

William Briguglio

READ ME:

To provide different color sets and maps to my program, assert two facts:

1. `color(["red", "blue", ..., "green"])`
-a list of colors
2. `graph([[a,b,c],[b,a],[c,a]])`
-a list of list where the head of each sub list is a vertex, and the tail contains the vertices adjacent to the head

For convenience, the example graph is already hardcoded into my program as a fact.

To try different graphs with different colors sets, delete the `graph()`. and `color()`. fact.

Below is a screen shot of my program executing, note the long execution time.

```
1 graph([[a,b,c,d,f],[b,a,c,e,f],[c,a,b,d,f],[d,a,b,c,e,f],[e,a,b,c,d,f],[f,a,b,c,d,e,f]])
2 colors(["blue", "red", "green", "yellow"])
3
4 color(X):- %True if X is in the provided colors
5     colors(Y),
6     member(X, Y).
7
8 node(X):-%True if X is in the provided graph
9     graph(Y),
10    member(X, Y).
11
12 vertex_aux(X):- %True if X is an element of the graph
13     node(Y),
14     member(X, Y).
15
16 vertexs_aux(R):- %True if R is a List of vertices
17     findall(X, vertex_aux(X), Y),
18     Y==R.
```

result(X).

X = [[f, "green"], [e, "blue"], [d, "red"], [c, "green"], [b, "red"], [a, "blue"]]

50.650 seconds cpu time

?- result(X).

Examples | History | Results | Run!

Notes:

-The complex method to obtain edges was to ensure that there were no two identical edges that were just the reverse of each other (i.e. [a,b] and [b,c]), as this increased the run time by a massive amount