Algorithm PA3 report
b07901088 電機三 陳秉嘉

First, consider the undirected weighted and unweighted graph, we can solve it by using Kruskal's algorithm to find the maximum spanning tree so that we can get the minimum deleted edge weight.

As for the data structure, I use the class Graph and the standard vector in C++ to store the adjacent list and implement the Kruskal's algorithm by using self defined class Tree and Vertex.

Then, for directed graph, we should seem the directed graph as undirected graph and run the Kruskal's algorithm once. And we can get the deleted edges list, then if the edge weight is positive in deleted edges list, we try to add this edge back to the graph and check whether it would create a cycle. If yes, we remove this edge again or add back to graph and remove it from the deleted edges list. The order to add edge back to graph is significant to effect the result, so I choose add them back in decreasing order. After doing above process for every positive weight edge, the edges in deleted edge list are the answer for directed graph.

The data structure for directed is basically the same as undirected graph.

Finally, It work properly in my own laptop, but when I upload to work station, it give me another answer ><, I have no idea for the same code but get different answer. I think it is the problem of the environment and I try my best to fix it but it seems my effort is wasted....