Robert Owens (rao7utn)
Wiring (Basic HW Graphs)

This problem requires that one find the total cost of a minimum spanning tree (MST) on a graph that models the electrical wiring of a house where vertices are call junctions and have types Breaker Box, Electrical Outlet, Junction Box, Switch and Light. Edges between these junctions are undirected and weighted intending to model wires. Additional there are restrictions on which junctions can be connected each other which are outlined in detail in the assignment.

My approach is solving this problem is as follows:
First, validate the input and importantly disregard any edges from the input that are not possible with the constraints defined by the problem.
Second, construct one MST for a subset of junctions of only Breaker Boxes, Electrical Outlets, and Junction Boxes and a subset of edges where only junctions from the previously described subset are vertices that comprise the edge. The resultant MST is an MST of all vertices that can be directly connected to the breaker. By similar methods I then construct a second MST with a subset of vertices Lights and Edges and subset of edges where the vertices involved in the edge can only be Lights and Switches. To construct these MSTs Kruskal's algorithm is implemented with the disjoint set data structure and this disjoint set data structure implements path compression and union by rank implementations to achieve the best overall runtime for Kruskal's algorithm.
Third, to finish constructing the MST for the entire graph my algorithm connects the first two MSTs through edges between switches and the first MST (only junctions that can be directly connected to the breaker). To do this I create an empty set that will store switches which have been connected and have a priority queue to examine the most cost edges first. Until the initially empty set has the same size as the number of switches in the whole graph my algorithm considers an edge and adds it to the graph if the switch involved in this edge is not an element of the set, and then add it to the set. This ensures that only 1 wire connects to every switch and that for every switch the minimum cost valid edge is selected.
At this point the MST for the graph is completed and all that is left to do is compute the sum of all edge weights in the MST and output that sum.

This algorithm runs in $\theta(E \cdot log(V))$ time since reading input requires $\theta(V + E)$ time, running optimized Kruskal's algorithm requires $\theta(E \cdot log(V))$ time, hooking the switches across the two MSTs to complete the total MST requires:$\theta(E)$ time and then finally computing the total cost of the complete MST requires $\theta(E)$ time. Hence, Kruskal's algorithm will dominate the runtime resulting in the final runtime for this algorithm being $\theta(E \cdot log(V))$.

The grader may look at the files of code uploaded with this submission and thus no pseudocode is required here.