Summary Report

Prelab and Lab 1 were the introductory to the class server, terminal, and the cad tools. Lab 2 is somewhat completely different, we had to generate a netlist parser, clique-based graph generator, and then the Kernighan-Lin algorithm. Each part requires formatting and coding that is needed to obtain information and then rearrange it to generate the algorithm. Using a programming language to generate each part of the lab is required.

Lab 2 Part 1

Generating the netlist parser from a given hyper graph file is needed to obtain information. Using c++ to read in a file to obtain the area constraints and then the nodes of the graph. This is then generated to the clique-based graph. The clique-based graph generates set of every two distinct vertices that are adjacent in the graph. Given the initial partition file it can then generate the Kernighan-Lin algorithm.

I had the hardest time to generate anything with the given data. Trying to parse in C++ was something I regret doing, but to give an insight on what I tried to do I’ll be brief. I could get the information from a different file. Getting the parameters of the graph such as the number of nodes and the number of netlists were easy to get from the file line. After getting the first line, getting the other information about each node was a little difficult because it goes:

(node #) << connected to node [#] …

So at this point using a data structure is necessary to contain all of this information. So I generated an array of vectors. The number of arrays is generated by the number of nodes and then the connects are the vectors within the arrays. so after getting this contain the information. I tried to make coordinates of each node connected to the other node(s). At this point I was going to reduce any redundancy to use the information to get around the edges of each node. But at this point I got carried away on trying to parse correctly cause in C++ things are not easier to develop data structures or converting certain data types. I generated over 200 errors and then got lost in trying to develop the second part of this lab.

Lab 2 Part 2

The program takes in the three inputs (hyper graph file (information), the maximum number of passes, and the initial partition file). This enables to compute and output the information for each node, each edge, total weight, total cut size, runtime for reading/writing files, and computational time. So I’m going to be honest about this lab I got too lost and what I was trying to develop before even using the data. I know that next time I need to address the professor for additional help or a colleague. I was completely lost on how to get information to become parsed.