Principles of Distributed Computing
Exercise 2

1 License to Match

In preparation of a highly dangerous mission, the participating agents of the gargantuan Liechtensteinian secret service (LSS) need to work in pairs of two for safety reasons. All members in the LSS are organized in a tree hierarchy. Communication is only possible via the official channel: an agent has a secure phone line to his direct superior and a secure phone line to each of his direct subordinates. Initially, each agent knows whether or not he is taking part in this mission. The goal is for each agent to find a partner.

a) Devise an algorithm that will match up a participating agent with another participating agent given the constrained communication scenario. A "match" consists of an agent knowing the identity of his partner and the path in the hierarchy connecting them. Assume that there is an even number of participating agents so that each one is guaranteed a partner. Furthermore, observe that[1] the phone links connecting two paired-up agents need to remain open at all times. Therefore, you cannot use the same link (i.e., an edge) twice when connecting agents with their partners.

b) What are the time and message (i.e., "phone call") complexities of your algorithm?

2 License to Distribute

We consider another day at the office of the LSS as in Exercise 1. After the above mission was successful, the involved agents collected a large number of sensitive documents. Some agents might have a lot of them and others have none. Now they need to distribute the documents throughout the agency such that each person in the LSS has the same amount of data to process.

a) Assume that there are n agents in the LSS and that there is also a total of n documents. Devise a way for the agents to distribute their sensitive data: in the end, each agent should have exactly one document. The communication scheme is the same as above.

b) How good is your algorithm with respect to time and number of messages? You may assume that arbitrarily many documents can be sent in a single message.

[1] in the case of an emergency where they lose contact

3 License to Connect

Catastrophe! LSS Headquarters in Vaduz have been compromised and all LSS antennas around the world have lost their primary communications channels. Luckily, just before going rogue, the HQ has managed to send a last message:

```
To all LSS antennas;
LSS HQ is compromised;
Shut down all modern communication channels;
Use GHS on old cable network to re-establish communication between all antennas.
```

Indeed, this old cable network has not been used in ages, and all information on its topology has long been forgotten. Antennas are only aware of the cost needed to upgrade low-throughput incoming cables to the modern LSS standards. Upon reception of this message, each antenna initiates the GHS (Gallager–Humblet–Spira) algorithm through the cable network in order to rebuild a modern communication spanning tree at a minimum cost.

a) For each phase of the GHS algorithm, list the fragments and their ID (i.e. the root ID). If there are two candidates, the fragment's root is chosen in alphabetical order.

b) Each time an edge is added to a fragment, specify if it's the result of a merge (the merge request was bilateral) or an absorption (the merge request was unilateral).

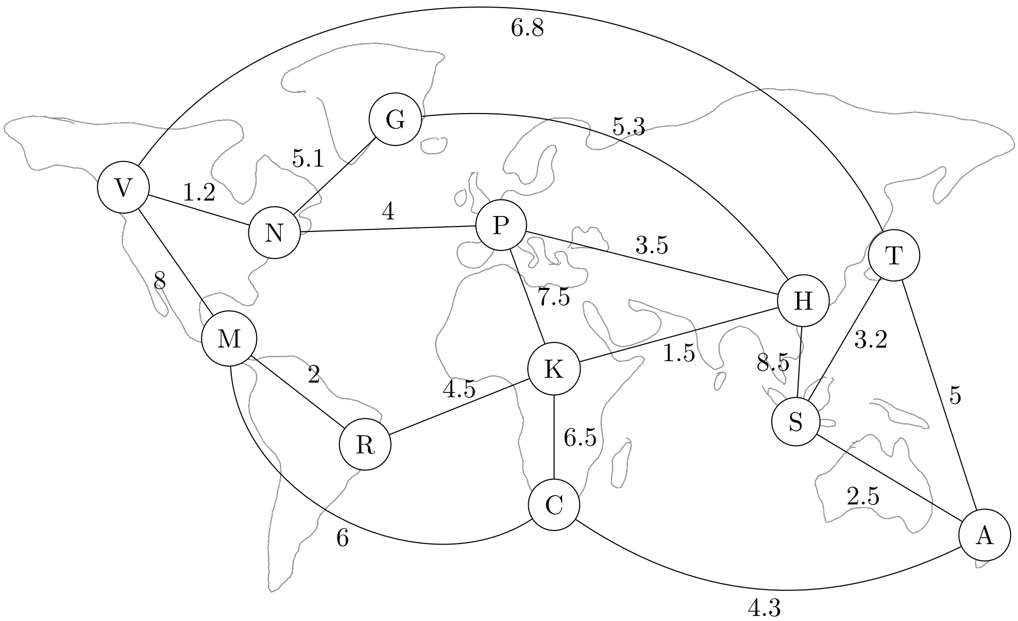

Figure 1: Old cable network (edges) between LSS antennas (nodes). Edge weights represent the upgrading cost of the cable.