Albert Chieu

CSC 220

7/28/11

m-Coloring Discussion

**What is the problem?**

Use the Backtracking algorithm for the m-Coloring problem to find all possible colorings of a given undirected graph.

**What is m-Coloring?**

m-Coloring problem for an undirected graph means how to color a vertex of an undirected graph using any color from a given set of defined colors such that no two adjacent vertices share the same color.

**What is backtracking algorithm?**

Backtracking algorithm is a way to try any subsequent given paths until a solution is found else back track to the previous point and retry a different path.

**How the program is implemented?**

The program starts by creating an undirected graph with a given number of vertices and using a random number generator to define some random connections. An array of three colors (Red, Green, and White) is defined to be used for coloring. The undirected graph is defined by an adjacency matrix with (i, j) = 1 means connected and 0 means not connected.

Starting with a given color at vertex 0, the algorithm will iterate through all possible colors by doing the following steps.

1. Check if the current color is promising for the current vertex. This means to check whether the color has no collisions with the color in any connected vertices.
2. If the color is promising, then
	1. Check whether all vertices have been visited with color, if so then return the graph.
	2. If not yet done, try all possible colors for the next vertex by repeating step 2.
	3. If all possible colors have been tried and no promising color is found for the vertex, then backtrack to the previous vertex.
3. If it goes back to the original vertex 0, then there is no solution.

**Generally how well does it do?**

The backtracking algorithm on m-coloring does not give all possible solutions. However, it will guarantee to find an available solution if there is possible solution(s).