Lucas Cook & Tyler Carrico
11/26/2017

Project 2: Maximum Color Sequence

Pseudocode Design

main(): // No Input

Load the graph (openGraph())
Get nodes from graph
Get edges from the graph
Find Max Independent Set (FindMaxIndependentSet(graph))
Color the graph accordingly (ColorGraph(graph))
Map node positions for the UI display
Draw all nodes
Draw all edges
Show UI

openGraph(): // Input Size: Size of .json object

Load and create graph from .json file
Return graph

FindMaxIndependentSet(G, v): //Input Size : |V + E |

Mark current node visited
Get node's neighbors
For each neighbor
    If it has been visited
        Add its color to adjacent colors set
    Else
        Add it to queue of nodes to visit
Color current node the lowest priority color that is not adjacent
For each node to visit
    Start function over with this node (findMaxIndependentSet(node))

colorGraph(G): //input size = |V + E|

For each node in the graph
    Set node label
    Add node to corresponding color list
Return nodes with labels

Data Structures Used

The language we have decided to use for this project was Python. We also used a python library called networkx for the graph display and fundamental graph functionality. The primary data structure used in this project is a dictionary containing tuples of information about the graph. Below is an in-depth analysis of each data type used in our project.

Main

G - dictionary of nodes and edges
- key= “node” or “edge”, contains another level of dictionaries listed below

nodes - dictionary of nodes
- key=node name, contains tuple of the “visited” and “color” attribute

edges - dictionary of nodes and their adjacent nodes
- key=node name, contains tuple of adj. nodes

current- tuple containing the current node with its attributes “visited” and “color”
- tuple is of constant size 3. (“Node Name”, “visited”, “color”)
- One row of the nodes dictionary

neighbors- tuple of neighbor nodes to current node

adjColors - list of adjacent colors to the current node
- This is utilized as a set to check if the color currently in priority is adjacent to the current node, if current color exists in adjColors then the priority moves on to the next highest priority color until a color not in the adjColors set is found.

next - queue of nodes to visit next
- When recursively calling the FindMaximumSet() function this data structure pops off the first unvisited node that was adjacent to the current, it then check again to see if the node has been visited in a recursive call since the last check. This allows each node to only be visited once.

UI

pos - dictionary of pixel location of each node for UI

labels - dictionary of nodes and their labels
- key=node name, contains UI’s label for the node (Name)

node_list_red - list of nodes with color red

node_list_green - list of nodes with color green

node_list_blue - list of nodes with color blue

node_list_yellow - list of nodes with color yellow

Complexities

Function Name	Complexity
main()	O(|V|+|E|), where V is the number of vertices and E is the number of edges.
openGraph()	O(n) where n is size of .json object
FindMaxIndependentSet()	O(|V| + |E|), where V is the number of vertices and E is the number of edges.
ColorGraph()	O(|V|) where V is number of vertices

main() and Overall Complexity - The overall complexity of this program is O(|V|+|E|). This is due to its underlying backbone of the depth-first search.

openGraph() - Reads in a .json file to construct a graph, which requires O(n) where n is size of input. Function is dependent on python library to convert .json to graph so complexity is estimated.

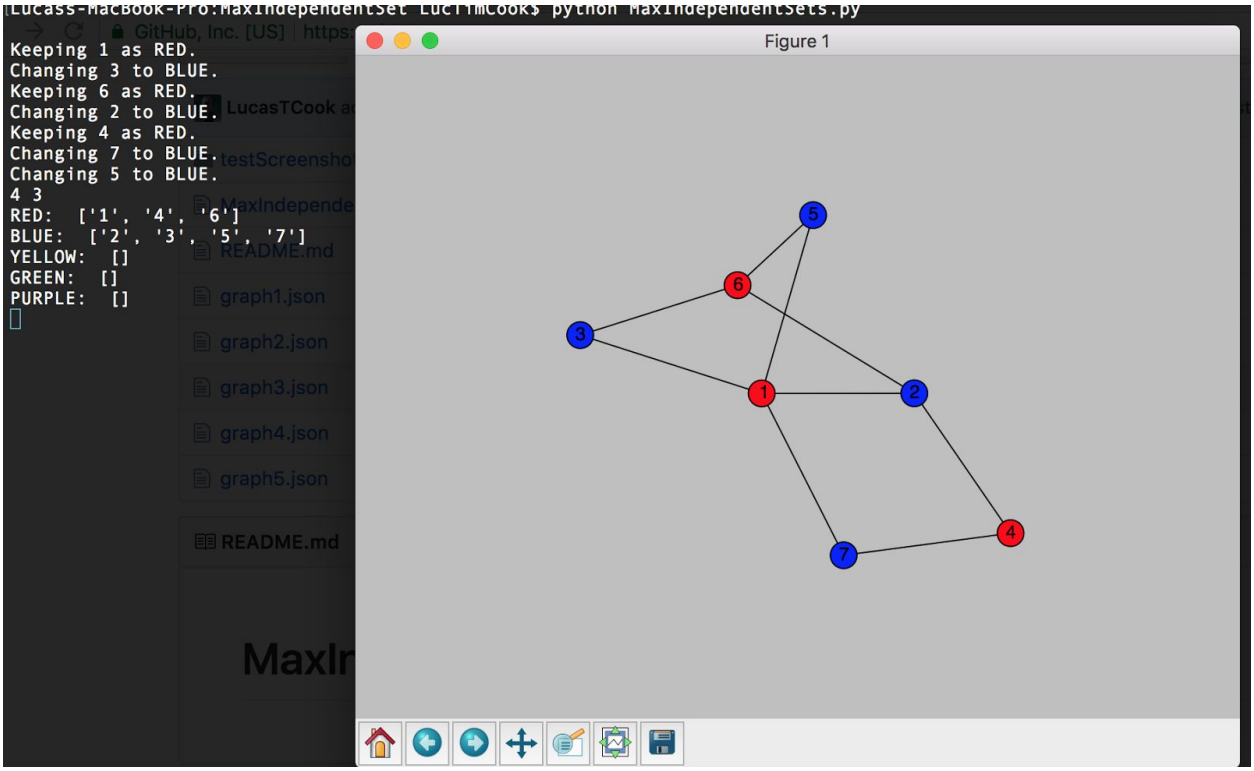
FindMaxIndependentSet() - This algorithm utilizes a recursive depth-first search to traverse through the graph. At each node it changes its color attribute to abide by the maximum independent set rule. Then the function recursively calls itself with one of the nodes that have not been visited from the current node. This requires O(|V|+|E|) , where V is the number of vertices and E is the number of edges.

ColorGraph() - For each vertex in graph must add it to the corresponding color set, which requires O(|V|) time where V is the number of vertices.

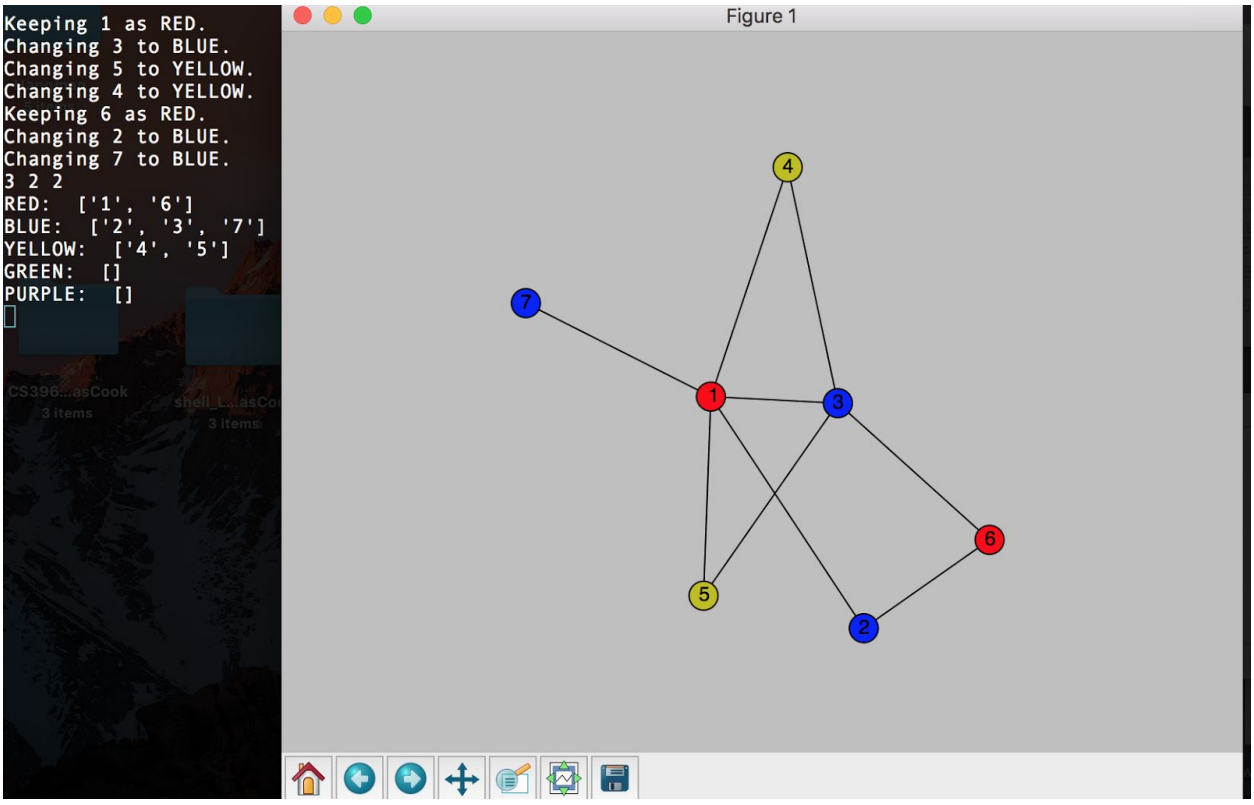
Test Cases

Below is the .json object for every test along with the output of the program.

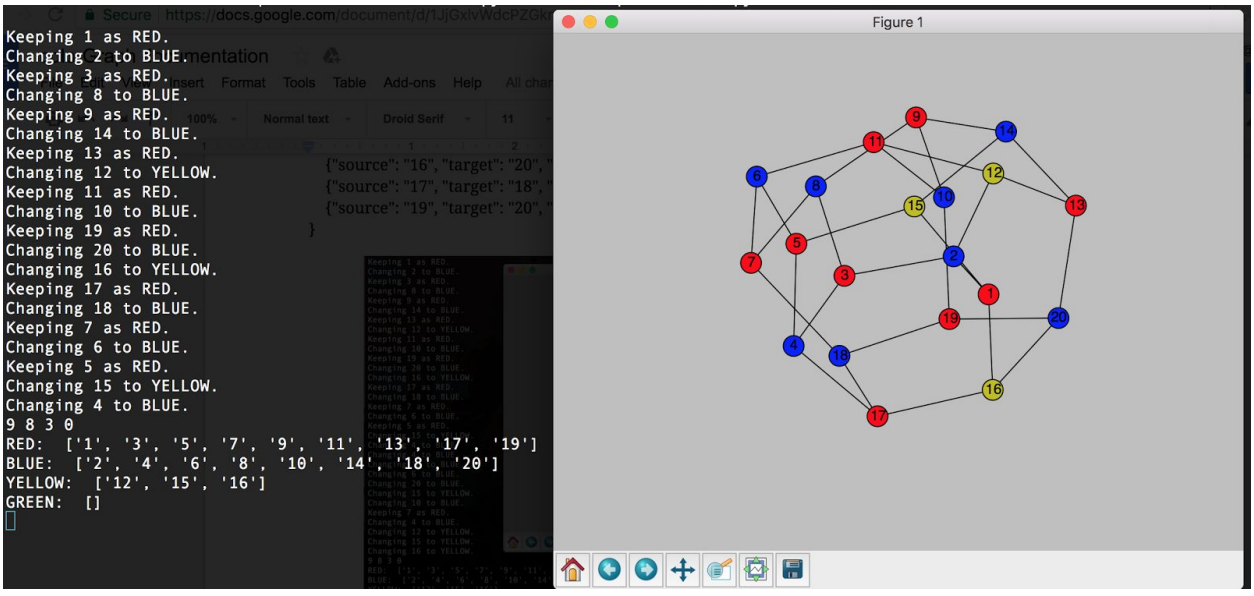

```
{"nodes": [
   {"id": "1", "color": 1, "visited": 0},
   {"id": "2", "color": 1, "visited": 0},
   {"id": "3", "color": 1, "visited": 0},
   {"id": "4", "color": 1, "visited": 0},
   {"id": "5", "color": 1, "visited": 0},
   {"id": "6", "color": 1, "visited": 0},
   {"id": "7", "color": 1, "visited": 0}]
,"links": [
   {"source": "2", "target": "1", "value": 1},
   {"source": "1", "target": "3", "value": 1},
   {"source": "2", "target": "4", "value": 1},
   {"source": "3", "target": "6", "value": 1},
   {"source": "2", "target": "6", "value": 1},
   {"source": "6", "target": "5", "value": 1},
   {"source": "7", "target": "1", "value": 1},
   {"source": "1", "target": "5", "value": 1},
   {"source": "7", "target": "4", "value": 1}]
}
```

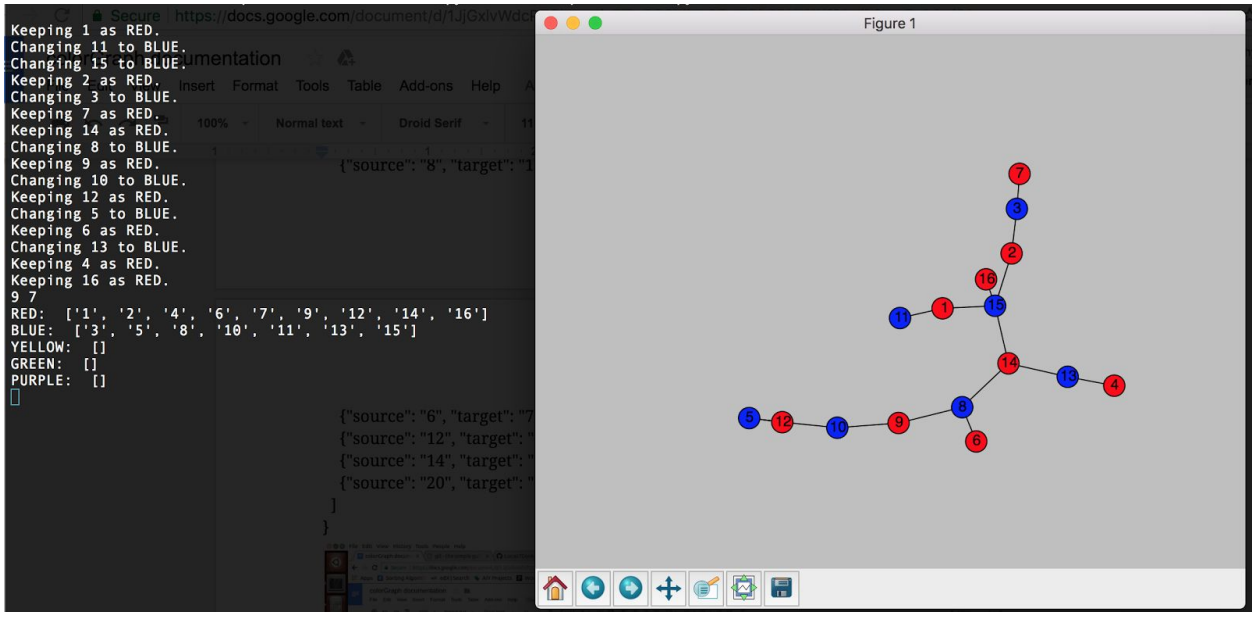
```
{"nodes": [
   {"id": "1", "color": 1, "visited": 0},
   {"id": "2", "color": 1, "visited": 0},
   {"id": "3", "color": 1, "visited": 0},
   {"id": "4", "color": 1, "visited": 0},
   {"id": "5", "color": 1, "visited": 0},
   {"id": "6", "color": 1, "visited": 0},
   {"id": "7", "color": 1, "visited": 0}],
 "links": [
   {"source": "2", "target": "1", "value": 1},
   {"source": "1", "target": "3", "value": 1},
   {"source": "3", "target": "4", "value": 1},
   {"source": "3", "target": "6", "value": 1},
   {"source": "2", "target": "6", "value": 1},
   {"source": "3", "target": "5", "value": 1},
   {"source": "7", "target": "1", "value": 1},
   {"source": "1", "target": "5", "value": 1},
   {"source": "1", "target": "4", "value": 1}]
}
```

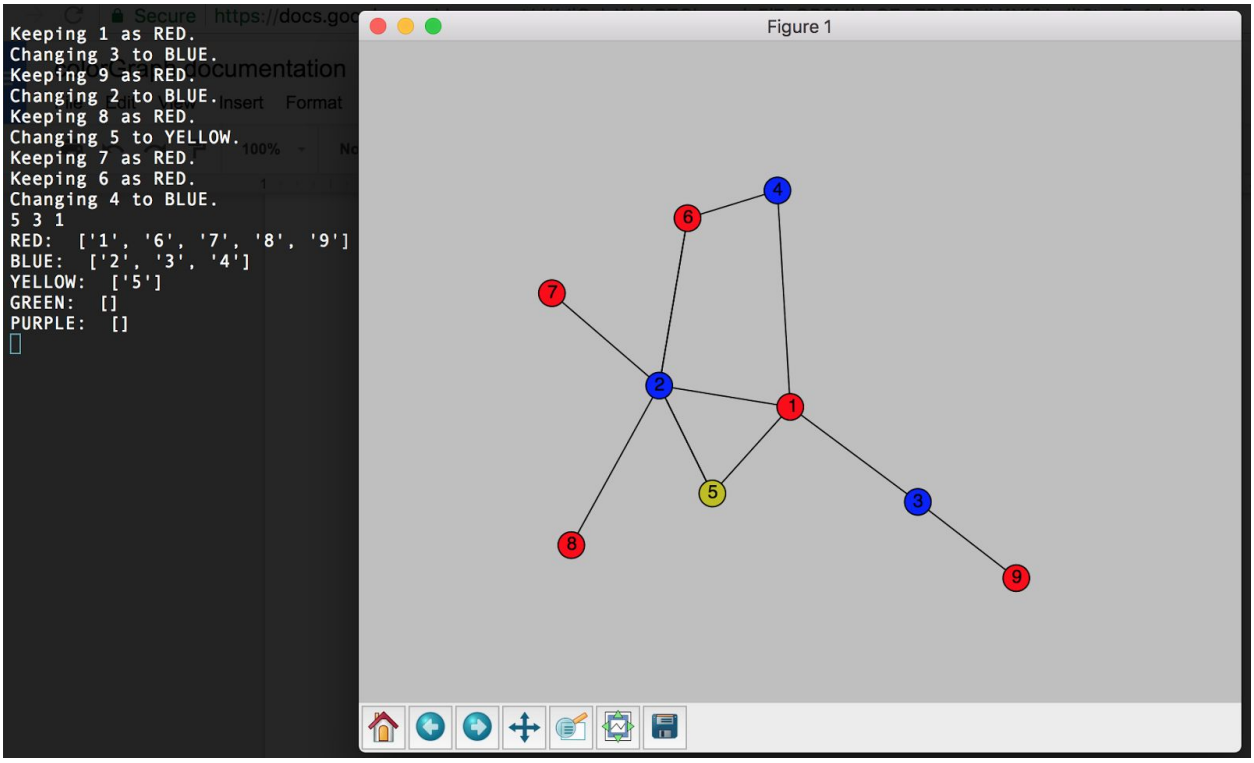
```
{"nodes": [
   {"id": "1", "color": 1, "visited": 0},{"id": "2", "color": 1, "visited": 0},
   {"id": "3", "color": 1, "visited": 0},{"id": "4", "color": 1, "visited": 0},
   {"id": "5", "color": 1, "visited": 0},{"id": "6", "color": 1, "visited": 0},
   {"id": "7", "color": 1, "visited": 0},{"id": "8", "color": 1, "visited": 0},
   {"id": "9", "color": 1, "visited": 0},{"id": "10", "color": 1, "visited": 0},
   {"id": "11", "color": 1, "visited": 0},{"id": "12", "color": 1, "visited": 0},
   {"id": "13", "color": 1, "visited": 0},{"id": "14", "color": 1, "visited": 0},
   {"id": "15", "color": 1, "visited": 0},{"id": "16", "color": 1, "visited": 0},
   {"id": "17", "color": 1, "visited": 0},{"id": "18", "color": 1, "visited": 0},
   {"id": "19", "color": 1, "visited": 0},{"id": "20", "color": 1, "visited": 0}],
  "links": [
   {"source": "1", "target": "15", "value": 1},{"source": "1", "target": "16", "value": 1},
   {"source": "1", "target": "2", "value": 1},{"source": "2", "target": "1", "value": 1},
   {"source": "2", "target": "3", "value": 1},{"source": "2", "target": "12", "value": 1},
   {"source": "3", "target": "8", "value": 1},{"source": "3", "target": "4", "value": 1},
   {"source": "4", "target": "5", "value": 1},{"source": "4", "target": "17", "value": 1},
   {"source": "5", "target": "6", "value": 1},{"source": "5", "target": "15", "value": 1},
   {"source": "6", "target": "7", "value": 1},{"source": "6", "target": "11", "value": 1},
   {"source": "7", "target": "8", "value": 1},{"source": "7", "target": "18", "value": 1},
   {"source": "8", "target": "9", "value": 1},{"source": "9", "target": "10", "value": 1},
   {"source": "9", "target": "14", "value": 1},{"source": "10", "target": "11", "value": 1},
   {"source": "10", "target": "19", "value": 1},{"source": "11", "target": "12", "value": 1},
   {"source": "12", "target": "13", "value": 1},{"source": "13", "target": "14", "value": 1},
   {"source": "13", "target": "20", "value": 1},{"source": "14", "target": "15", "value": 1},
   {"source": "16", "target": "20", "value": 1},{"source": "16", "target": "17", "value": 1},
   {"source": "17", "target": "18", "value": 1},{"source": "18", "target": "19", "value": 1},
   {"source": "19", "target": "20", "value": 1}]
}
```

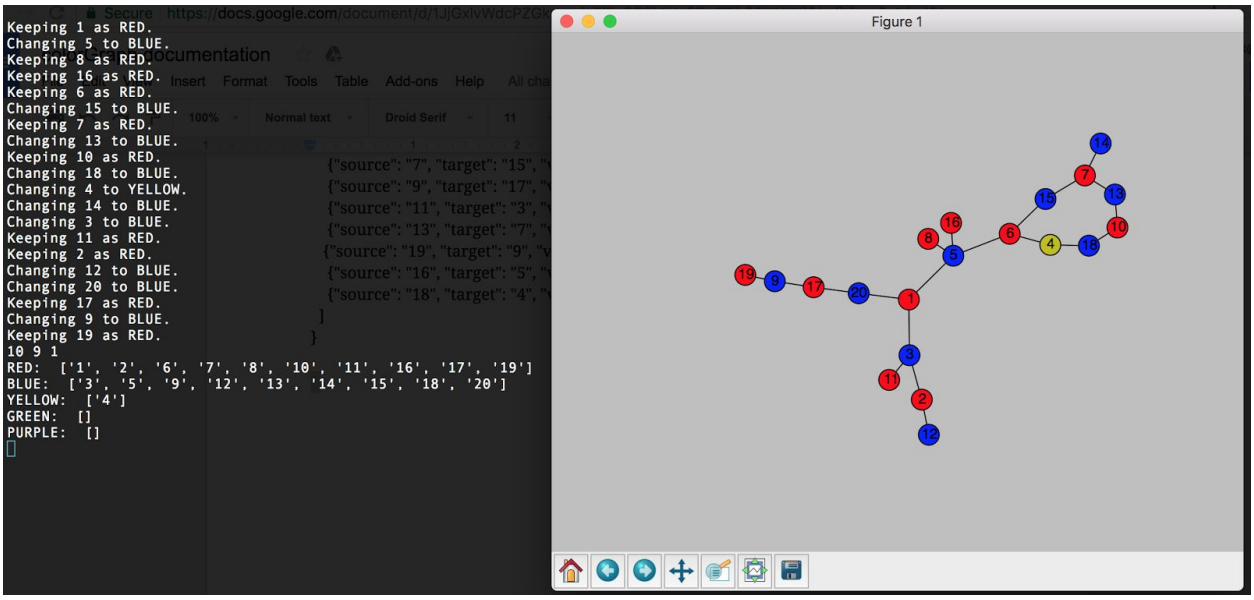
```
{
 "nodes": [
  {"id": "1", "color": 1, "visited": 0},{"id": "2", "color": 1, "visited": 0},
  {"id": "3", "color": 1, "visited": 0},{"id": "4", "color": 1, "visited": 0},
  {"id": "5", "color": 1, "visited": 0},{"id": "6", "color": 1, "visited": 0},
  {"id": "7", "color": 1, "visited": 0},{"id": "8", "color": 1, "visited": 0},
  {"id": "9", "color": 1, "visited": 0},{"id": "10", "color": 1, "visited": 0},
  {"id": "11", "color": 1, "visited": 0},{"id": "12", "color": 1, "visited": 0},
  {"id": "13", "color": 1, "visited": 0},{"id": "14", "color": 1, "visited": 0},
  {"id": "15", "color": 1, "visited": 0},{"id": "16", "color": 1, "visited": 0}
 ],
 "links": [
  {"source": "1", "target": "15", "value": 1},{"source": "15", "target": "14", "value": 1},
  {"source": "14", "target": "13", "value": 1},{"source": "15", "target": "16", "value": 1},
  {"source": "12", "target": "10", "value": 1},{"source": "14", "target": "8", "value": 1},
  {"source": "11", "target": "1", "value": 1},{"source": "12", "target": "5", "value": 1},
  {"source": "9", "target": "10", "value": 1},{"source": "9", "target": "8", "value": 1},
  {"source": "6", "target": "8", "value": 1},{"source": "12", "target": "5", "value": 1},
  {"source": "3", "target": "2", "value": 1},{"source": "2", "target": "15", "value": 1},
  {"source": "7", "target": "3", "value": 1},{"source": "13", "target": "4", "value": 1}
 ]
}
```

```
{
 "nodes": [
  {"id": "1", "color": 1, "visited": 0},{"id": "2", "color": 1, "visited": 0},
  {"id": "3", "color": 1, "visited": 0},{"id": "4", "color": 1, "visited": 0},
  {"id": "5", "color": 1, "visited": 0},{"id": "6", "color": 1, "visited": 0},
  {"id": "7", "color": 1, "visited": 0},{"id": "8", "color": 1, "visited": 0},
  {"id": "9", "color": 1, "visited": 0}
 ],
 "links": [
  {"source": "1", "target": "2", "value": 1},
  {"source": "1", "target": "3", "value": 1},
  {"source": "1", "target": "4", "value": 1},
  {"source": "2", "target": "5", "value": 1},
  {"source": "2", "target": "6", "value": 1},
  {"source": "2", "target": "7", "value": 1},
  {"source": "2", "target": "8", "value": 1},
  {"source": "2", "target": "6", "value": 1},
  {"source": "2", "target": "5", "value": 1},
  {"source": "6", "target": "4", "value": 1},
  {"source": "1", "target": "5", "value": 1},
  {"source": "3", "target": "9", "value": 1}
 ]
}
```

```
{
 "nodes": [
  {"id": "1", "color": 1, "visited": 0},{"id": "2", "color": 1, "visited": 0},
  {"id": "3", "color": 1, "visited": 0},{"id": "4", "color": 1, "visited": 0},
  {"id": "5", "color": 1, "visited": 0},{"id": "6", "color": 1, "visited": 0},
  {"id": "7", "color": 1, "visited": 0},{"id": "8", "color": 1, "visited": 0},
  {"id": "9", "color": 1, "visited": 0},{"id": "10", "color": 1, "visited": 0},
  {"id": "11", "color": 1, "visited": 0},{"id": "12", "color": 1, "visited": 0},
  {"id": "13", "color": 1, "visited": 0},{"id": "14", "color": 1, "visited": 0},
  {"id": "15", "color": 1, "visited": 0},{"id": "16", "color": 1, "visited": 0},
  {"id": "17", "color": 1, "visited": 0},{"id": "18", "color": 1, "visited": 0},
  {"id": "19", "color": 1, "visited": 0},{"id": "20", "color": 1, "visited": 0}
 ],
 "links": [
  {"source": "1", "target": "20", "value": 1},{"source": "2", "target": "3", "value": 1},
  {"source": "3", "target": "1", "value": 1},{"source": "4", "target": "6", "value": 1},
  {"source": "5", "target": "1", "value": 1},{"source": "6", "target": "5", "value": 1},
  {"source": "7", "target": "15", "value": 1},{"source": "8", "target": "5", "value": 1},
  {"source": "9", "target": "17", "value": 1},{"source": "10", "target": "18", "value": 1},
  {"source": "11", "target": "3", "value": 1},{"source": "12", "target": "2", "value": 1},
  {"source": "13", "target": "7", "value": 1},{"source": "14", "target": "7", "value": 1},
  {"source": "19", "target": "9", "value": 1},{"source": "15", "target": "6", "value": 1},
  {"source": "16", "target": "5", "value": 1},{"source": "17", "target": "20", "value": 1},
  {"source": "18", "target": "4", "value": 1},{"source": "10", "target": "13", "value": 1}
 ]
}
```

```
{
 "nodes": [
  {"id": "1", "color": 1, "visited": 0},
  {"id": "2", "color": 1, "visited": 0},
  {"id": "3", "color": 1, "visited": 0},
  {"id": "4", "color": 1, "visited": 0},
  {"id": "5", "color": 1, "visited": 0}
 ],
 "links": [
  {"source": "1", "target": "2", "value": 1},
  {"source": "1", "target": "3", "value": 1},
  {"source": "1", "target": "4", "value": 1},
  {"source": "1", "target": "5", "value": 1},
  {"source": "2", "target": "3", "value": 1},
  {"source": "2", "target": "4", "value": 1},
  {"source": "2", "target": "5", "value": 1},
  {"source": "3", "target": "4", "value": 1},
  {"source": "3", "target": "5", "value": 1},
  {"source": "4", "target": "5", "value": 1}
 ]
}
```