

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Vertex Number** | **N(V) Colors** | **Color Assigned** |  | **Vertex Number** | **N(V) Colors** | **Color Assigned** |
| 1 | NONE | 1 (Blue) |  | 1 | NONE | 1 (Blue) |
| 2 | NONE | 1 (Blue) |  | 2 | NONE | 1 (Blue) |
| 3 | NONE | 1 (Blue) |  | 3 | 1 | 2 (Green) |
| 4 | 1 | 2 (Green) |  | 4 | 1 | 2 (Green) |
| 5 | 1 | 2 (Green) |  | 5 | 1,2 | 3 (Yellow) |
| 6 | 1 | 2 (Green) |  | 6 | 1,2 | 3 (Yellow) |

In the left example, the greedy graph coloring algorithm will return the optimal solution because the linear iteration done by the algorithm will process the first three nodes before encountering a node that has an already-colored neighbor. In the right example, however, the algorithm does not find the optimal solution as it did before, returning a total of three colors required to color this graph, despite it having the same layout as before, only with a different arrangement of vertex IDs. This non-optimized solution is also result of the way in which the algorithm iterates since it will encounter nodes with already-colored neighbors after only two iterations.