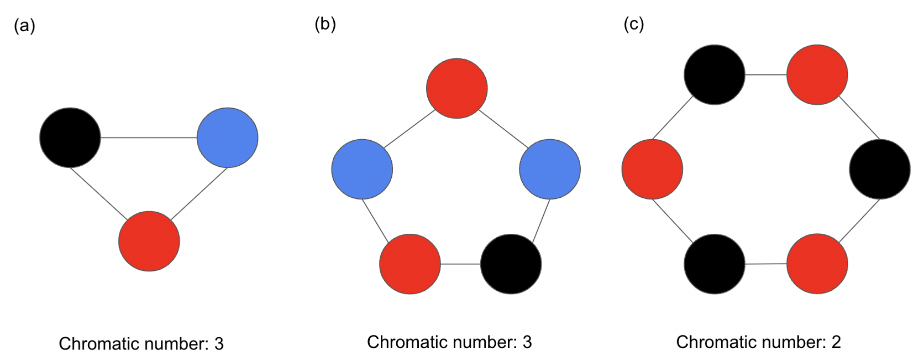
# CS271: Project 6

Krystal Ly, Hamy Nguyen, Casey Kotbyul Kim

2 December 2022

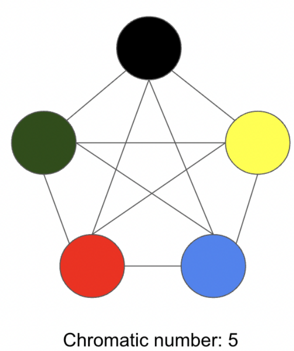
## 1 Exercises

1. Consider each of the following graphs. Show a proper coloring of each using only the chromatic number. of colors.




2. Coloring of cliques

   (a) Show a proper coloring of the following clique using only the chromatic number of colors.



(b) What can you say about the relationship between $|V|$ in a graph G = (V, E) and G's chromatic number if G is a clique?

- If G is a clique, the chromatic number of G is equal to $|G.V|$.

(c) Prove your answer from 2b.

*Proof.* For a proof by contradiction, assume that $G$ is a clique and the chromatic number of $G$ is not equal to $|G.V|$. Since the number of colors cannot be greater than the number of vertices, we have that the minimum number of colors in a proper coloring of $G$ is less than $|G.V|$. Therefore, there are two vertices, $J, I$, that have the same color. $J, I$ must not be connected to maintain proper coloring. If $J, I$ is not connected, $G$ is not a clique. This is a contradiction.
Because we got a contradiction, our assumption that $G$ is a clique and the chromatic number of $G$ is not equal to $|G.V|$ is wrong. Therefore, we proved that if $G$ is a clique, the chromatic number of $G$ is equal to $|G.V|$. □

3. Odd-Length Cycles

(a) Prove the following: An undirected graph is bipartite if and only if it contains no cycles of odd length.

*Proof.* In order to prove the statement "An undirected graph is bipartite if and only if it contains no cycles of odd length", we have to prove two statements:

- If an undirected graph is bipartite, it contains no cycles of odd length
- If an undirected graph contains no cycle of odd length, it is bipartite

1. If an undirected graph is bipartite, it contains no cycles of odd length

For a proof by contradiction, assume that an undirected graph G is bipartite and it contains at least one cycle of odd length $n$. Let $v_1, v_2, v_3, ..., v_n$ be vertices that create one cycle of odd length such that there is an edge between $v_1$ and $v_2$, $v_2$ and $v_3$,...,$v_n$ and $v_1$. Let $S_1$ and $S_2$ be two sets that the vertices can be partitioned into. Let $v_1 \in S_1$. Since there is an edge between $v_1$ and $v_2$, $v_2$ can't be in the same set as $v_1$. Therefore, $v_2 \in S_2$. Since there is an edge between $v_2$ and $v_3$, $v_3 \in S_1$. Since there is an edge between $v_3$ and $v_4$, $v_4 \in S_2$. It follows by induction that for an odd number $i$, $v_i \in S_1$ and for an even number $j$, $v_j \in S_2$. Since this is an odd-length cycle of length $n$, we have $v_n \in S_1$. We also have $v_1 \in S_1$. However, since $G$ is bipartite and there is an edge between $v_n$ and $v_1$, $v_n$ and $v_1$ can't be in the same set. This is a contradiction.
Because we got a contradiction, our assumption that an undirected graph

G is bipartite and it contains at least one cycle of odd length $n$ is wrong. Therefore, we proved that if an undirected graph is bipartite, it contains no cycles of odd length.

2. If an undirected graph contains no cycle of odd length, it is bipartite
For a proof by contradiction, assume that an undirected graph contains no cycle of odd length and it is not bipartite. We partition the graph into two sets $S_1$ and $S_2$. Let $v_i \in S_1$ ($i$ is even) and $v_j \in S_2$ ($j$ is odd). Let $k$ be an even number. Since G is not bipartite, there will always be a cylcle $C =< v_0, v_1, ....v_{k-1}, v_k, v_0 >$. We have that the length from $v_0$ to $v_{k-1}$ is odd because $v_{k-1} \in S_2$. We also have the length from $v_{k-1}$ to $v_0$ is 2. Therefore, the cycle $C =< v_0, v_1, ....v_{k-1}, v_k, v_0 >$ has odd length. This is a contradiction. Because of the contradiction, our assumption is wrong. Therefore, if an undirected graph contains no cycle of odd length, it is bipartite

□

(b) Design a linear time algorithm to determine whether an undirected graph G is bipartite. Provide pseudocode for your solution. Pseudocode syntax should be consistent with your textbooks. For all functionality required by your algorithm, the corresponding pseudocode must be provided.

**Algorithm 1** BIPARTITE($G$)

```
for each vertex v ∈ G.V do                          ▷ Initialize vertices
    v.color = WHITE
end for
for each vertex v ∈ G.V do          ▷ Run BFS on non-discovered vertices
    if v.color == WHITE then
        BFS − BIPARTITE(G, v)
        if BFS − BIPARTITE(G, v) == False then return False
        end if
    end if
end for
return True
```

**Algorithm 2** BFS-BIPARTITE$(G, s)$

```
s.color = RED                                        ▷ Color the source vertex RED
Q = ∅
ENQUEUE(Q, s)
while Q ≠ ∅ do
    u = DEQUEUE(Q)
    for each vertex v ∈ G.Adj[u] do
        if u == v then                    ▷ if there is a self-edge, G is not bipartite
            return False
        else if v.color == WHITE then ▷ Color adjacent nodes in different color
            if u.color == RED then
                v.color = BLUE
            else if v.color == BLUE then
                v.color = RED
            end if
            ENQUEUE(Q, v)
        else
            if u.color == v.color then    ▷ if neighbors share same color, G is not bipartite
                return False
            end if
        end if
    end for
end while
return True
```

(c) At most, how many colors are needed to color an undirected graph with exactly one odd-length cycle?

- The largest chromatic number of an undirected graph with exactly one odd-length cycle is 3. A bipartite graph, which contains no odd length cycles needs 2 colors for its coloring. If we add exactly one odd cycle in that graph, the graph no longer becomes bipartite and would need an extra color to recolor one of the vertices in the odd cycle. Therefore, $2 + 1 = 3$.