Self-Practice Week 8 - Graphs (part 2)

Minimum Spanning Tree & Shortest Path

The goal of this assignment is to develop a better understanding of the minimum spanning tree and shortest path algorithms and their use in practice.

Exercise 1 – Is an edge e in some MST?

Given an edge-weighted graph G and an edge `e`, design and implement a linear-time algorithm to determine whether `e` appears in some MST of G. Note that since your algorithm must take linear time in the worst case, you cannot afford to compute the MST itself.
Hint: consider the subgraph G' of G containing only those edges whose weight is strictly less than that of `e`.

Exercise 2 – Monotonic Shortest Path

Given an edge-weighted digraph G, design and implement a linearithmic algorithm to find a *monotonic* shortest path from a source node `s` to every other vertex. A path is *monotonic* if the sequence of edge weights along the path are either strictly increasing or strictly decreasing.
Hint: relax edges in ascending order to find a best monotonically increasing path; relax edges in descending order to find a best monotonically decreasing path.

Exercise 3 – Centrality metrics in the IMDB actor graph

Centrality metrics identify the most important vertices in a graph. Consider the social network of actors co-starring in IMDB movies (file `4-imdbcostars.txt`). Represent the co-starring relationship as a sparse graph, then efficiently implement the following API:

- `degreeCentrality(a)`: returns the number of edges incident to `a` (i.e., the number of actors `a` has co-starred with);
- `closenessCentrality(a)`: returns the reciprocal of the average length of the shortest path between `a` and all other vertices in the graph, normalised by the number of vertices in the graph (i.e., the more central actor `a` is in the IMDB graph, the closer it is to all other actors)

Write a tester program to find the actor in the IMDB graph with highest degree centrality, and the one with highest closeness centrality. What is the computational complexity of the latter?