Assignment 11. **Graph, spanning trees, etc.**

Given April 20, due Monday, April 27.

1. We have an undirected graph, G, defined by giving an edge list for each vertex. We assume that the list is symmetric, that is if there is an edge from v to u, then there is also an edge from u to v. We also suppose that each vertex has a color, given by `v.color`. For each positive integer, d, the number of vertices at distance d from the source, s, is $v(d)$. The number of edges that contain such a vertex is $e(d)$.

a. Modify the BFS search algorithm to print out the values $v(d)$ and $e(d)$.

b. Sketch a routine, `near_color( s, color)`, that finds the vertex of the given color nearest to the vertex, s. If the distance is d, the algorithm should have work on the order of $e(d)$.

c. Suppose instead that the edges are weighted and that we want to find the nearest vertex of the specified color using weighted distance. Sketch a routine that does this in work fo order $e_w(d_w)\log(e_w(d_w))$, where d_w is the weighted distance and e_w is the number of edges within the weighted distance.

2. We have a connected undirected graph, G, with all edges marked either red or blue (but not both). Let G_b be the subgraph having the same vertices but only the blue edges. The connected components of G_b are C_1, …, $C_l l$. The "reduced graph" has one vertex for each of the C_k and an edge from C_k to C_j if there is a red edge in G that goes from a vertex in C_k to one in C_j.

a. Suppose that C_k is connected to C_j in the above sense, and that $v \in C_k$, and $u \in C_j$ are two vertices of G. Show that there is a path in G from v to u that visits only vertices from C_k and C_j.

b. Show that, conversely, if there is a path from v to u as in part a., then the component containing v and the component containing u are connected in the reduced graph.

c. Sketch an algorithm that constructs the reduced graph (makes an array of vertices and edge lists for the vertices), using the set operations `makeset`, `find`, and `union`.

3. G is an undirected graph with positive edge weights. The weight of a path is the sum of the weights of the edges in the path. Sketch an algorithm that finds the the minimum path weight between edges s and v among all paths of length not more than L (a given value).