Combinatorial Optimization

Paths and Trees

Dias, Schiewe, Pattanaki

§ Assignment I §

Context

In the fictional land of Sinooh, the governor is concerned with the reliability of its energy system. The primary source of energy production in this land is a gigantic solar farm incorporated directly into the design of its capital, Hearthome City. All the other towns do not have their production, only small storage that could supply power to only vital services for a limited amount of time. Therefore, there must always be a reliable power line between the capital and the towns and efficient routes allowing quick access.

As a recently graduated scientist, you have been hired by the governor to manage essential projects regarding their energy system infrastructure. This project is divided into two segments: in the first part, the goal is to have a reliable subset of power links connect all towns and the capital without any cycles and with the minimum possible cost, allowing all cities and towns to be connected in a grid. Hence, those power lines will not fail in an emergency, otherwise leaving some cities without a direct power supply.

In the second part, it is essential to have the quickest way to access every town from the capital in perfect condition. Hence, in an emergency, each town can be visited by help sent from the capital in the fastest way possible.

Description

Consider a set composed of the capital and the other towns. Always assume that the power line and the roads are **independent**, meaning that if a road exists between two towns or a town and the capital, it does not mean a power line between them and vice-versa. In addition, do not assume that the roads are both ways.

Your tasks are:

1. What is the minimum subset of power links that need to be reinforced, according to the description from the governor?
2. What is the shortest distance to access all towns starting from the capital?

Submission

Deadline: February 23rd, 2024
Implementation and Language: Whichever you prefer.
Preferably Python or Julia.
For Python check out Networkx (https://networkx.org/) and for Julia (https://juliapackages.com/p/networks).
Individual or in group of two

In the provided dataset, you find two folders: **Task I** and **Task II**. Each folder contains 10 files containing the nodes, edges and weights for a given graph.

Each file has the following format:

1. Number of Nodes;
2. A set of lines containing an edge and its weights;

Your submission will be graded from 0 to 100 in accordance with the following grading criteria: 75 points for the implementation and 25 for the report (see below).

For the implementation:

- Implement a function that reads a graph from a file and stores it in an adjacency matrix and in an adjacency list. Both representations should be obtained in your code. You can opt to write two separate functions. (15 points);
- Implement a solution method for Task 1 (see Section) either using an adjacency matrix or an adjacency list. Note that you can choose which data structure you prefer. (30 points);
- Implement a solution method for Task 2 (see Section) either using an adjacency matrix or adjacency list. Note that you can choose which data structure you prefer. (30 points);

Besides your code, a two-page report should also be submitted. In this report, the following sections are required:

1. Introduction: brief description of the problems;
2. Modelling choices: each task corresponds to a classical problem discussed in the lectures (see notes, slides and charts for assistance). Please explain why the task corresponds to the solution methods you decided to implement;
3. Discussion: the following question should be answered in your report:

> *Based on your expertise and point of view,* ***how*** *would you adapt the methods you implemented to incorporate extra constraints?* ***What*** *challenges would you face to do that?*

A template for this report can be found on "MyCourse".