

PA3 Report

B07901032, Kuo Hao Wang

I. Data structure, variables

- `class edge`

This class store three integer denoting source, destination, and weight.

- `class disjointSet`

This class is the implementation of disjoint set on the textbook.

- `class solver`

This class is served as base class for undirected and directed graph solver that has two integer denoting amount of vertices and edges and a Boolean denoting if the graph is weighted. Though we don't have different implementation for weighted and unweighted.

- `class usolver : solver`

This class is solver for undirected graph using Kruskal's algorithm on finding maximum spanning tree. The maximum spanning tree is the complementary of minimum feedback arc set. Thus, it contains an array of edge and an array of bool indicating if the relating edge is in minimum feedback arc set.

- `class dsolver : solver`

This class is solver for unweighted directed graph using the concept presented in *A fast and effective heuristic for the feedback arc set problem*. There are two adjacency matrix of directed graph named `in` and `out` and a `deque` of integer named `s` denoting the vertices sequence in the reference. To put it simple, here is the pseudo code presented in reference.

```
1  Procedure GR(G, s)
2  {
3      s_1 = empty set
4      s_2 = empty set
5      while(G is not empty)
6      {
7          while(G contains a sink)
8          {
9              choose a sink u
10             s_2 = concatenation(u, s_2)
11             G = remove(G, u)
12         }
13         while(G contains a source)
14         {
15             choose a source u
16             s_1 = concatenation(s_1, u)
17             G = remove(G, u)
18         }
19         if(G is not empty)
20         {
21             choose vertex u which delta(u) is maximum
22             s_1 = concatenation(s_1, u)
23             G = remove(G, u)
24         }
25     }
26     s = concatenation(s_1, s_2)
27 }
```

II. Programming Flow(main.cpp)

Read in the input file and store to `_type`, `v`, `e`

→ Read in the input file and add edges to graph using `solver(s, d, w)`

→ solve the minimum feedback arc set problem by `solver.solve()`

→ get result by `solver.result()` and send into output file

III. Experiments

- Undirected graph tested on 40055

V	E	Real Time (s)	User Time (s)	Max mem (byte)
10	30	0	0	1488
50	300	0	0	1568
10	9	0	0	1456
5000	10000000	5.63	5.31	509684

- Directed graph

V	E	Real Time (s)	User Time (s)	Max mem (byte)
100	1000	0.06	0	1448
30	870	0.02	0	1420
500	3000	0.03	0	1584

Since the algorithm is aimed for unweighted directed graph, for weighted instances it remains the correctness but no quality. However, it is quick in execution time ($O(|E|)$) and easy to implementation.

IV. Reference

- Introduction to Algorithms, third edition. Ch21 Data Structure for Disjoint Sets, Ch23 Minimum Spanning Trees.
- P. Eades, X. Lin, W.F. Smyth, A fast and effective heuristic for the feedback arc set problem, in: Information Processing Letters Volume 47 (1993) 319-323.