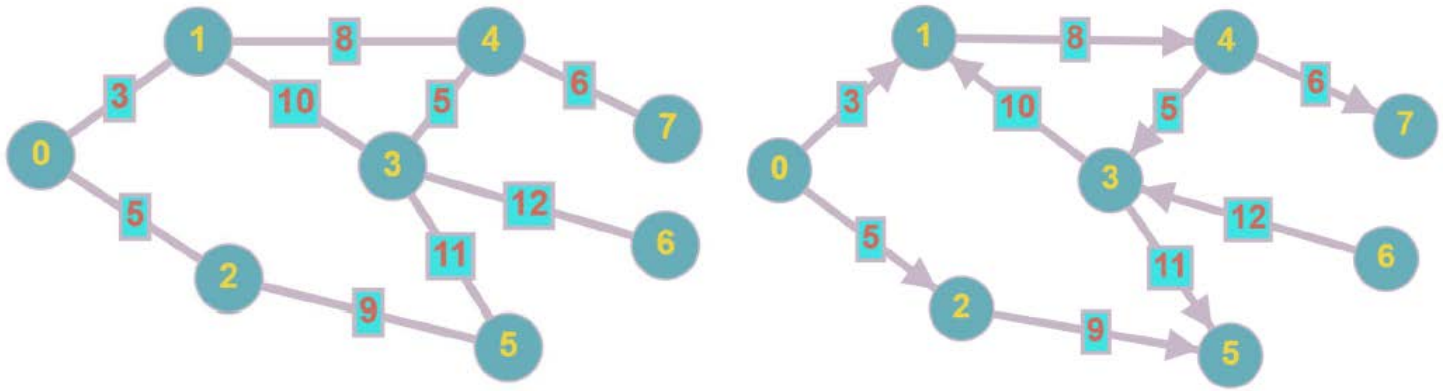
Recitation: Week 15

EE4033 Algorithms, Fall 2019
Instructor: Yao-Wen Chang, James Chien-Mo Li, and Iris Hui-Ru Jiang

Presenter: Shang-Chien Lin, Yi-Ting Lin

Programming Assignment #3

- Given
  - A graph $G = (V, E)$ that might contain cycles
- Objective
  - Find a set of edges with minimum total weight that if these edges are removed from the graph, the graph will become acyclic



(a) weighted undirected graph (b) weighted directed graph

Figure 1: Graph with cycles.

Instance Type

- unweighted undirected graph
  - All edge weights equal to 1
- weighted undirected graph
  - General case
  - Including positive/negative/zero edge weights
- weighted directed graph
  - Edges are directional
  - Minimum feedback arc set problem
  - NP-hard problem
  - You don’t need to find the optimal solution

Input Format

- First line
  - A character 'u' or a character 'd' indicates the input graph is an undirected graph or a directed graph
- Second and third line
  - The total number of vertices n and edges m
  - The index of the vertices will be continuous from 0 to $n - 1$
- m edges
  - Each line contains three integers i, j and w, denoting an edge from vertex i to vertex j with weight w
- A single 0 in the input line signies the end of input

Output Format

- First line
  - The total weight of removed edges to make the input graph acyclic
- List of these removed edges and their weights
  - The output edges can be in arbitrary order
  - The order of i and j can be different from the input for undirected graph
- If the input graph has no cycles, you should output a line with a single 0 in your output file

Important Assumptions

- The input graph has only one connected component
- The output graph (after all the reported edges have been removed) should remain connected
- For undirected graph instances
  - $1 \leq n \leq 100000$
  - $1 \leq m \leq 50000000$
- For directed graph instances
  - $1 \leq n \leq 5000$
  - $1 \leq m \leq 10000$
  - There might be some bonus cases with larger sizes

Command-line Parameter

- The executable binary should be named as **cb**
- Command format

```
./cb [input_file_name] [output_file_name]
```

- Example

```
./cb public_case_1.in public_case_1.out
```

Checker

- A checker is provided for you to verify your results
  - A binary file that can be executed on Linux systems
  - It will check if your output edges are from the input set, if your resulted graph is connected, and if your resulted graph contains cycles

- Usage

```
./pa3_checker [input_file_name] [your_output_file_name]
```

- Example

```
./pa3_checker public_case_1.in public_case_1.out
```

Evaluation

- Runtime limit for each case: 60 seconds

- Eight public cases are provided
- More hidden cases will be used in the final test
  - unweighted undirected graph: 20%
  - weighted undirected graph: 50%
  - weighted directed graph: 30%
  - The score is determined by correctness for undirected graph instances
  - The score is determined by correctness and quality (total weight) for directed graph instances

- Evaluate on one of the EDA Union servers
  - EDAU1, EDAU5, EDAU8, EDAU15

Required Files

- Create a directory named **<student ID>_pa3/** (e.g. b06901000_pa3/)
  - A directory named **src/** containing all your source codes, and no directories in src/
  - A executable binary named **cb**
  - A makefile named **makefile or Makefile**
  - A text readme file named **readme.txt**

- Compress your directory into a *tgz* file named **<student ID>_pa3.tgz** by the following command:

```
tar zcvf b06901000_pa3.tgz b06901000_pa3/
```

- Use **checkSubmitPA3.sh** to check if your submission satisfies all the requirements!

```
bash checkSubmitPA3.sh b06901000_pa3.tgz
```

Submission

- Submit your **<student ID>_pa3.tgz** to the NTU COOL system before
  1pm, Dec. 29, 2019 (Sunday)

- Penalty for late submission: 20% per day

- All submissions will be subject to duplication checking

Tips

- What is a undirected graph without cycles?
- How to deal with difficult optimization problems?
  - Develop efficient heuristics
  - Greedy/local search methods
- If you have any questions about the problem, please email TA Shang-Chien Lin at r07943106@ntu.edu.tw
- If you have any questions about the checker, please email TA Yi-Ting Lin at r07943102@ntu.edu.tw