Time Spent: 1 hour

Collaborators and Resources:

Bonus Problem

For this problem I wanted to demonstrate that it might be possible to test if any graph G is tripartite. Namely, instead of running DFS or BFS on a graph to get a list D of distances from the start vertex S and then coloring even-distanced vertices red and odd-distanced vertices blue, I wanted to demonstrate a fundamental approach. We know that if a graph contains a K4 complete subgraph (that is, a subset of four vertices, each of which is connected to all of its three neighbors) it can't be colored using three colors and won't be tripartite (because a tripartite graph consists of three groups which are not intraconnected and so we can assign one color to each group). The result is that we have a way to determine if a graph contains such a subgraph ...

1. Given a graph G = (V,E)
2. For each group of four vertices $\in$ V
   (a) Check whether they are all connected to each other
   (b) If so, return False
3. Otherwise, return True

Claim 1. The algorithm will determine, at least for some graphs, if a graph is tripartite in $O(\binom{n}{4}) \rightarrow O(n^4)$ time

Proof. If there are four interconnected vertices then we can be sure that the graph is not tripartite because there will be a fourth vertex which cannot be colored with any of the 3 colors available. The algorithm might not identify all such graphs but will be fairly accurate. Also, it runs in $O(n^4)$ time because of the fact that the number of ways to choose a group of four vertices in V is $\binom{n}{4}$ which is -$\frac{n}{4}$ + $\frac{11}{24}n^2$ - $\frac{1}{4}n^3$ + $\frac{1}{24}n^4$, which has the largest exponent n^4 within the polynomial. $\square$