0.1 Optimisation Example: Kruskal's Algorithm

Example 0.1. Suppose we have a country containing an n number of cities that are currently isolated from each other. As the new minister for transport it is your idea to correct this tansport issue and to lay a railroad which should connect each city to any other city by a unique path. However, you have a budget. Each railway line will cost a certain predefined amount to lay (with no difficulties or unforeseen costs). How will you decide which city-links are the optimal ones to lay railtracks on?
We will soon see that this can be done by finding a minimal spanning tree. Which can be found through a greedy algorithm process. A suitable example of this kind of algorithm which should solve our problem is Kruskal's Algorithm. Which is detailed below.

Algorithm 1 Kruskal's algorithm

Let G be a connected graph with vertex set $V = \{1, ..., n\}$ and $\omega : E \longrightarrow \mathbb{R}^+$ a weight function. The edges of G are ordered according to their weight, that is, $E = \{e_1, ..., e_m\}$ and $\omega(e_1) \leq ... \leq \omega(e_m)$.

procedure KRUSKAL(G, ω, T)
    $T \leftarrow \emptyset$
    for $k = 1$ to m **do**
        if ACYCLIC($T \cup \{e_k\}$) **then**
            append e_k to T

---

1) Create a graph F containing just the vertices of G.
2) Create a set $S = E(G)$; the edge set of G.
3) While S is non-empty and F is not yet spanning
3(a) Remove an edge with minimum weight from S.
3(b) If the removed edge introduces no cycles to F
then add the edge to F
