ACYCLIC EDGE COLORING OF GRAPHS

By:-
Lavish Kothari
15IS12F

CONTENTS

- Terminologies related to coloring of graphs
- Notations used
- Definitions
- Theorem (Problem statement)
  - Related work done in this field
  - Examples
- Operations
- Proof of the Theorem
- Implementation Details
  - Algorithm
  - Code Details
- Demo
- Acyclic Edge Coloring of graphs with $\Delta \leq 5$
- Future Enhancements

Terminologies related to coloring of Graphs

- **Proper Edge Coloring of Graph**

  A Proper Edge Coloring of G=(V,E) is a map c : E→C (where C is the set of available colors) with $c(e) \neq c(f)$ for any adjacent edges *e*, *f*.

- **Edge Chormatic Index**

  The minimum number of colors needed to properly color the edges of G, is called the Edge Chromatic Index of G and is denoted by $\chi'(G)$.

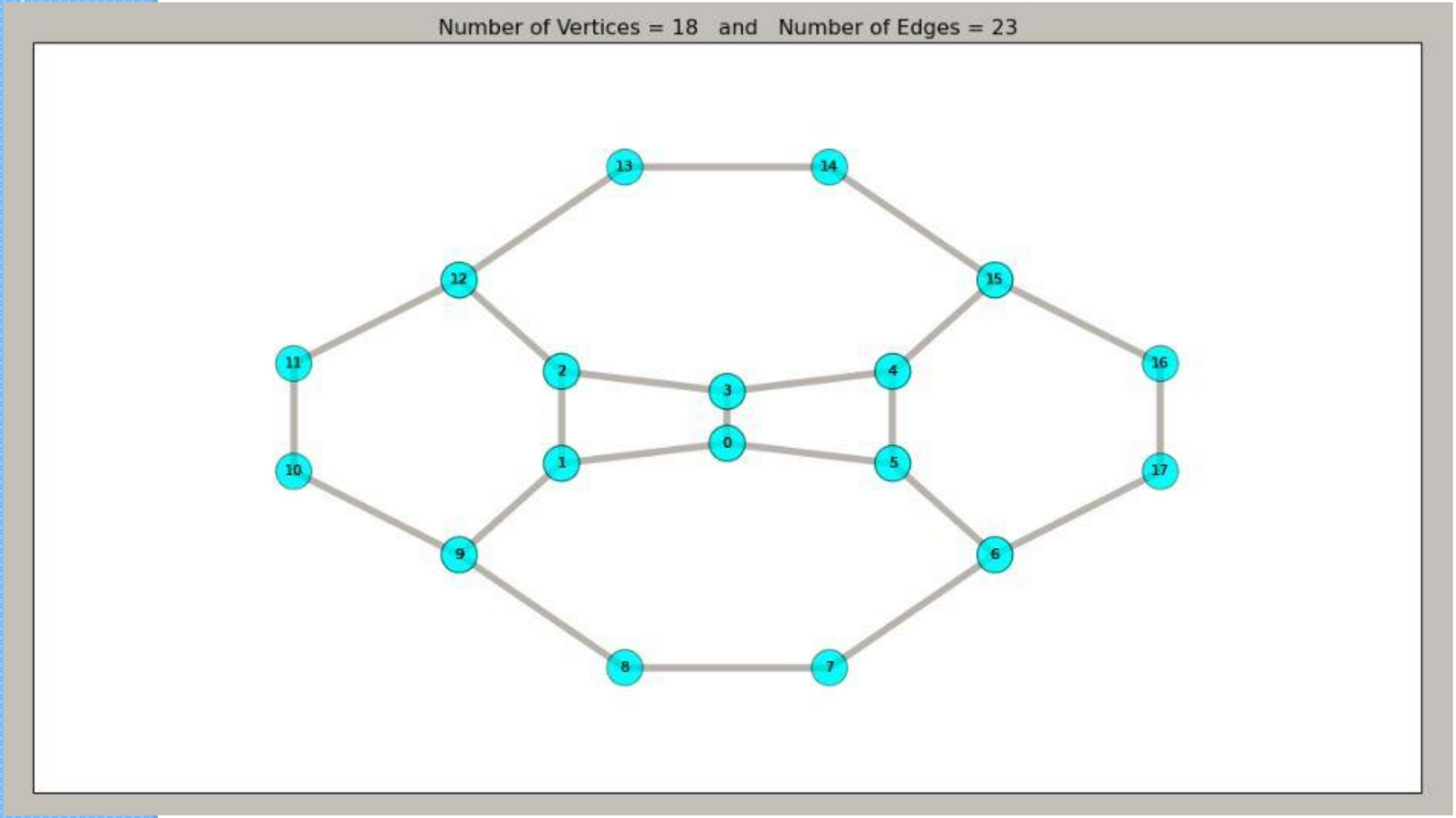
- **Acyclic Edge Coloring Graph**

  A proper edge coloring c is called Acyclic if there are no bi-chromatic cycles in that graph.

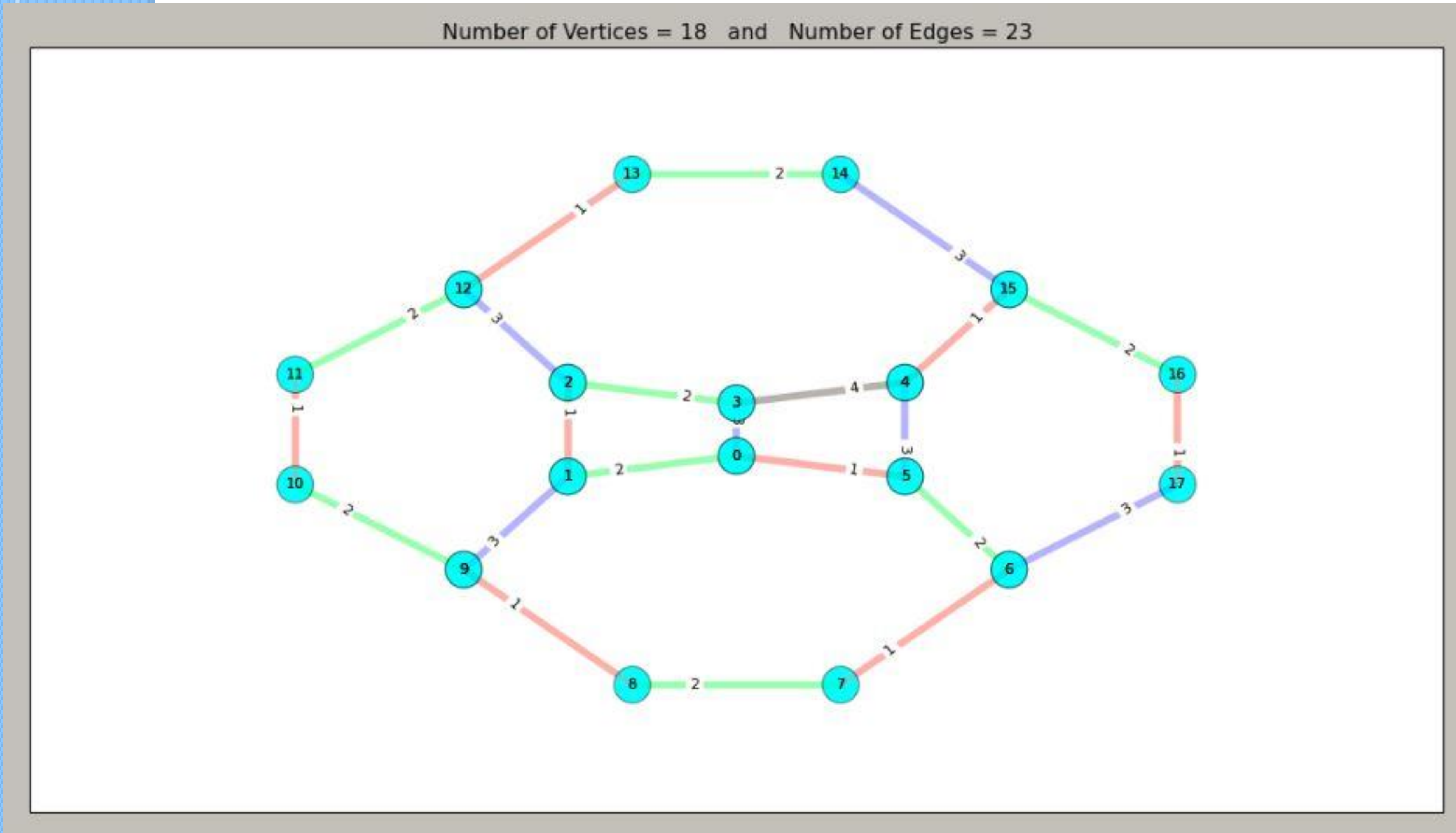
- **Acyclic Edge Chromatic Number**

  The Acyclic Edge chromatic number (also called Acyclic Chromatic Index) denoted by $a'(G)$, is the minimum number of colors required to acyclically edge color G.

Example (without Edge Coloring)



Example (with Edge Coloring)



Acyclic Edge Coloring Conjecture

Conjecture by Alon, Sudakov, and Zaks (and independently by Fiamcik)

For any Graph G,

$$a' \leq \Delta(G) + 2$$

where Δ represents the maximum degree amongst all the vertices.

It has been proved by Esperet and Parreau, that for graph with maximum degree Δ, $a'(G) \leq 4\Delta - 4$, and improved by Giota to $a'(G) \leq \lceil 3.74(\Delta-1) \rceil + 1$

Basics and Notations

- **$G = (V, E)$** such that **$E \subseteq V \times V$**
  where E is the Edge set and V is the Vertex Set.
  We usually denote
  $|V| = n$
  $|E| = m$
- **$\delta(G) = \min \{\deg_G(v) \mid v \in V(G)\}$**
- **$\Delta(G) = \max \{\deg_G(v) \mid v \in V(G)\}$**
- **$N_G(u)$** = neighbors of vertex u in G.
- For $e \in E$
  G-e denotes the graph obtained by the deletion of edge e.
- A coloring c, of a graph is denoted by **$c : E \rightarrow \{1,2,3\ldots.k\}$**
  c(e) denotes the color given to the edge e with respect to the coloring c.
- For any vertex u, **$F_u(c) = \{ c(u, z) \mid z \in N_G(u) \}$**
- For a, b $\in$ V, **$S_{ab}(c) = F_b(c) - \{c(a, b)\}$**

(α, β ,a , b) Maximal Bichromatic Path

Let a,b ∈ **V,** The (α, β ,a , b) Maximal Bichromatic Path is a maximal path that starts at vertex a with an edge color α, and ends at b.

Note:

- The edge of (α, β ,a , b) Maximal Bichromatic Path incident on a is colored α.
- The edge of (α, β ,a , b) Maximal Bichromatic Path incident on b can be either coloured as α or β.
- (α, β ,a , b) and (α, β ,a , b) Maximal Bichromatic Paths have different meaning.
- Maximal Bichromatic Path have at least 2 edges.

(α, β ,ab) Critical Path

Let a, b $\in$ **V**, ab $\in$ **E** and c be the partial coloring of G**,** then (α, β, a, b) Maximal Bichromatic Path which starts out from the vertex a via an edge colored α and ends at vertex b via an edge colored α, is called (α, β, ab) Critical Path.

Note:

- Every Critical Path will be of odd length.
- For a critical path, the smallest length possible is 3.

Candidate Color

A color $\alpha \neq c(e)$ is a candidate color for an edge e in G with respect to parital coloring c of G if none of the adjacent edges of e are colored α.

Valid Color

A candidate color α is valid for an edge e if assigning the color α to e does not results in any bichromatic cycles in G.

Some Important Observations

- Given a pair of colors α and β of proper coloring c of G, there can be at most one maximal (α, β) bichromatic path containing a vertex v with respect to c.

- A candidate color of an edge e = uv is valid if

$$\mathbf{F_u \cap F_v - \{ c (u, v)\} = (S_{uv} \cap S_{vu}) = \emptyset}$$

- Let c be a partial coloring of G, a candidate color β is not valid for the edge e = (a , b) if and only if ∃α ∈ $S_{ab} \cap S_{ba}$ such that there is a (α, β, ab) critical path in G with respect to coloring c.

Operations Used

1. Color Exchange

Let c be a partial coloring of G. Let $u, i, j \in V(G)$ and $ui, uj \in E(G)$.

We define Color Exchange with respect to the edge ui and uj as the modification of the current partial coloring c by exchanging the colors of ui and uj to get a partial coloring c'.

$c' (u , i) = c (u , j)$
$c' (u , j) = c (u , i)$
$c' (e) = c (e)$ for all other edges e in G.

The above color exchange is denoted by
$c' = \text{ColorExchange} (c, ui, uj)$

Operations Used

2. Recolor

We define $c' = \text{Recolor}(c, e, \gamma)$ as the recoloring of the edge e with a candidate color γ to get a modified coloring c' from c, i.e., $c'(e) = \gamma$ and $c'(f) = c(f)$, for all other edges f in G.

- The recoloring is said to be proper, if the coloring c' is proper.
- The recoloring is said to be acyclic (valid), if in coloring c' there exists no bichromatic cycle.

Lemma

- Let c' be the partial coloring obtained from a valid partial coloring c by the color exchange with respect to the edges ui and uj.

  The partial coloring c' will be proper if and only if the following two conditions are true.

  - $c(u, i) \notin S_{uj}$
  - $c(u, j) \notin S_{ui}$

Acyclic Edge coloring of Sub-Cubic Graphs

A Graph is called **sub cubic** if the maximum degree of that graph is 3.

Theorem

Let G be a non-regular connected graph of maximum degree 3, then $a'(G) \leq 4$

Note:

- If $\Delta(G) < 3$ then $\boldsymbol{a'(G) \leq 3}$

Proof of the Theorem

Induction on number of Edges:

Base Case : Smallest possible edges on a non regular connected graph G of maximum degree 3 on n vertices is n-1. Then G is a tree and is acyclically colorable using 3 colors. (As there will be no cycles, we don't need to worry about the acyclicity of edges. Every proper edge coloring of a tree will be acyclic.)

Induction Hypothesis :

Let G be a

- Connected graph
- Non-regular Graph
- $\Delta(G) = 3$
- $m \geq n$

m = number of edges

n = number of vertices

Let the theorem be true for all non regular connected graphs with maximum degree 3 with at most m-1 edges.

Assumption

- Without the loss of generality we can assume that G is 2-connected.

So **$\delta(G) \geq 2$**

If there are cut vertices in G, then the acyclic coloring of blocks of G can be easily extended to G.

Proof Contd.

Since G is not 3 regular and **$\delta(G) \geq 2$.**

Then there is definitely a vertex of degree 2, let this vertex be x.

$\deg_G(x) = 2$

Let $y \in N_G(x)$

$G' = G - \{ xy \}$

G' is connected, since G is 2-connected.

Two Case

$\Delta(G') < 3$

Then

$a'(G') \leq \mathbf{3}$

$\Delta(G') \leq 3$

Then

$a'(G') \leq \mathbf{4}$

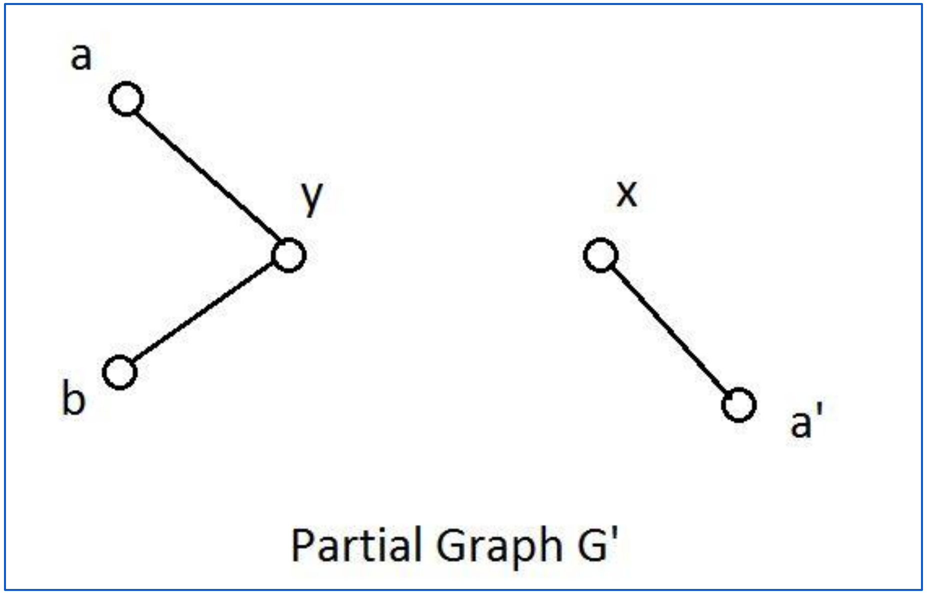
(by induction hypothesis)

Approach towards proving

We will try to extend the edge coloring of G' to G by giving a color to the edge xy from available 4 colors.

Since $|\, F_y \cup c(x, a')\,| \leq 3$,

So there is at least one candidate color for the edge xy.



Exhaustive Cases

Two Cases

$c(x,a') \notin F_y$

So all the candidate colors are valid for the edge xy

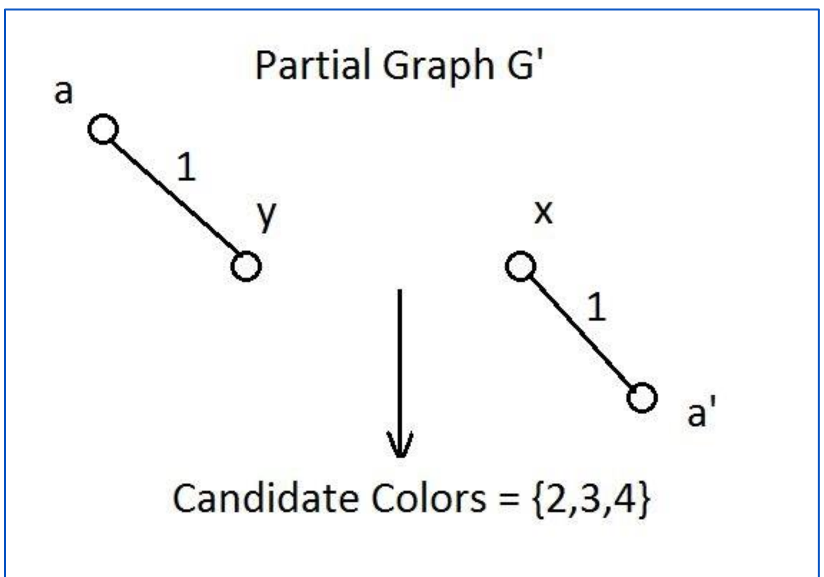
$c(x,a') \in F_y$

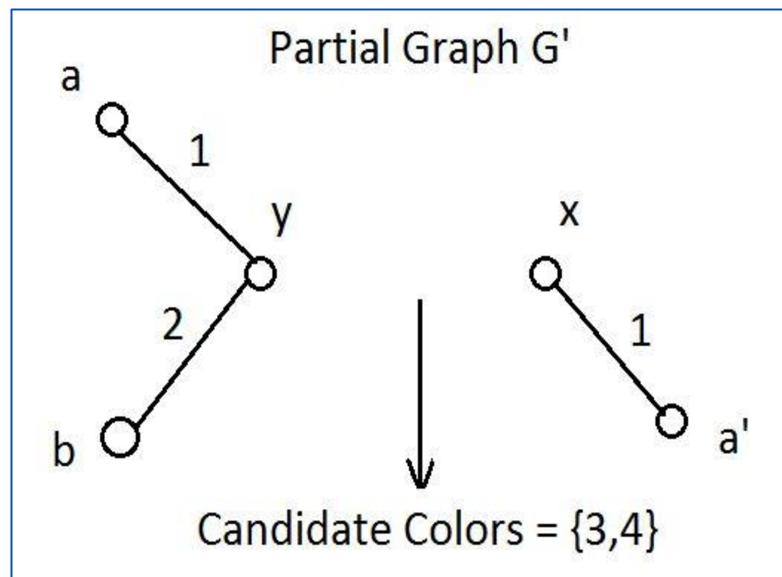
Cases contd.

$c(x\,,\,a') \in F_y$

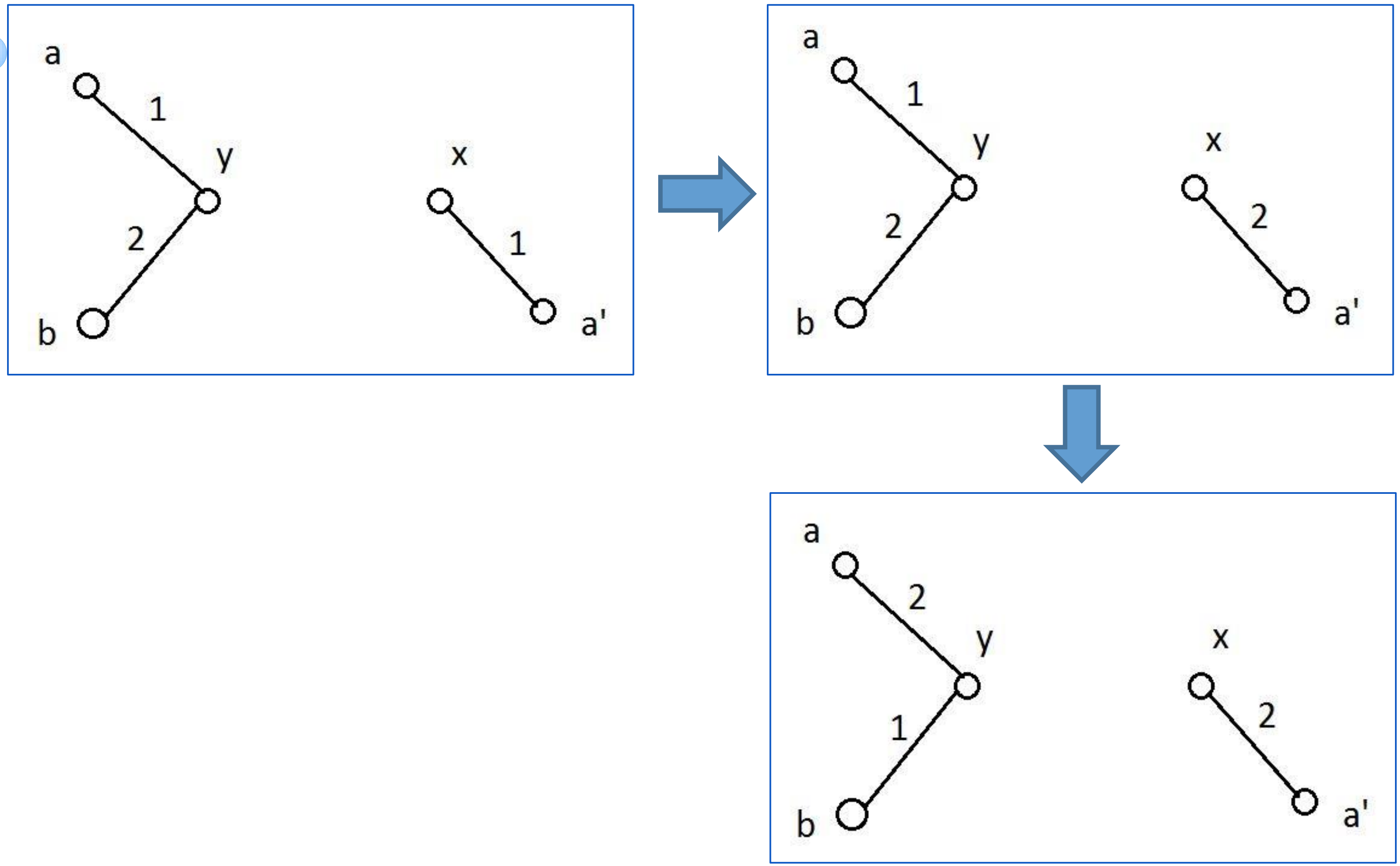
$|N_G{'}(y)| = 1$

$|N_G{'}(y)| = 2$





Cases contd.



Implementation Details

```
class Vertex
{
    public:
    int vertexNumber;
    list<int>adjacencyList;
};
```

```
class Edge
{
    public:
    int start,end,color;
    Edge();
    Edge(Edge const &);
    Edge(int,int);
};
```

Implementation Details

```
class Graph
{
    public:
    int numberOfVertices,numberOfEdges;
    Vertex* vertexArray;
    list<Edge> edgeList;
    set<int>C;

    Graph(int,int);
    void addEdge(int,int);
    void print();
    void deleteGraph();
    void arrangeEdges(); // very important thing. for description see the function definition.
    Vertex& findVertexWithDegreeAtMost2(bool*);
    void colorEdge(Edge&);
    Edge findEdge(int,int); // given vertexNumber number of two vertices, this function returns the ed
    int findColor(int,int); // given two vertices of graph this function returns its color.
    set<int> findCandidateColors(Edge const&); // finds the candidate colors for a given edge.
    set<int> S(int,int); // for S(x,a) this method returns the set of colors {c : c=color(a,b) for eve
    bool isCriticalPath(Edge const&e,int a,int b); // this method checks where there exists a (e,a,b)
    void recolor(Edge &e,int c); // recolors the edge e with color c.
    void colorExchange(Edge &e1,Edge &e2); // exchanges the color of edge e1 and e2.
    bool isInConfigurationA(Vertex const&,Vertex const&,Vertex const&,set<int>const&,set<int>const&);
};
```

Algorithm

- **Pick the edges in certain order.**
- The initial ordering of edges is necessary because if the edges are picked arbitrarily, then the edge may not fall at all into any case discussed before.
- So the edges should be picked in such a way that we always have a vertex (x) of degree 1 in G'.
- **How preferable edge ordering is decided?**
  - Pick an edge with at least one of it's endpoint with degree 2 or less, let us call this end point x.
  - Now give numbering to all the edges incident on x and then delete this vertex.
  - Repeat the above two steps and keep enumerating the edges incrementally.
- **How to use the current Edge Ordering?**
  - Pick the edges with decreasing edge number, so that each time you have the edge (to be colored) that falls in one of the discussed case.

Other Research Papers

- Graphs with $\Delta \leq 4$
  - Theorem : Let G be a connected graph on n vertices with m $\leq$ 2n-1 edges, and maximum degree $\Delta \leq 4$ then $a' \leq 6$.

Extending the concept for $\Delta \leq 5$

- Graphs with $\Delta \leq 5$
  - Let G be a connected graph on n vertices with $m \leq 2n-1$ edges, and maximum degree $\Delta(G) \leq 5$ then using the operations of Color Exchange and recolor I have shown that $a'(G) \leq 15$.

What I Learnt

- Basics of acyclic coloring
  - Vertex coloring and edge coloring
- Critical Paths and Maximal Bichromatic Paths
- Operations
  - Color Exchange
  - Recolor
- Implementation of graphs in C++ Standard Library using Object Oriented Programming concepts.
- Drawing Graphs using matplotlib and networkx library.

Future Enhancements

- Searching for some new Operations that could be easily used to prove the general conjecture For any Graph G, $\boldsymbol{a' \leq \Delta (G) + 2}$
- Extending the concept to prove that graphs with $\Delta \leq 5$ can be colored using at most 11 colors.
- Extending the program of sub-cubic graphs to implement program for graphs with $\Delta \leq 4$

Thank you