

# NETWORKS AND COMPLEXITY

## Solution 1-11

*This is an example solution from the forthcoming book *Networks and Complexity*.*

*Find more exercises at <https://github.com/NC-Book/NCB>*

### **Ex 1.11: Pruning networks [4]**

Develop your own algorithm for finding minimal spanning trees. In contrast to Kruskal's algorithm start with a fully-connected network and then remove links until only the minimum spanning tree is left. Make sure that your algorithm always yields the optimal result.

#### Solution

To find the desired algorithm we need to think about the problem differently. In Kruskal's we structured our thinking around the number of components, we needed to bring it down. But if start from the fully connected network the number of components is already one, so what can we consider instead?

Actually our goal is to remove links. The fully connected network will have a number of links, let's call this number  $M$ . We say that a network is optimal with  $M$  links if it a) contains exactly  $M$  links, b) only a single component, and c) has the minimum combined link length of all networks that meet conditions (a) and (b).

In analogy to our previous reasoning we can say that the fully connected network is the optimal network with  $M$  links.

If we remove the longest link from the fully-connected network we arrive at the network that must be the optimal network with  $M - 1$  links, because every network with  $M - 1$  links requires one removal and we have removed the longest link we could.

This reasoning continues to work until eventually the next shorter link cannot be removed because its removal would break the network into two components. Let's call this link  $X$ . We argue that  $X$  must be present in every optimal network with a given number of links. The argument goes as follows: If we removed  $X$  then we would need to add at least one of the links we have previously removed to reconnect the network into a single component. But all those links we have removed before  $X$  are longer than  $X$ , so it is always better to keep  $X$ .

We can thus consider the next shorter link either it can be removed or it is also one of the links that must be present in every optimal network. If we continue this process we end up removing all the links except for those that must be present in every optimal network. These links then form the minimum spanning tree.

In summary: Start with the fully connected network. Then consider the links one by one in the order of reverse cost. If removing a link increases the number of components then retain it, otherwise remove it. When all links have been considered you have arrived at the minimum spanning tree.

For Moravia this means:

1. Consider (O,J) [198km] – remove
2. Consider (Z,J) [151km] – remove
3. Consider (B,O) [137km] – remove
4. Consider (L,J) [121km] – remove
5. Consider (B,J) [77km] – retain
6. Consider (O,Z) [76km] – remove
7. Consider (O,L) [75km] – retain
8. Consider (B,Z) [74km] – remove

The final step leaves us with a tree, so we can't remove any more links. As before we find that the optimal power grid has the edge set  $E = \{(O, L), (L, Z), (B, L), (B, J)\}$