Algorithm PA3 Report

B07901047 范育瑋

(I) Data Structures

Class CycleBreaker:
The main solver that will read in input file, solve the cycle-breaking problem and output to an output file specified.

Struct Subset:
Subset is simply an array of integer, representing disjointset, where arr[i] = (-2) means that node i is the representative of such subset and the size is 2. arr[i] = 2 means that node i is in the same subset of node 2.

Struct Edge:
Edge is a data structure contains three uint x, y and w representing e=(x,y) where weight(e) = w.

(II) Algorithm

In this homework, I modified the DFS algorithm to detect cycles in directed graph. We perform DFS on the original graph, whenever see a back edge, use the predecessor array to trace the cycle. During tracing, we dynamically maintain a map<Edge*, list<unsigned>> edge2Cycle that each edge key corresponds to a list of cycle that contain the edge. For example, cycle1 = A ~> B ~>A , cycle2=A~>B~>C~>A, then E(A,B) = [cycle1, cycle2], E(B,A) = [cycle1]…

For each iteration, we use the edge2Cycle to determined how to remove edges. A heuristic is use that for each edge, we choose the edge that has the smaller cost = weight(e)/#(cycles in the list). Notice that the meaning of the cost is the “Average cost of each cycle ” when breaking such edge. Therefore, we should minimize the cost. Whenever we remove the edge, we properly remove all the occurance of cycles that contain the edge until each list in edge2Cycle is empty.

We need to further perform DFS to detect cycles and break the cycles until there is no cycle.

The last step of the algorithm is to generate a Maximum-Spanning-Tree in the directed acyclic graph acquired. I use modified Kruskals's MST algorithm that when adding edge, if such edge is negative and the two endpoint is in the same set, simply removed it.

(III) Finding

When coping with difficult problem such as Cycle-Breaking, a NP-Hard problem, what we need is to try as more heuristic as possible and finding out some special structure which may be useful in some instance. I really have fun in this PA！！