph tree) in the tree.

When we collapse a neighbor we are choosing one of its possible states to become, forcing it to no longer be in a superposition. The value we choose for a collapsed node must satisfy the following equation

$$[2]\ Equation\ of\ Collapsion\ C\ =\ C_0\ \&\ C\ where$$

1. $C\ is\ the\ state\ of\ the\ node\ after\ collapsion$
2. $C_0\ is\ the\ state\ of\ the\ node\ prior\ to\ collapsion$
3. $C\ is\ a\ value\ that\ in\ base\ 2\ has\ hamming\ weight\ =\ 1$