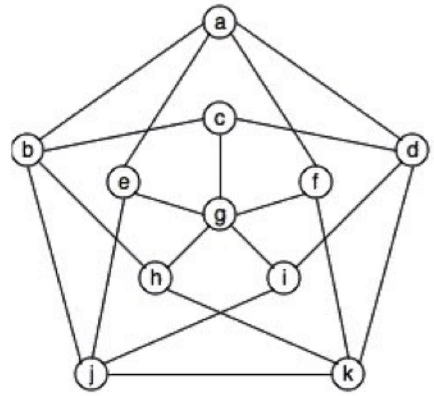
Q5.3) Apply your algorithm (from 5.2) to the graph in 5.1. Show step by step. How many colors did your algorithm use?



Let *G* be the graph above, with 11 vertices and 20 edges.

So, what we seek is a ***k-coloring*** of our graph with k as small as possible.
Now, we know that a vertex is assigned one of the k colors and every pair u,v of adjacent vertices, u and v are assigned different colors.

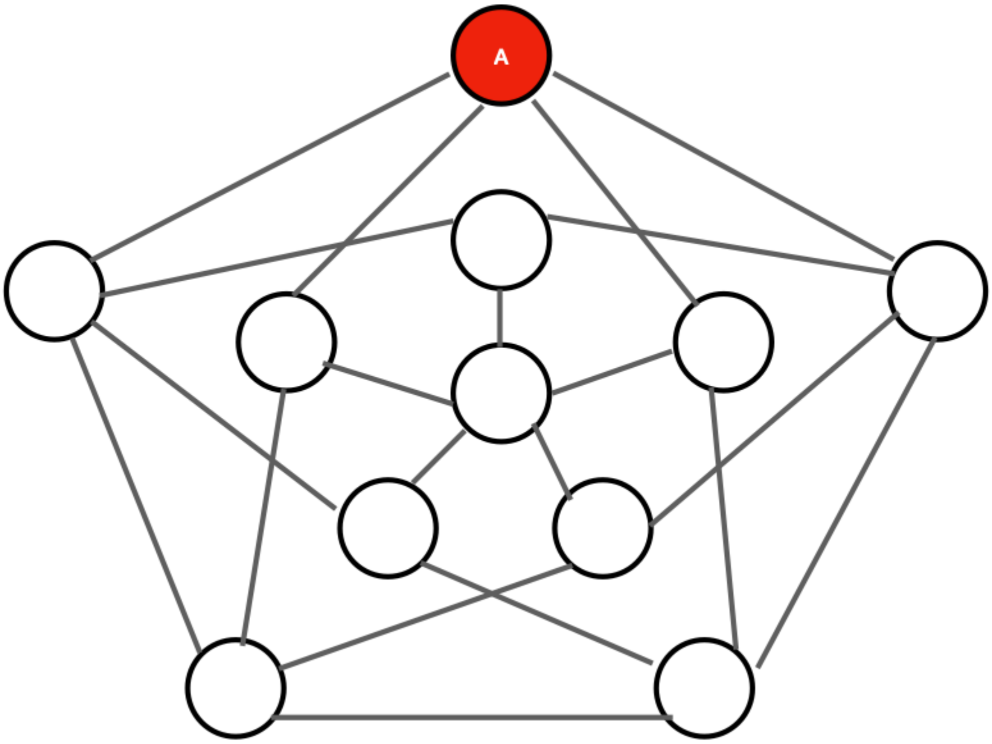
Procedure

A simple greedy algorithm for creating a proper coloring is shown below. The basic idea is do a single pass through all vertices of the graph in some order and label each one with a numeric identifier. The procedure requires us to number consecutively the colors that we use, so each time we introduce a new color, we number it also. A vertex will labeled/colored with the lowest value that doesn't appear among previously colored neighbors.

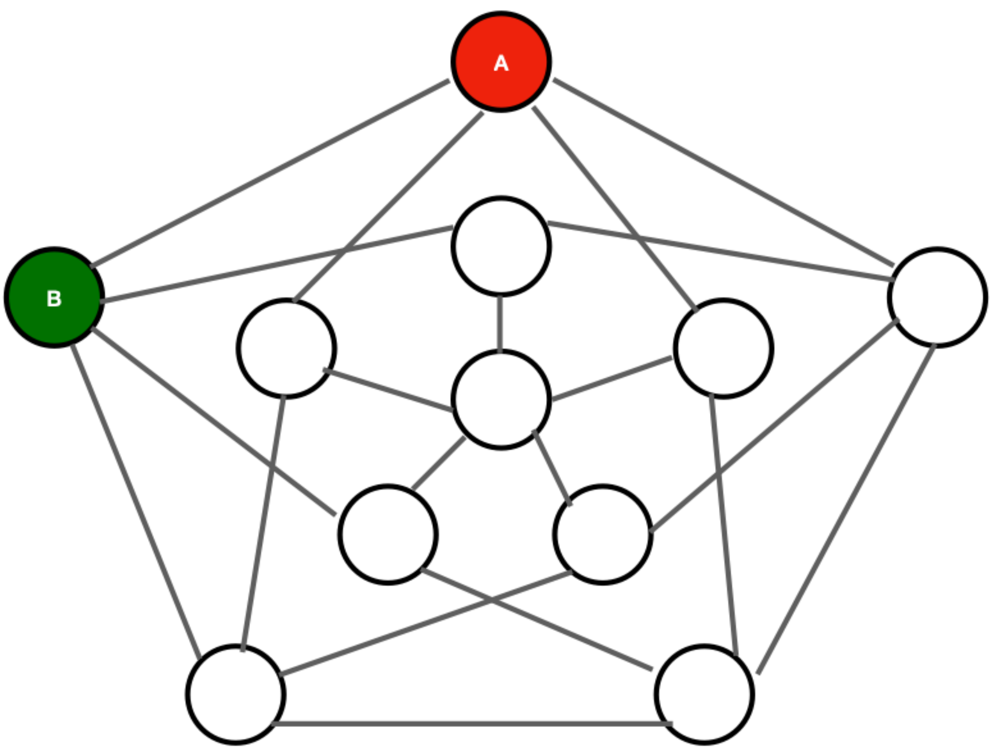
1. Color the starting vertex with color 1.
2. Pick an uncolored vertex v. Color it with the lowest-numbered color that has not been used on any previously-colored vertices adjacent to v. If all previously-used colors appear on vertices adjacent to v, this means that we must introduce a new color and number it.
3. Repeat the step-2 until all vertices are colored.

We define the k-color list as follows,
k-colors = [red, green, yellow, blue, pink, purple, gray, orange, brown, black]

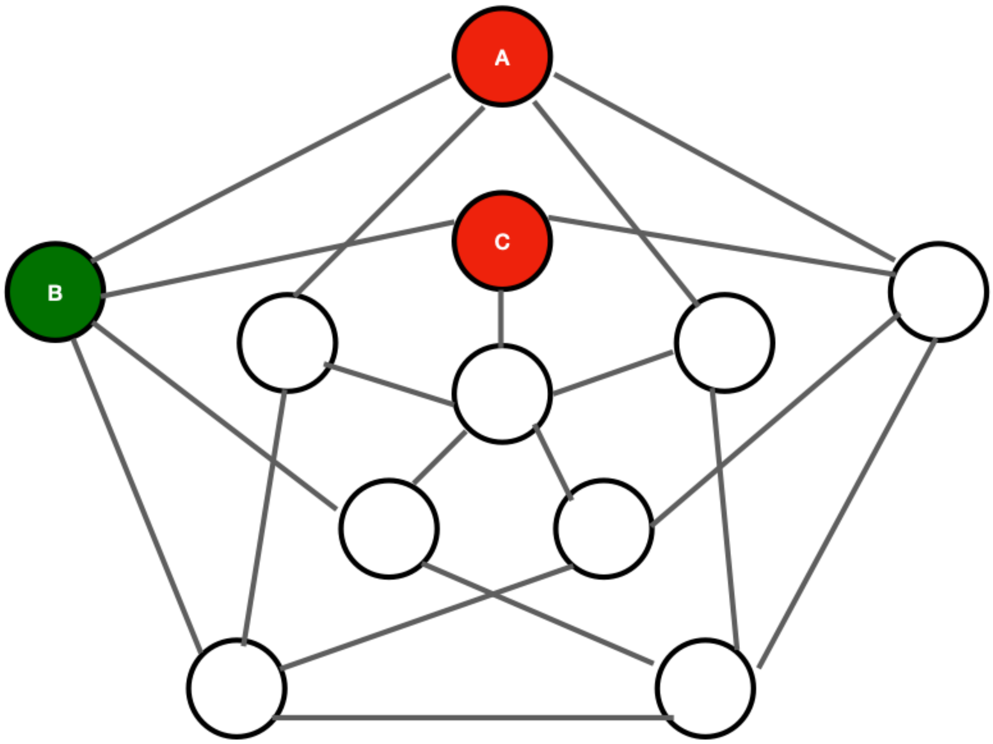
So, starting with vertex A, let us assign the vertex one of the k colors., which is red. So, the graph G would look like,



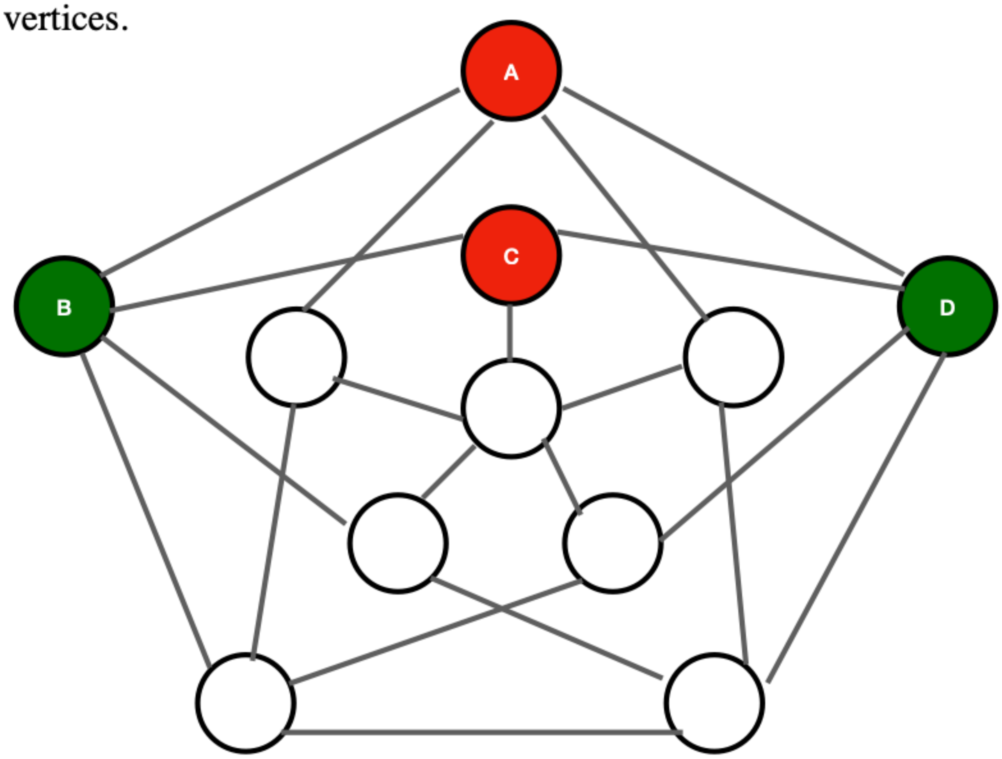
So, now adjacent vertex A are vertices {B, E, F, D}. So, we cannot assign the same color. Let us assign the next green k-color from the list to the vertices B, as it does not share edges with previously green-colored vertices.



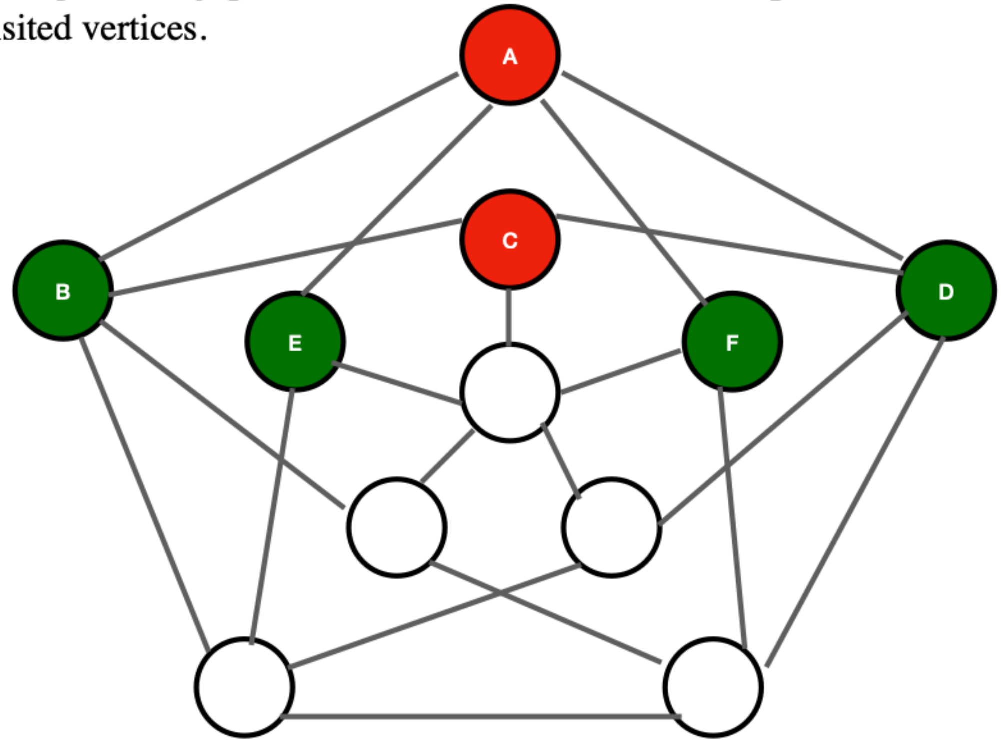
Now, vertex C shares an edge with vertex B and D and does not share an edge with vertex A. So, we can assign the red color to vertex C.



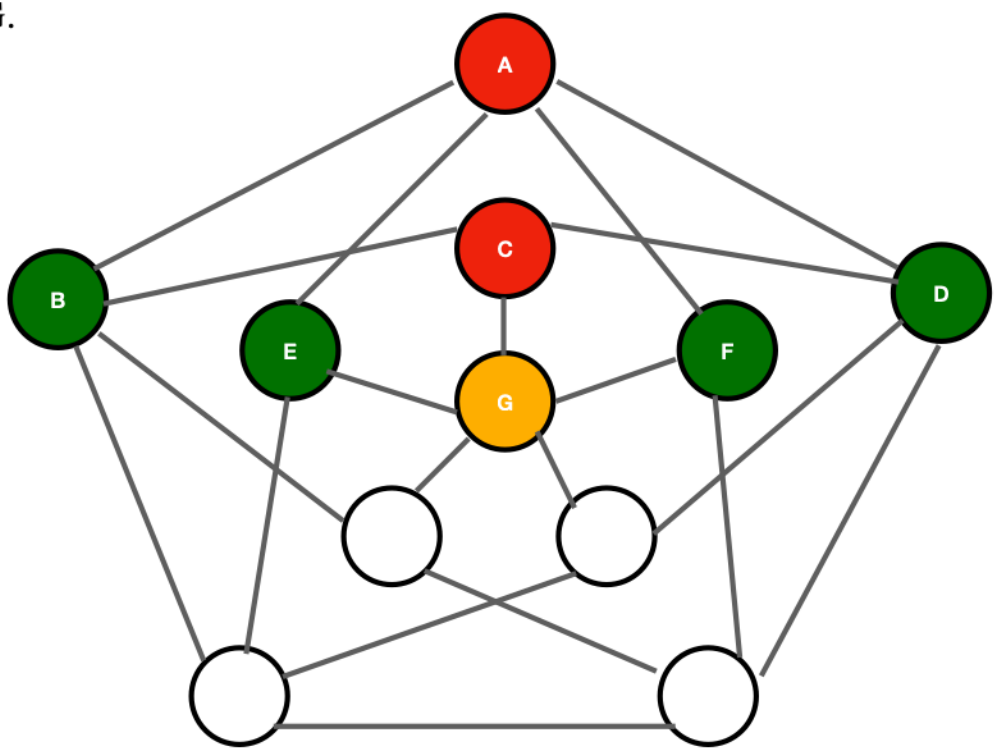
Next, we see that vertex D in outer ring shares an edge with vertex A and C, so we can assign green color to D, as it does not share edges with previously green-colored vertices.



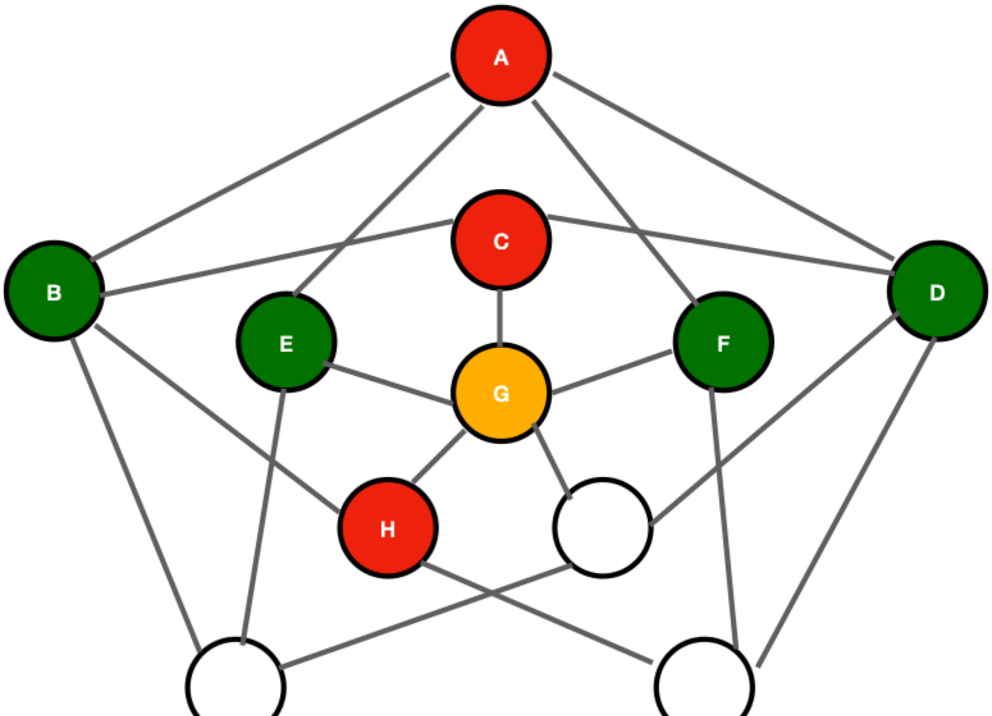
Next, we check that vertex E and F in inner ring shares an edge with vertex A, J, K and G, so we can assign green color to E and F, respectively, as they do not share edges with previously green-colored vertices and A is assigned red color. J and G are unvisited vertices.



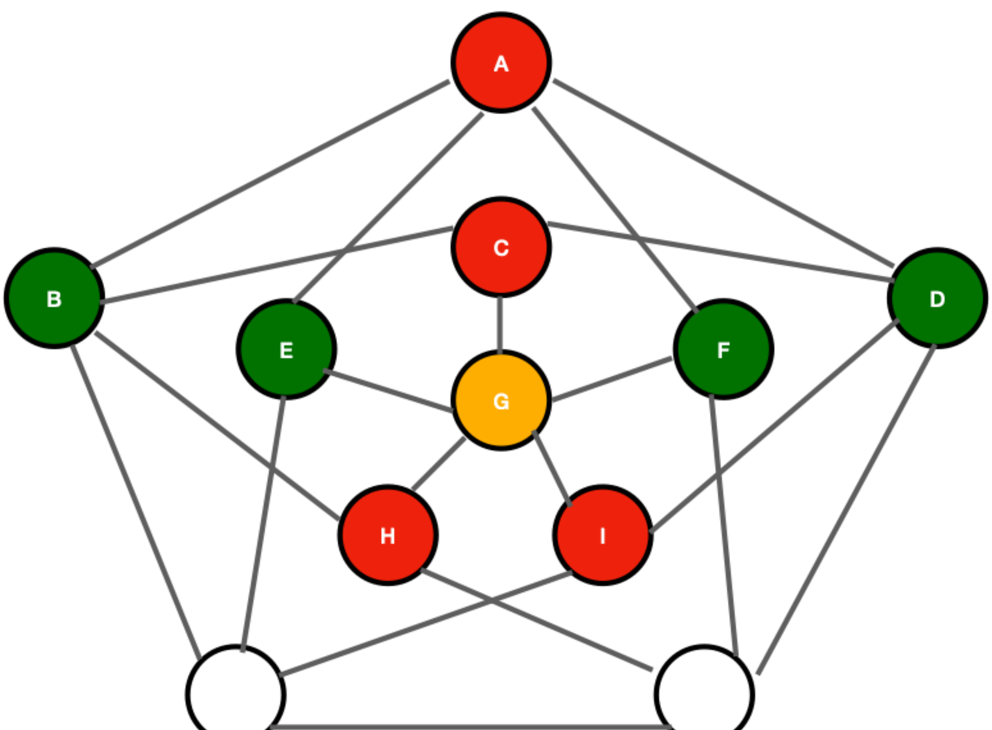
Now, we can see that vertex G in the innermost ring share an edge with vertex C, E, F, H and I in the inner ring, so we cannot assign green and red color to the vertex G as it shares an edge with previously colored red and green vertices. We must introduce a new yellow color from the visited k-colors and assign the color to vertex G.



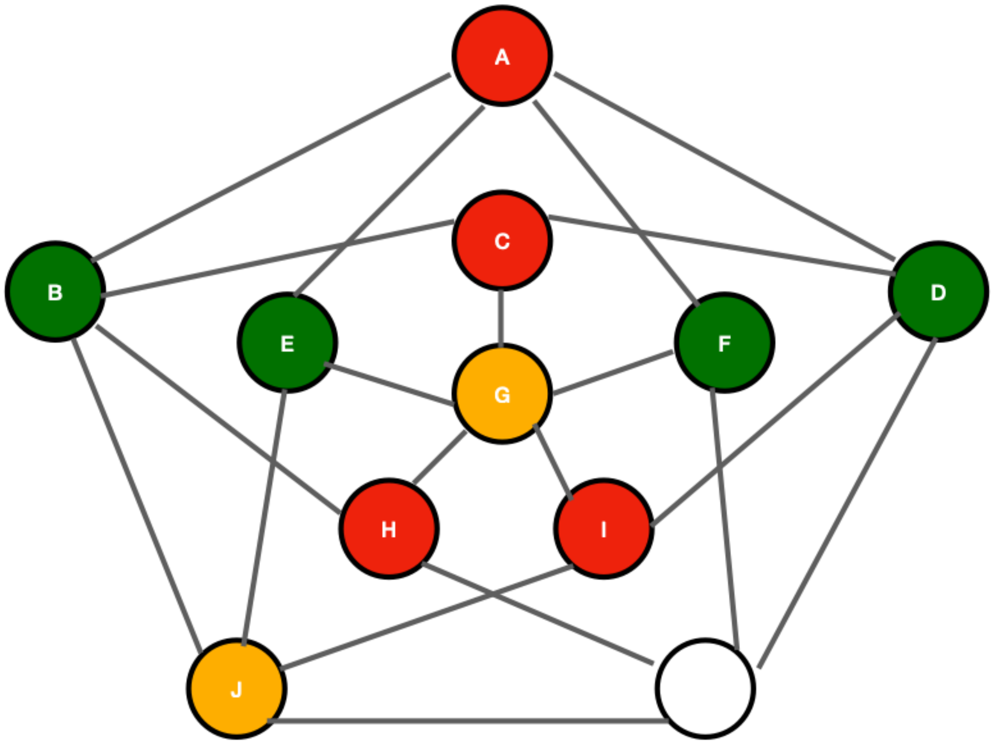
Now, we can see that vertex H shares an edge with vertex B, G, and K, so we cannot assign green and yellow color to the vertex H as it shares an edge with previously colored yellow and green vertices. We can assign a previously visited color red to vertex H as it does not share edges with red colored vertices.



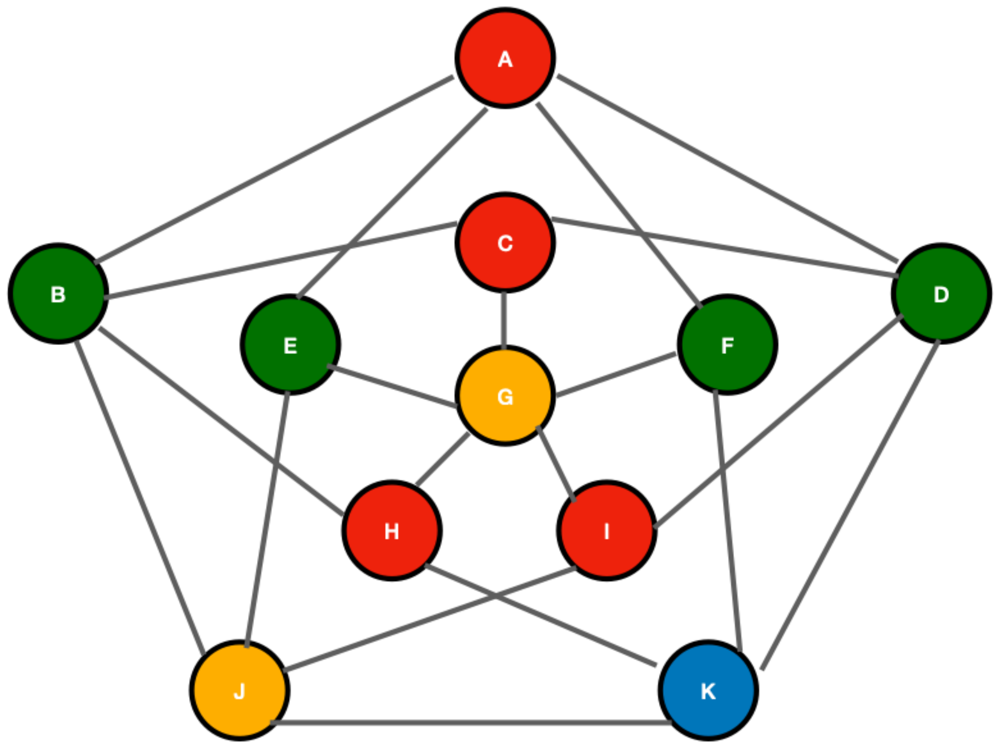
Similarly, vertex I shares an edge with vertex D, G, and J, so we cannot assign green and yellow color to the vertex I. We can assign a previously visited color red to vertex I as it does not share edges with red colored vertices.



Now, we can see that vertex J shares an edge with vertex B, E, I and K, so we cannot assign green and red color to the vertex J as it shares an edge with previously colored red and green vertices. We can assign a previously visited color yellow to vertex J as it does not share edges with yellow colored vertices.



Lastly, we can see that vertex K shares an edge with vertex J, H, F and D, so we cannot assign green, yellow and red color to the vertex K as it shares an edge with previously colored red, yellow and green vertices. We must introduce a new blue color from the visited k-colors and assign the color to vertex K as all the other colors are exhausted and used in the adjacent vertices by Greedy algorithm.



Finally, we can see the graph is k-colored as every vertex of G is assigned one of the k-colors and number of colors needed for the proper coloring of the graph is 4 using the Greedy algorithm. The order in which the vertices are assigned color is,

Color assigned to vertex A is RED
Color assigned to vertex B is GREEN
Color assigned to vertex C is RED
Color assigned to vertex D is GREEN
Color assigned to vertex E is GREEN
Color assigned to vertex F is GREEN
Color assigned to vertex G is YELLOW
Color assigned to vertex H is RED
Color assigned to vertex I is RED
Color assigned to vertex J is YELLOW
Color assigned to vertex K is BLUE
Chromatic number of the graph is: 4

We know that the chromatic number of a graph G, denoted by $\boldsymbol{\chi(G)}$, is the smallest integer k for which graph G is k-colorable.
In this graph, we used 4 different colors to assign colors using the Greedy algorithm to all the vertices for which the graph G is proper colored. Hence, $\chi(G) = 4$, for graph G with 11 vertices and 20 edges.