(a). Greedy algorithm:

P5

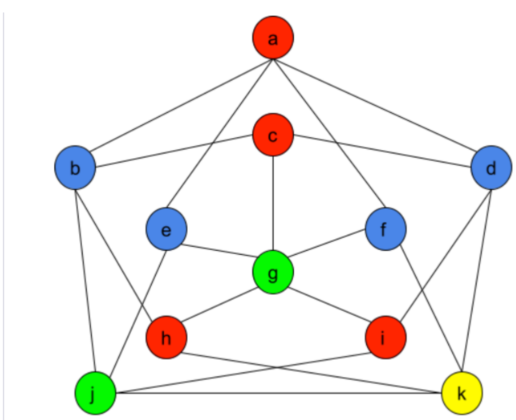
① Color first vertex like vertex a with first color;

② Do following for remaining V−1 vertex, considering the currently picked vertex, then color it with the lowest numbered colour that has not been used on any previously colored vertices adjacent to it. If all previously used color appear on vertices adjacent to v, which means we have to assign a new color to it.

③ Repeat the previous step until our all vertices are colored.

By applying the Greedy algorithm, we can get:

$\chi(G) = 4.$



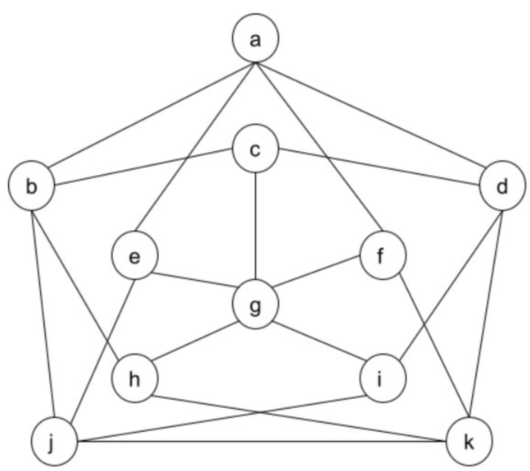
(b). Greedy Algorithm:

① Color a vertex with first color;

② Picked 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 previous used colors appear on vertices adjacent to V, we have to assign a new color to it.

③ Repeat the previous step until all vertices are colored.

I will take an example to show the color order and how my algorithm works.



① Consider the given graph, apply the graph coloring from vertex a, color the vertex to Red.

② Consider the next vertex b, as vertex b is directly connected with vertex a, so color the vertex b to Blue.

③ Consider the next vertex c, since c is directly connected with vertex b, so color vertex c to Red.

④ Consider the next vertex d, which is directly connected with a and c, so color the vertex d to Blue.

⑤ Consider the next vertex e, which is directly connected with a, so color vertex e to Blue.

⑥ Consider next vertex f, same as vertex e, we color it to Blue.

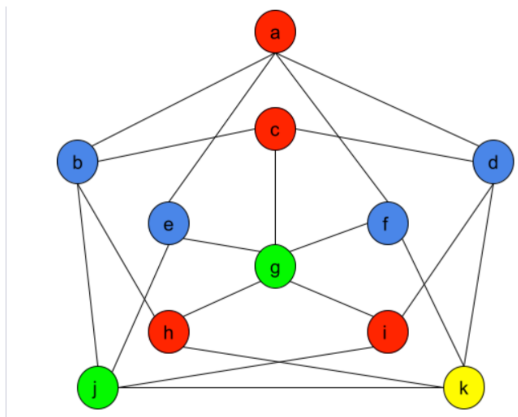
⑦ Consider next vertex g, since g is directly connected with d, e, f, which have been colored, so we assign a new color Green to g.

⑧. Consider the next vertex h, which is directly connected with b and g, so we color the vertex h to Red.

⑨. Consider the next vertex i, which is directly connected with d and g, so we color vertex i to Red.

⑩ Consider the next vertex j, which is directly connected with vertex b, e and i, which have been colored before, so we color vertex j to Green.

(ii) Consider the next vertex k, since k is directly connected with vertex d, f, h and j, so the color of vertex k will not be Blue, Red, Green. So, we will assign a new color Yellow to vertex k.

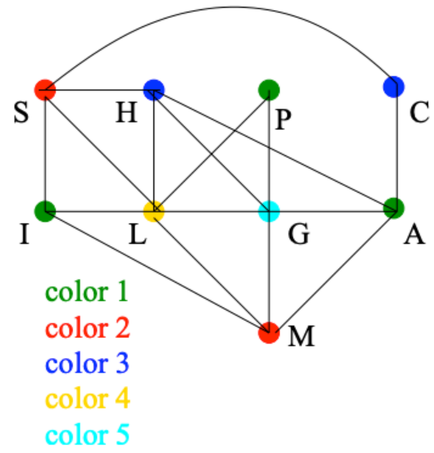


So, from the above graph, it is clearly that the greedy algorithm needs 4 colors to color this graph.

(c). NO.

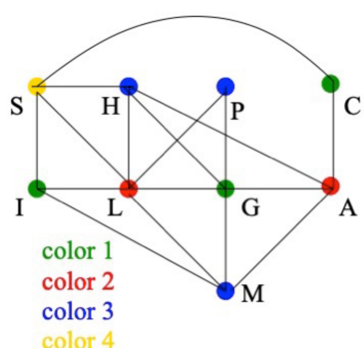
For some graphs, the algorithm does not always use exactly $\chi(G)$ colors.

Suppose we choose to color the vertices in order A, I, P, M, S, C, H, L, G. First we will color A with color 1 (green) and also I and P color green. Finally, we can get the following graph color:



So, we can see that the quality of our coloring from Greedy algorithm is dependent on the order in which we color the vertices.

If we decide to color the vertices in order G, L, H, P, M, A, I, S, C. we will only use 4 colors to color all vertices.



(d). If a chromatic number is 2, this graph is known as Bipartite graph. we can use a similar BFS to tell.

so, the algorithm is as follows:

① Creat 2 sets, set A and set B, initially both are empty.

② Start with any vertex and perform BFS:
add first vertex into set A, and add all its neighbors into set B. And we make sure that this vertex is not present in set B already. As it happens, we will return false if is present in both sets.

③ If we are able to seperate all the vertices into two sets, then we will return true.

The total time complexity is $O((V+E)V)$, where E is number of edges, and V is number of vertices.

so, we taking n as number of vertices edges, the total complexity is $O(n^3)$.