Identification:

Rian Neufelt a00723442

unrealking@hotmail.com

Algorithm Design:

input:

Using my 'reader' object I read the lab data input into a 2d ArrayList for each lab, for each test case. I then read in the pre placed cable data and create a graph (represented by an adjacency matrix) for each test case.

Part1:

To solve the minimal spanning tree problem I implemented Kruskal's algorithm. My 'Kruskal' object takes in the 'half' completed graph(pre placed cable), a Distance matrix which represents the distance from one lab(vertex) to another, eg. Distance[1][2] is the distance from 1 to 2.

Kruskal has 2 priority queues, one for use in the algorithm using the weight(length) of the cable for comparisons, and the other using the source then destination for comparisons is used for printing the needed cables. Minor changes were made to the algorithm to account for the pre placed cable, during the set initialization I union the sets of cable already placed(they're already on the solution graph).

My implementation of Kruskal's algorithm makes use of disjoint subsets which I have defined in my 'DisjointSet' object as an ArrayList of HashSets. Find, union, and makeSet are all implemented to make this algorithm work.

Part2:

To solve where to place the data-center I used Dijkstra's algorithm on every vertex to every other vertex and recorded the longest distances for each vertex, picking the smallest 'longest' distance for the data-center location.

To make Dijkstra's algorithm work for this problem I had to first convert my graph from a directed graph to an undirected graph, then because we're concerned with the number of hops and not the total distance I added one instead of the weight.

Results:

Mostly just tests based on the sample input. More thorough testing done with the ParsingValidator to test out of memory/time functionality. Based on that I am confident it will fail gracefully when given ridiculously large test data. Logical wise I am certain that everything is implemented correctly and that given a workable(expected to complete within the time limit) set of data it will return the correct answers.