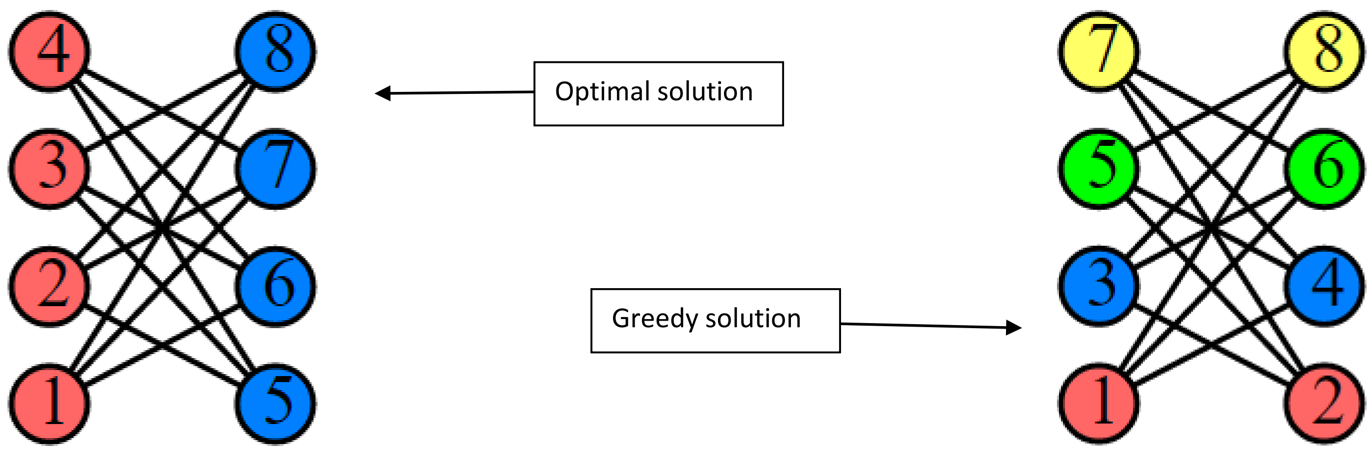
**Problem 1)**




To show that the greedy coloring of a graph does not approximate the optimal coloring to within any constant ratio we will look at the coloring of a bipartite graph using greedy coloring with the vertex order $\{a_1, b_1, a_2, b_2, \ldots, a_n, b_n\}$. First, each of $a_1$ and $b_1$ will be assigned color 1, since they are not adjacent to any already-colored vertices. Then, the vertices $a_2$ and $b_2$, which are each adjacent to a vertex already assigned color 1, will be assigned color 2. Then, $a_3$ and $b_3$, each adjacent to vertices of colors 1 and 2, must be assigned color 3. This procedure can be continued indefinitely, until $a_n$ and $b_n$, each adjacent to vertices of every color from 1 through $n - 1$, must be assigned color n. So, if the vertex ordering places two vertices consecutively whenever they belong to one of the pairs of the removed matching, then a greedy coloring will use *n* colors (as shown on the right), while the optimal number of colors for this graph is two (as shown on the left).

Citations: 1) https://en.wikipedia.org/wiki/Greedy_coloring, Author: Unknown, Title: Greedy Coloring, Date: Unknown.

2) http://aleph.math.louisville.edu/teaching/2010SP-682/PS04-100323-solutions.pdf, Author: Unknown, Title: Problem Set #4, Date: 2010.

**Problem 2)**

The vertices of any graph may always be ordered in such a way that the greedy algorithm produces an optimal coloring. For, given any optimal coloring in which the smallest color set is maximal, the second color set is maximal with respect to the first color set, etc., we can order the vertices by their colors. Then when we use a greedy algorithm with this order, the resulting coloring is automatically optimal.

To show the above described ordering method we will let k be the optimal coloring. Then graph G has a k-coloring $c : V(G) \to \{1, 2, \ldots, k\}$. Let $V_1, V_2, \ldots, V_k$ be the partition of V (G) into color classes; that is, $v \in V_i$ if $c(v) = i$. Let us have an arbitrary ordering $(v_{i,1}, v_{i,2}, v_{i,3}, \ldots, v_{i,|V_i|})$ on each $V_i$, and assemble an overall ordering of the vertices of G:

$$(v_{1,1}, v_{1,2}, \ldots, v_{1,|V_1|}, v_{2,1}, v_{2,2}, \ldots, v_{2,|V_2|}, \ldots, v_{k,1}, v_{k,2}, \ldots, v_{k,|V_k|})$$

Now, let $c_g$ be the greedy coloring induced by this ordering. We shall see that $c_g(v) \leq c(v)$ for all v, and so $c_g$ uses only k colors (it cannot use any fewer, since G is not $(k - 1)$-colorable).

Using inductive reasoning on i we can argue that $c_g(v_{i,j}) \leq c(v_{i,j})$. For $i = 1$, each $v_{1,j}$ has no colored neighbors (since the $v_{1,j}$ vertices are colored first, and are not adjacent to each other), so the greedy coloring assigns them the color 1, so $c_g(v_{1,j}) = 1 = c(v_{1,j})$.

For the inductive step, let us consider the step at which the greedy coloring chooses a color for $v_{i,j}$ . At this point, the greedy coloring has assigned colors only to vertices of the form $v_{i',j'}$ for $i' < i$, or $v_{i,j'}$, where $j' < j$. Vertices of the latter form are in $V_i$ , and must be nonadjacent to $v_{i,j}$ . Thus, the neighbors of $v_{i,j}$ which have already been colored are all of the form $v_{i',j'}$ for $i' < i$. By the inductive hypothesis, each $cg(v_{i',j'}) \leq c(v_{i',j'}) = i' < i$, so every neighbor of $v_{i,j}$ which has been colored has been assigned a color less than i. Thus, since $v_{i,j}$ is assigned the smallest color not used in its neighborhood, it is assigned a color less than or equal to i, so $c_g(v_{i,j}) \leq i = c(v_{i,j})$.

Unfortunately, the greedy algorithm is not guaranteed to find the particular optimal coloring c, and in fact there are minimal colorings not achievable by greedy coloring; for instance, if we have a $K_3$ with vertices $\{v_1, v_2, v_3\}$, and we attach a vertex u adjacent to $v_3$, and assign the coloring $c(v_1) = 1$, $c(v_2) = 2$, $c(v_3) = 3$, and $c(u) = 2$, no greedy coloring will ever match this coloring, since u will preferably be assigned color 1 by a greedy algorithm.

Citations: 1) https://en.wikipedia.org/wiki/Greedy_coloring, Author: Unknown, Title: Greedy Coloring, Date: Unknown.

2) http://aleph.math.louisville.edu/teaching/2010SP-682/PS04-100323-solutions.pdf, Author: Unknown, Title: Problem Set #4, Date: 2010.

**Problem 3)**

To prove NP-Complete we will first see if it is in the class NP meaning that its solution should be verifiable in polynomial time. But to see if an ordering is the optimal ordering we would need to compare it to all possible orderings. So, for a graph with n vertices we would have to find and compare all $2^n$ orderings that the graph could have. So, the verification would take exponential time, not polynomial time. Therefore since this problem of finding the optimal ordering, is not in the class NP so it is not NP-complete.

Otherwise, determining the optimal ordering is NP-hard. This is because it could be used to solve the NP-complete graph coloring problem. We can show this by the following reduction: G-COLOR $\leq_p$ OPT-ORDER. Graph coloring can be reduced to optimal ordering by just classifying vertices by their color. This will allow us to keep an ordering that can be followed by the greedy algorithm to have optimal coloring. This reduction is done in polynomial time. Therefore, optimal ordering is only NP-hard.

Citations: 1) https://en.wikipedia.org/wiki/Greedy_coloring, Author: Unknown, Title: Greedy Coloring, Date: Unknown.