Homework #5

Kruskal's Algorithm

Yunmin Go

School of CSEE

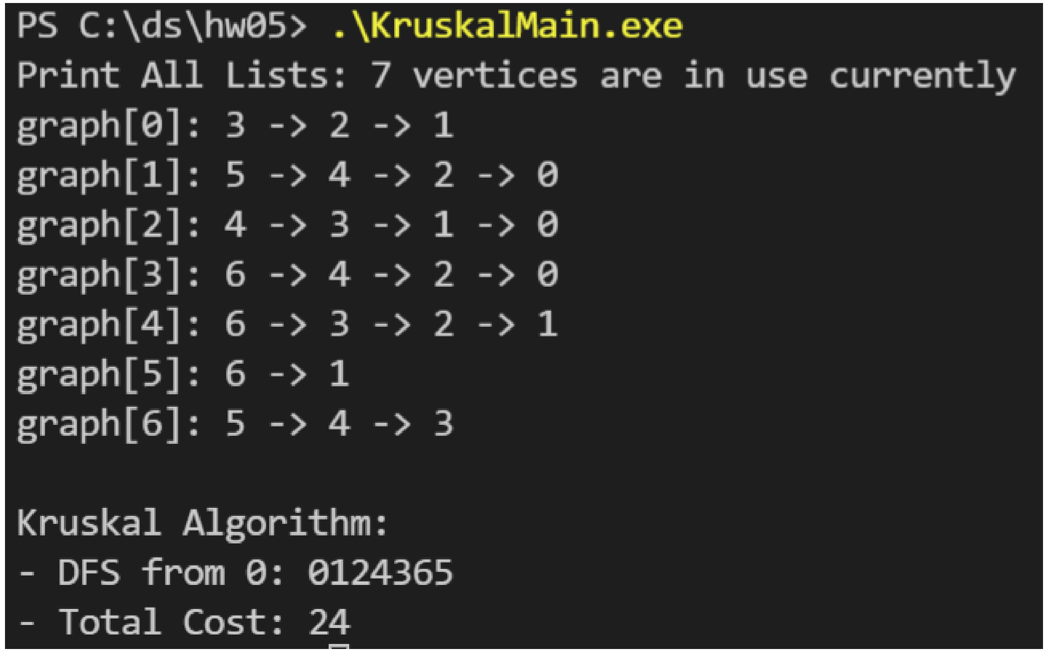
HANDONG GLOBAL UNIVERSITY

HW#5

- Write a member function of Graph class that finds a minimum cost spanning tree using Kruskal’s algorithm.
  - You can modify Graph.cpp we implemented in Practice #11.
  - We assume undirected graph with weight.
  - Use adjacency list for graph representation.
  - Make a member function to check existence of cycle in the minimum cost spanning tree.
    - Implement a function based on DFS.
    - For an efficient program, we had better to use the *union* and *find* operations (Chapter 5 in textbook). However, you don’t have to use it. If you want to use the *union* and *find* operations, please submit also a document about that topic.
  - Use a min heap to get a least cost edge
  - Print DFS path from vertex 0 and total cost for minimum cost spanning tree

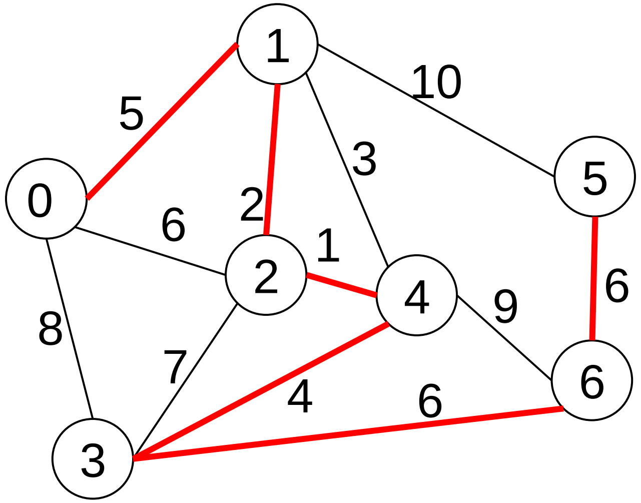
Requirements

- Expected results



```
PS C:\ds\hw05> .\KruskalMain.exe
Print All Lists: 7 vertices are in use currently
graph[0]: 3 -> 2 -> 1
graph[1]: 5 -> 4 -> 2 -> 0
graph[2]: 4 -> 3 -> 1 -> 0
graph[3]: 6 -> 4 -> 2 -> 0
graph[4]: 6 -> 3 -> 2 -> 1
graph[5]: 6 -> 1
graph[6]: 5 -> 4 -> 3

Kruskal Algorithm:
- DFS from 0: 0124365
- Total Cost: 24
```



Requirements

- All of C-style functions and headers are allowed.
  - E.g., printf, fopen, fgets, etc.

- Write clean source code
  - Add proper comment in your source code
  - Consider code indentation for enhancing readability

- Submit your screenshots.

Requirements

- For unmentioned requirements, you can implement freely.
- Test your source codes with many cases for self verification.
- Upload ZIP file on LMS by compressing all your source codes and screenshots
  - File name: hw05_student id.zip (ex: hw05_20400022.zip)
- Due date: 11pm, 5/27 (Thu)