Graph Coloring

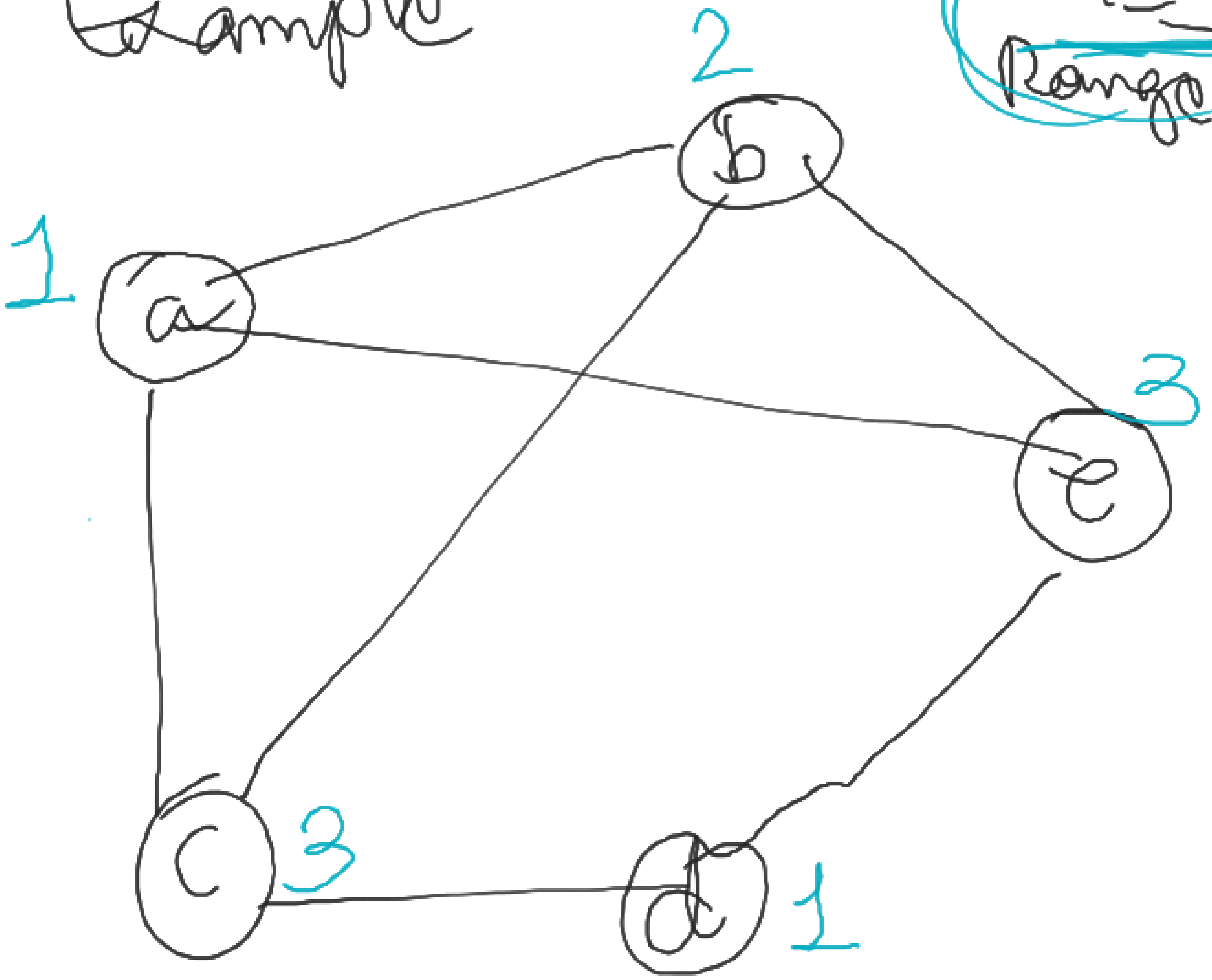
m Coloring (By default talking about coloring vertices)

Let G <V,E> be a graph.
(nxn matrix)

and m be a given positive integer.

"We want to discover whether the nodes of G can be colored in such a way that no two adjacent nodes have the same color yet only m colors are used. (1, 2, ..., m) Range of color

Example



$m = 3$

Range of colors 1, 2, 3

name of vertices
a, b, c, d, e

$n = 5$

	a 1	b 2	c 3	d 4	e 5
a 1	0	1	1	0	1
b 2	1	0	1	0	1
c 3	1	1	0	1	0
d 4	0	0	1	0	1
e 5	1	1	0	1	0

1
2
3
1
3

Answer

vertex a should be colored "1"
b "2"
:

(possible multiple solutions)

Graph's chromatic number

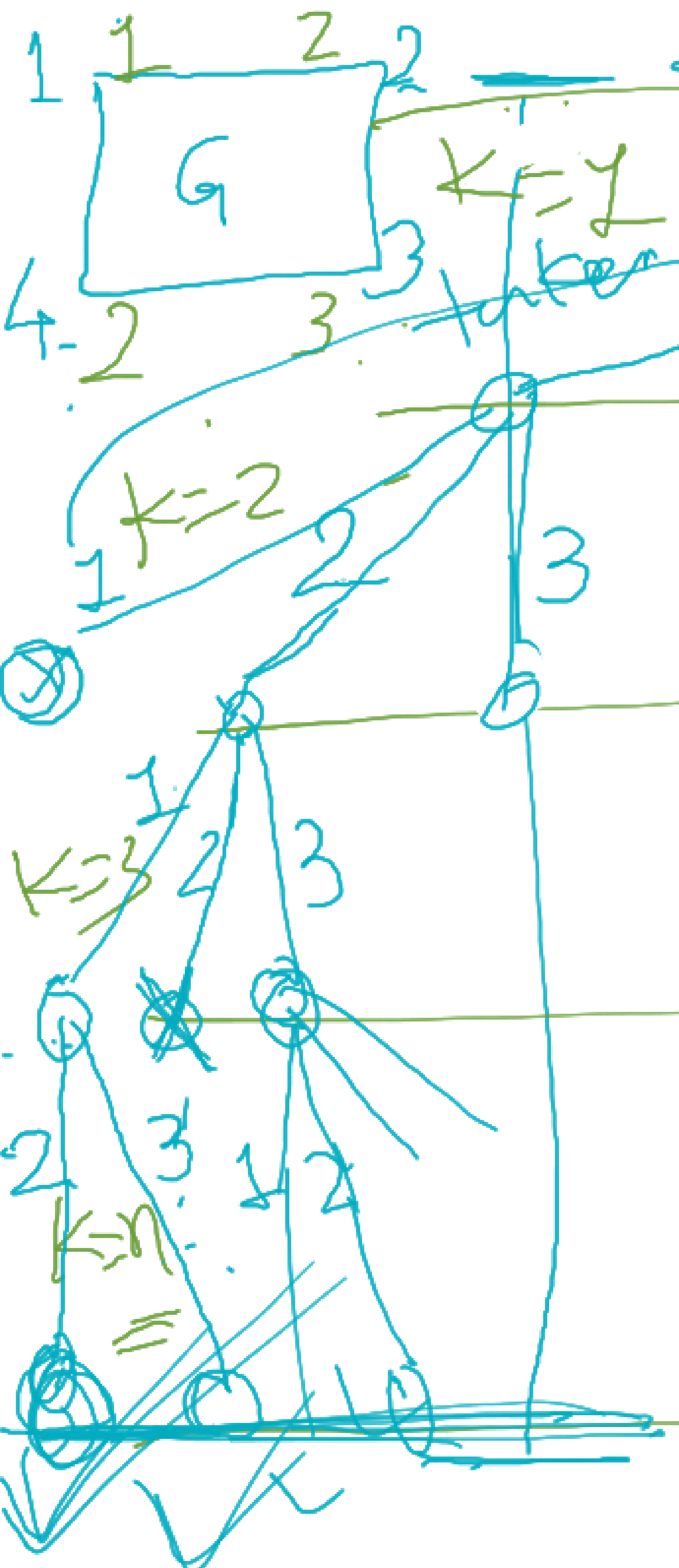
The m-colorability optimization problem asks for the smallest integer m for which graph G can be colored.
This smallest number of color range i.e m=3 is called chromatic number for the graphs.

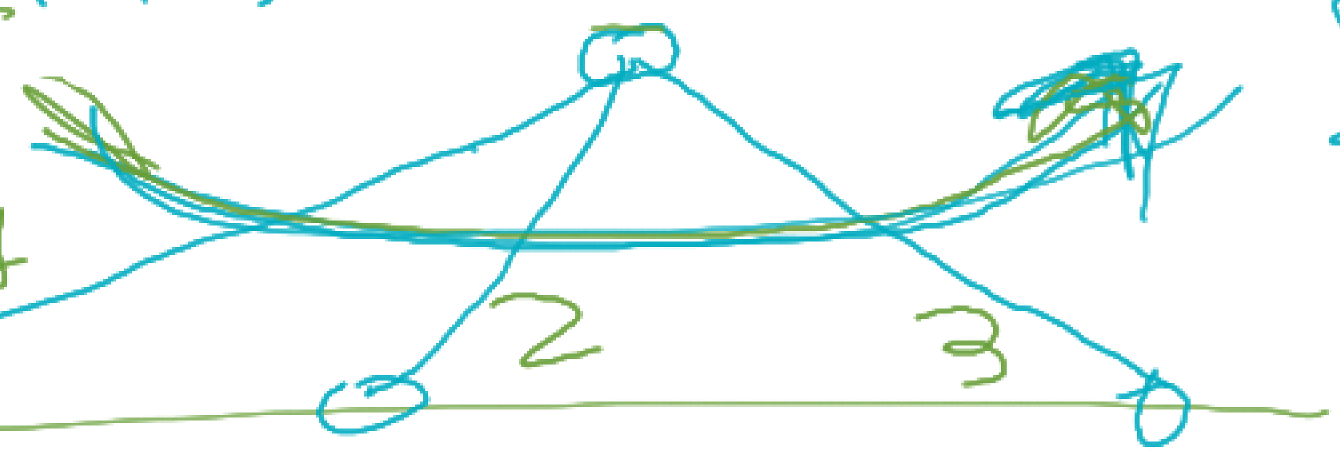
scientists actually have proven

$m=4$ max

Any graph can be colored.

$\{1,2,3\}$ $m=3$, $n=4$





colorResult

1	1	1
2	2	2
3	1	3
4	2	2

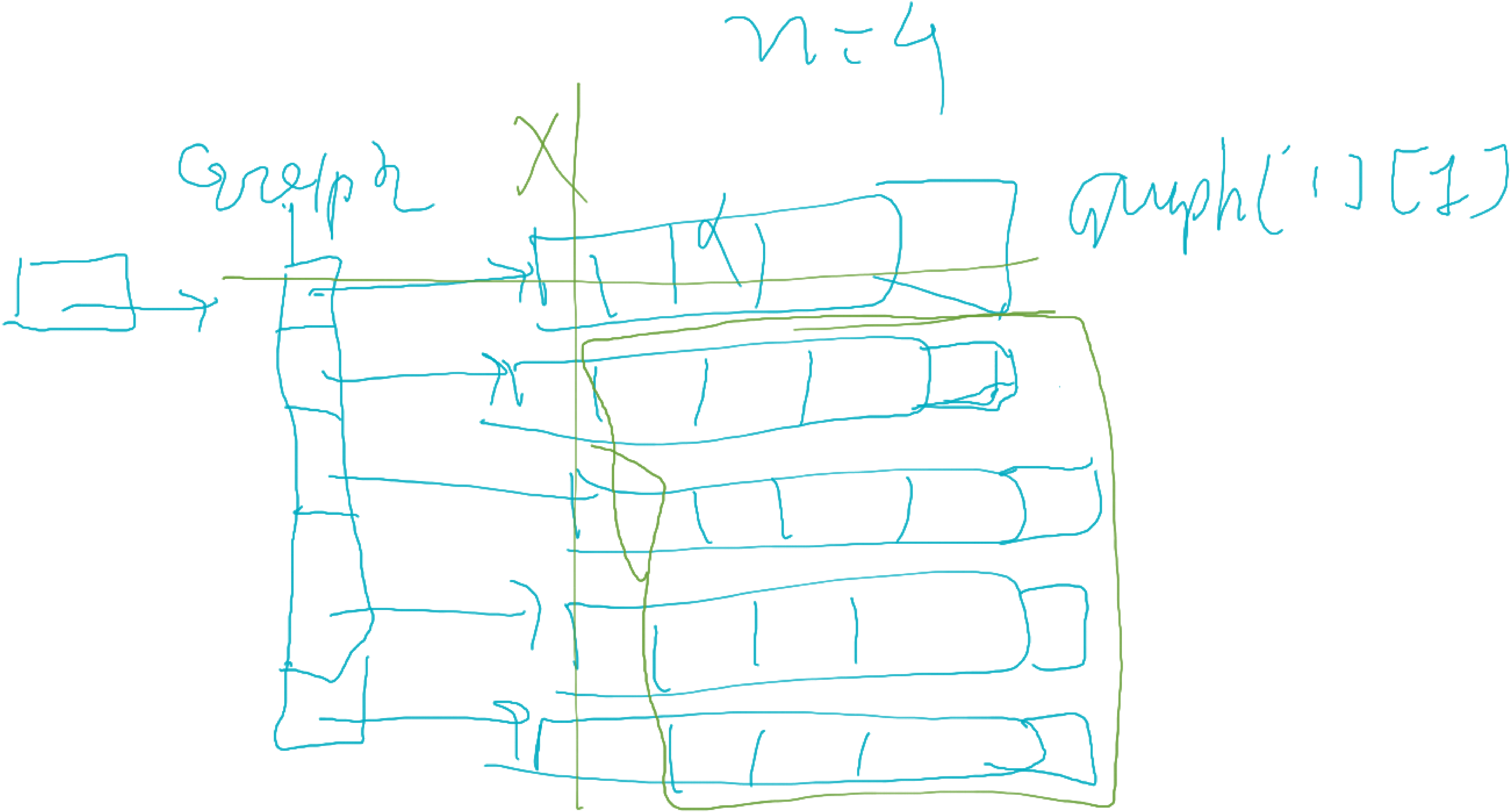
1, 2, 3

	1	2	3	4
1	0	1	0	1
2	1	0	1	0
3	0	1	0	1
4	1	0	1	0

① Cross color part

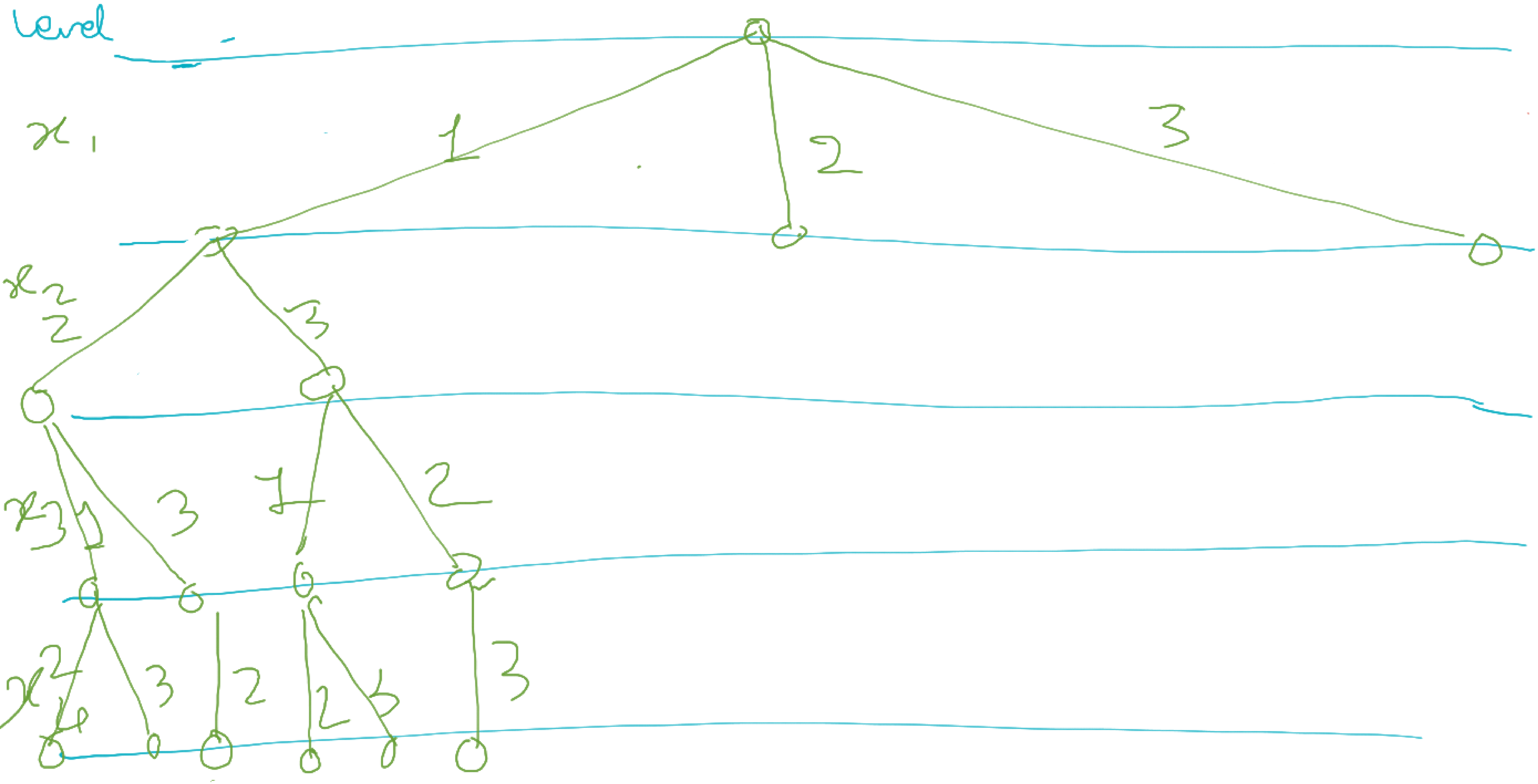
1

② All Vertices may not be connected


n = 4
Graph
graph[i][1]

1	2	1	2
1	2	1	3
1	2	3	2
1	3	1	2
1	3	1	3
1	3	2	3
2	1	2	1
2	1	2	3
2	1	3	1
2	3	1	3
2	3	2	1
2	3	2	3
3	1	2	1
3	1	3	1
3	1	3	2
3	2	1	2
3	2	3	1
3	2	3	2

Total solutoins found are 18


Level
x1
1
2
3
x2
2
3
x3
1
3
1
2
x4
2
3
2
2
3
3

Algorithm mColoring(k)
```
{
repeat
{
  NextValue(k); // Assign to x[k] a legal color
  if (x[k] = 0) then return; // No soln
  if (k = n) then
      write(x[1:n]); // display soln
  else
      mColoring(k+1); // Recursive move onto next level
} until(false);
}
```

NextValue(k) //

```
{
  repeat
  {
    x[k] = x(k)+1 mod (m+1)
    if(x[k] = 0) then return; //All colors have been used
    for(j=1 to n) do
    {
      //check if the color is distinct from adjacent colors
      if((G[k,j] ≠ 0) and (x[k] = x[j]))
      then break
    }
    if (j = (n+1)) return; // new color found
  } until(false);
}
```

<0 to 4>

<0 to m>

3 | 8

3 | 9