Algorithm Programming Assignment

B10901027 楊竣凱

1. algorithm

   To achieve minimum cost we need to select edges in descending order, and use Kruskal's algorithm to form a spanning tree.

   (1). sort edges by radix sort

   (2). from largest to smallest weighted edge, add into tree(or discard it) by some condition

   for undirected graph and directed graph, there are different conditions

   - undirectred graph:
     - as long as two end vertices are not in the same group, we can add the edge (won't form a cycle)

     => using disjoint set with path compression and union by rank

     => O($\boldsymbol{\alpha}$(V)) for each step
   - directred graph:

     we use a path matrix to store whether there is a path (paths[i][j] is true when there is a path from i to j) and disjoint set to store whether they are connected

     1. check whether it will form an edge => O(V)
     2. if yes, discard it
     3. else if two ends are in the same set (already be a spanning tree)
        a. if weight< 0 then there is no need to add a negative edge between them, because it can reduce out cost
        b. but if it is positive, as long as it won't form a cycle, we can add it
     4. to add an edge
        a. We need to update all vertices that have a path to the start end ($\exists$ i→start end), for each vertex it need to add every vertex that the stop end can reach into its path => O(V^2)

           (if i→**start end** and **stop end**→j, then after this step i→j)
        b. merge two ends into the same group => O($\boldsymbol{\alpha}(V)$)
2. time complexity
   - undirectred graph:
     - radix sort => O(E)
     - initialization of disjoint set => O(V)
     - for each edge, some operation with disjoint set => O(E$\boldsymbol{\alpha}(V)$)

     total => O(E+V+E$\boldsymbol{\alpha}(V)$) = O(V+E)
   - directred graph:
     - radix sort =>O(E)
     - initialization of disjoint set and path matrix => O(V+V^2) = O(V^2)
     - for each edge, some operation analysed above => O(E(V^2+$\boldsymbol{\alpha}(V)$)) = O(EV^2)

     total => O(E+V^2+EV^2) = O(EV^2)

3. running result

   using local machine:

   Intel® Core™ i7-8565U CPU @ 1.80GHz

   Ubuntu 20.04.6 LTS

case	type	total weight of removed edges	running time(ms)
public_case_1	u	21	5.882552
public_case_2	u	-3330	3.965029
public_case_3	d	-21468	4.953965
public_case_4	u	0	3.369494
public_case_7	d	-10515	3.141739
public_case_8	d	-70938	75.361813