Algorithm 2022 spring
PA3 Report
B07502071 陳志臻

1. Method
   I used a n*n metric weights[n][n] to restore the weights within every two edges, which n is the number of vertices. And I also used another metric PI[n][2] to restore the states including a vertex's predecessor and whether it's selected. I used the same data structure for both undirected and directed graphs. Besides, I used some constants to define each situation. LIMIT is the minimum value of weight -100, BOUNDARY is a value less than LIMIT which means no edge between the two edges, OUB(out of boundary) is a value less than BOUNDARY, and INFINITY is an extremely large number. Then I run Prim's algorithm to reweight the graphs for n times and find out the maximum spanning tree. For directed graphs, I used BFS everytime to check whether there was a cycle when I added an edge to the tree. If it formed a cycle, the edge was what I was going to remove. Finally, I summed up the weights to get the total removed weight and returned.

2. Conclusion
   a. Using arrays as the main data structure could be a little faster than using vectors.
   b. For undirected graphs, we just find the maximum spanning tree then we can get the removed weights.
   c. For directed graphs, they are NP-complete problems.
   d. The Algorithm is really too hard to fully understand in one semester. I still need more practice.