AD – Assignment 5

Shortest Paths and MST

Søren Dahlgaard

Hand-in date: March 21st. NO RESUBMISSION!

Should be done in groups of 2–3 students

1 Motivation

The goal of this assignment is to familiarize you with the properties of distances in graphs used for shortest paths and MST algorithms. It is also an illustration of how one can use the MST of a graph as a basic starting point for solving a problem. Finally, the assignment should also familiarize you with usage of disjoint sets.

2 Problem statement

The CPS Brewing Company controls a complicated network of breweries connected by roads. Having learnt quite a lot of algorithms, the company now only uses shortest paths to transfer ingredients between breweries. However, the company has only kept track of these shortest distances, and not the actual roads between breweries.

The company is planning to expand, and would therefore like you to calculate how the network actually looks. The CEO of CPSBC has told you that the network satisfies the following properties:

1. There are exactly n breweries and $n-1$ roads.
2. All breweries are connected with each other – i.e. the network consists of just *one* connected component. The network is thus a tree.
3. All roads have positive length (strictly greater than 0). All roads can carry traffic in both directions.
4. You are given the shortest path distance between each pair of breweries.

Given this information the CEO has asked you to figure out how the network could possibly look.

More formally: Let the actual brewery network be a tree T. Given just the shortest path distances of T, you have to reconstruct the original network T.

Input: An $n \times n$ distance matrix H with $H_{ij} = \delta_T(i,j)$, where T is the actual network of breweries and $\delta_T(i,j)$ is the shortest path distance between breweries i and j in T.

Output: The $n-1$ edges of T.

3 Example

We will use a few definitions to help with the notation:

- T is the actual brewery network.
- H is the $n \times n$ shortest path distance matrix.
- $G(H)$ is the complete graph on n nodes, where edge (i, j) has weight H_{ij} – i.e. the shortest path distance in T.

As an example you may look at Figure 1. It shows the transformation from T to H and H to $G(H)$ and $G(H)$ back to T.

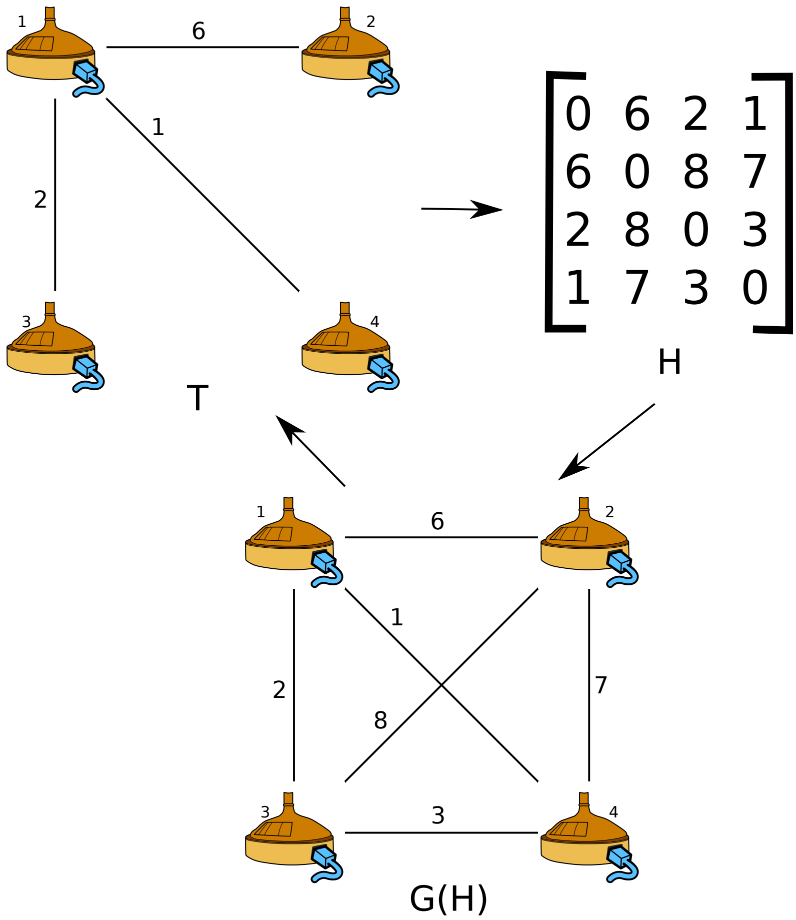

Figure 1: Example problem instance.

4 Assignment

Task 1: Prove that an entry H_{ij} of minimal value must be an edge of T – i.e. if H_{ij} has minimum value among all entries of H, then (i, j) is an edge of

T and has $w(i,j) = H_{ij}$; the minimum is only taken over non-diagonal entries of H. Do not mention minimum spanning trees in your proof.

Hint: Try to prove this by contradiction. Recall that all edges of G have positive weight (strictly greater than 0).

Task 2: Consider the complete graph $G(H)$ as described in the example above. Prove that a minimum spanning tree T' of $G(H)$ is equal to T. You must prove the statement for the general case and not just the example.

Hint: Consider an edge e added in a step of Prim's or Kruskal's algorithm. Is it possible that e was not a part of T? You should be able to argue the same way as in task 1.

Task 3: What is the running time of the algorithm resulting from running a MST algorithm on the input and returning the list of edges as a function of n? (Note that $|E(G(H))| = \Theta(n^2)$).