

演算法PA3 Report

學生：林咏毅

學號：b10901059

- **Data Structure**

- edge：紀錄graph中，邊的資訊，如：頭、尾和權重。
- vertex：紀錄節點資訊，如：編號、距離和起始點。
- compare_edge：定義priority queue的sorting方式。
- util_function：定義輔助用函式。

```
1 struct edge{
2     int head, tail, weight;
3     edge();
4     edge(int &_amp;_head, int &_amp;_tail, int &_amp;_weight);
5 };
6
7 struct vertex{
8     int index, distance, parent;
9     vertex();
10    vertex(int _index);
11 };
12 // define priority queue sorting method
13 class compare_edge{
14     public:
15         bool operator()(edge &a, edge &b){
16             return a.weight < b.weight;
17         }
18 };

```

```

1 struct util_function{
2     void print_edges(vector<edge> &edges);
3     void print_vertices(vector<vertex> &vertices);
4     void print_priority_queue(priority_queue<edge, vector<edge>, compar
5     void print_adj_matrix(vector<vector<int>> &adj_matrix);
6
7     int trace_root(vector<vertex> &vertices, int vertex_index);
8     // compare distance between two vertices
9     void adjust_parent(vector<vertex> &vertices, int index1, int index2
10    void DFS(int head, int tail, vector<vector<int>> &adj_matrix, vecto
11    int DFS_traversal(int head, int tail, vector<vector<int>> &adj_matr
12 };

```

• Observation

- 計算 undirected 和 directed 方法相似，差別在於當graph為directed時，我們可以將已拿出之邊加入maximum spanning tree，以此拿到最小total weight。
- 利用不同方式計算test_case_8時會有不一樣結果，原因為時間複雜度不同。
- 使用動態陣列儲存資訊較使用STL套件更快速。
- 寒假快樂！