Exercise session 5: Weights, distances, flows · 1MA020

Vilhelm Agdur[1]

[1] vilhelm.agdur@math.uu.se

6 November 2023

In the previous lecture, we learned how to count the number of spanning trees of a graph. Now, we study how to find spanning trees, and minimal spanning trees. We also think about the vertex version of Eulerian circuits, which are called Hamilton paths. Finally, we consider flows in graphs.

Weighted graphs and minimum spanning trees

We saw in our final theorem of the previous lecture that there is an effective way of computing how many spanning trees there are. Is there also a good way of finding one? The answer to this is also yes – and in fact we can do something much stronger in an efficient way.

To explain the problem we are concerned with, let us introduce the notion of a *weighted* graph. We will consider simple weighted graphs, but the same notion makes perfect sense also for multigraphs and directed graphs.

Definition 1. A *weighted graph* is a simple graph $G = (V, E)$ together with a *weight function* $w : E \to \mathbb{R}$. If $H = (V', E')$ is a subgraph of G, its *weight* is defined as $w(H) = \sum_{e \in E'} w(e)$, whenever this is well defined.[2] A *minimum spanning tree* (MST) is a spanning tree T of G such that $w(T)$ is minimal among all spanning trees.

[2] If it contains infinitely many positive and infinitely many negative weights, the sum might not converge, but this case is entirely pathological and won't occur for us.

It is clear that for finite weighted graphs, there always exists at least one minimum spanning tree, but they are not necessarily unique – if all edges have the same weight, then of course every spanning tree is minimal. However, if all edges are given different weights, then the minimum spanning tree is indeed unique.

Exercise 1. The things we do here don't work at all for infinite graphs. To demonstrate this, find an infinite weighted graph without a minimum spanning tree. Can you find one with only positive weights?

Exercise 2. Prove that if all edges have different weights, then the minimum spanning tree is unique.

The problem of actually finding these minimum spanning trees algorithmically is one we will be considering during our next exercise session, and then dealing with in the lecture after that.