Practice #11

Graphs (DFS & BFS)

Yunmin Go

School of CSEE

HANDONG GLOBAL UNIVERSITY

Practice #11 TO-DO List

To-Do	Submission	Notes
Graph Class	Screenshot and source code (All files including Graph.cpp)	p.33, 35, 39, Chapter 6

- Upload your screenshot and source codes on LMS by 11pm on 5/12 (Wed).
  - All your screenshots should be merged in one pdf file, screenshot.pdf.
  - Your pdf and all source codes should be compressed into zip file.
- File name: practice11_Your Student ID_Name.zip (only zip, not pdf, docx, c, etc)
  - ex) practice11_20400022_고윤민.zip

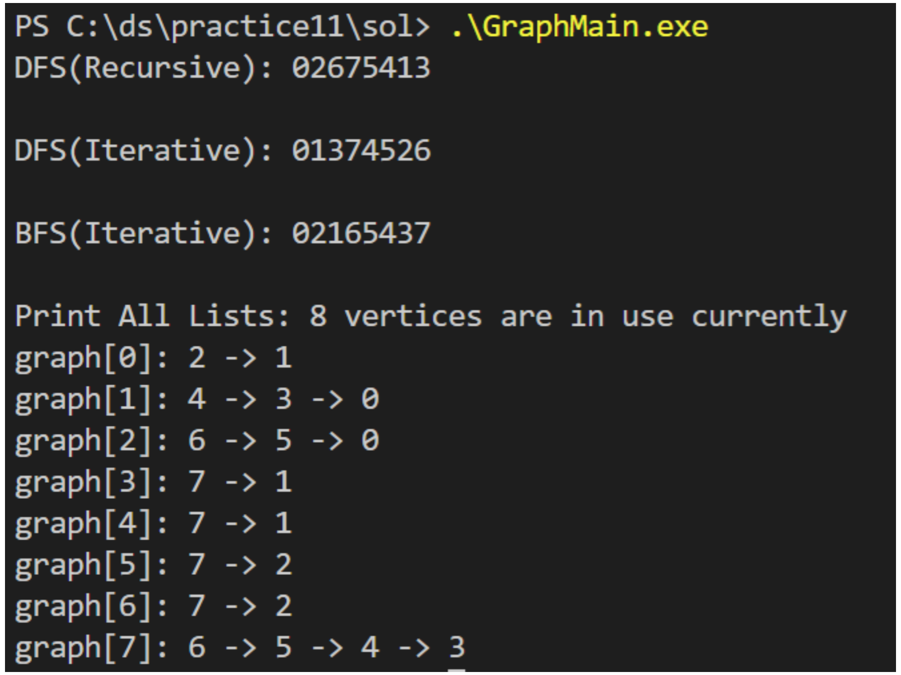
Graph DFS & BFS

- Implement a graph class for undirected graph
  - Complete a Graph.cpp (GraphMain.cpp: no need to change)
    - Graph.cpp defines a Graph class and its member functions
  - Refer to p.33, 35, 39, Chapter 6
  - We use adjacency list for graph representation.
  - Implement following member functions. You can modify the source codes and add additional member functions.
    - InsertEdge(int *src*, in *dest*): insert an edge between vertex *src* and vertex *dest*
      - Add new node at head of list (i.e., graph[])
    - DFS_recur(int *v*): iterative DFS algorithm (starts from vertex *v*)
    - DFS_iter(int *v*): iterative DFS algorithm (starts from vertex *v*)
    - BFS_iter(int *v*): iterative BFS algorithm (starts from vertex *v*)
    - PrintAdjList(): print all adjacency lists in graph[]

※ You can use given Stack.cpp and Queue.cpp for DFS and BFS.

Graph DFS & BFS

- Expected results

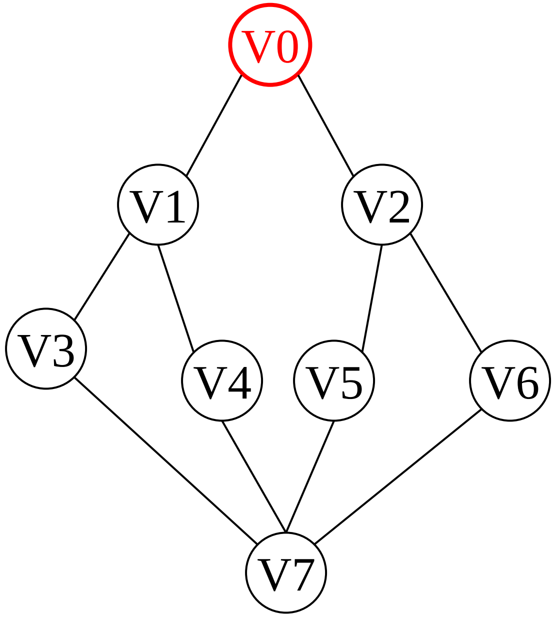


```
PS C:\ds\practice11\sol> .\GraphMain.exe
DFS(Recursive): 02675413

DFS(Iterative): 01374526

BFS(Iterative): 02165437

Print All Lists: 8 vertices are in use currently
graph[0]: 2 -> 1
graph[1]: 4 -> 3 -> 0
graph[2]: 6 -> 5 -> 0
graph[3]: 7 -> 1
graph[4]: 7 -> 1
graph[5]: 7 -> 2
graph[6]: 7 -> 2
graph[7]: 6 -> 5 -> 4 -> 3
```


V0
V1
V2
V3
V4
V5
V6
V7