Project 4. Graph Traversal

[Problem Description]

Lots of algorithms operating on graphs are based on graph traversal. Your task is to take the following graph as the test data and write a program to visit all vertices in an undirected connected graph.

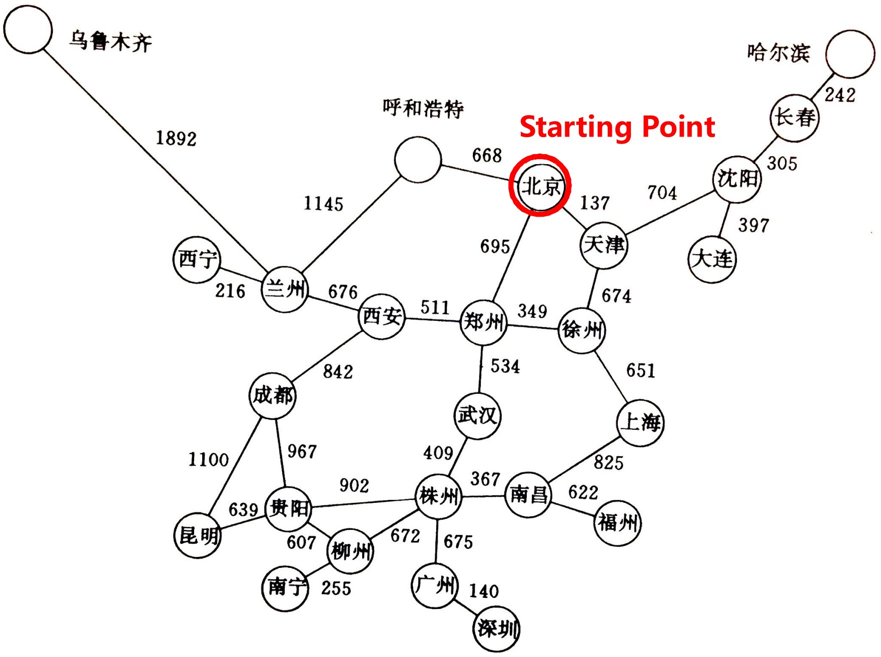
[Requirement]

(1) Use adjacency multi-list to implement the depth-first traversal and breadth-first traversal of the undirected connected graph. Taking a vertex as the starting point, output the visiting sequences and edge set of spanning trees respectively (preorder, in-order, post-order).

(2) Use stack to implement depth-first search traversal with non-recursive algorithm.

(3) Use adjacency list to build the depth-first spanning tree and the breadth-first spanning tree, and output the trees.

[Test Cases]



[Hints]

(1) Give a graph with numbered vertices (1, 2, 3, …, n, n≤30).

(2) The edges of the spanning tree are directed, the order of vertices cannot be reversed.

[Grading]

Implementation: 50% Interface: 30% Coding Style: 20%

Notice: This project will be checked on the experimental lesson in the 17th week (2016.12.20).