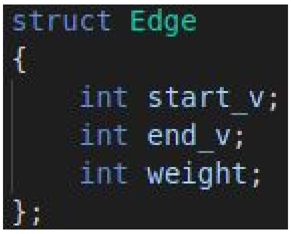
Algorithm programming assignment#3 report

B07611033 電機三 林書宇 alg20177

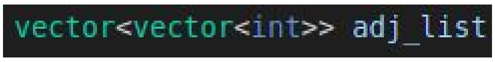
1. data structures used in my program:

in order to store the properties of an edge, I declared a structure called "Edge", which contains three integers: start_v, end_v, weight, respectively represents the first vertex and the second vertex listed in the input, and their corresponding weight.

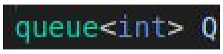


```
struct Edge
{
    int start_v;
    int end_v;
    int weight;
};
```

for the use of BFS, I constructed a two dimensional vector for the adjacency list, and a queue called "Q" to implement BFS

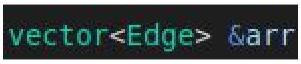


```
vector<vector<int>> adj_list
```



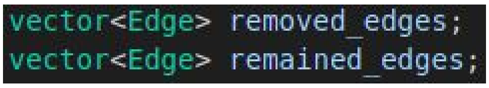
```
queue<int> Q
```

for the implementation of kruskal's algorithm for undirected graph, I used a vector of Edge to get the properties and run the algorithm



```
vector<Edge> &arr
```

Finally, I used two vectors of Edge to store the remaining edges in the graph and the removed edges.



```
vector<Edge> removed_edges;
vector<Edge> remained_edges;
```

2. My findings in this programming assignment:

the undirected case is easy to get the optimal solution, however, the directed case is definitely not. As indicated in the pa3.pdf, the problem of cycle breaking for a weighted directed graph is a NP-hard problem and is also known as minimum feedback arc set problem, there are several ways to try to push the solution of this problem to optimal, some of my classmates used the same algorithm, but result in obtaining different results, this may happen since the selection sequence of edges may be different. Classmates using different algorithms also results in solutions with different sum of weights.