Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Departement of Computer Science
Markus Püschel, David Steurer

10. December 2018

Algorithms & Data Structures — Homework 12 — HS 18

Exercise Class (Room & TA): ______________________

Submitted by: ______________________

Peer Feedback by: ______________________

Points: ____

Exercise 12.1 *Ancient Kingdom of Macedon **(1 Point)**.*

The ancient Kingdom of Macedon had N cities and M roads connecting them, such that from one city, you can reach all other $N-1$ cities. All roads were *roman roads* i.e. stone-paved roads that did not require any maintenance and no two roads were of the same length. With the technological developments in the Roman Kingdom, a new type of carriage was developed, called the *Tesla Carriage* that was much faster than all the alternatives in the Ancient Macedon Kingdom. However, the Tesla Carriage required *asphalt roads* to operate, such that the roads had to be maintained every year, or otherwise the asphalt would wear off, rendering the road unusable as if it was a roman road.

With the effort to modernise the kingdom, Phillip II promised the Ancient Macedonians that he will provide them with asphalt roads by paving some of the existing roman roads, such that every two cities can be reached through a Tesla Carriage. The price to pave a roman road or maintain an asphalt road is equal, and is proportional to the length of the road. To save money Phillip II decided to pave sufficient roman roads to fulfil his promise, while minimizing the overall yearly maintenance price.

Even in the first years, the new Tesla Carriages improved the lives of the average Ancient Macedonians, but at the same time, they also provided means for robbers to commit crimes and escape to another city. To resolve this, Phillip II decided to create checkpoints the second year, one at each asphalt road. Each of the checkpoint will have a fixed cost for both building and maintenance.

Assuming a fixed price k for each checkpoint, does Phillip II have to consider paving new roman roads, or he can maintain the same set of roads in order to make sure that the overall maintenance price of the roads and the checkpoints is still minimal? Prove your reasoning, or provide a counter example.

Note: For simplicity, assume that the roman roads were paved all at once, on the first day of the year, and maintenance will be done the same day next year, again all at once. Also assume that checkpoints can also be built at once for all roads, as well a they can be maintained all at once in a day.

Solution.

Let's think of all cities in the Ancient Macedon Kingdom as vertices in a graph G and all roads as edges. In order to minimize the overall maintenance price, Philip II must pave only roman roads that from a minimum spanning tree T in G. As a result, we can rephrase the problem into a graph problem: Let T be a minimum spanning tree of a weighted graph $G(N, M)$. Construct a new graph G' by adding a weight of k to every edge of G. Do the edges of T form a minimum spanning tree of G?

In a graph with N vertices, every spanning tree has $N-1$ edges. Thus the weight of every spanning tree is increased by exactly $(N-1) \cdot k$. The set of minimal weight spanning trees remains the same.

Exercise 12.2 *The Swiss Federal Roads Office* **(2 Points)**.

The Swiss Federal Roads Office (FEDRO) decided to modernize the existing roads in Switzerland, creating electric roads. Each electric road will have an electric rail embeded in the road, such that each electric car can be charged by induction as it moves on the road. FEDRO promised its citizens, that their electric cars will no longer need batteries, and that induction system will be used to traverse between any two cities in the country.

The price for modernizing any road would be proportional to the length of the road. As this was a big investment, the proposal was put on referendum and each canton made a choice. Some cantons would pay the federal office for modernizing some roads within the canton and the surrounding, but other cantons decided that the money for modernization should be provided by the FEDRO office.

FEDRO decided to go on with construction anyway, knowing that for some roads they will have to pay money to modernize, while for others they will get money for building them. Your task is to devise an (as efficient as possible) algorithm to decide which roads need to get modernized such that the price that FEDRO has to pay is minimized, and output the overall price.

For simplicity - assume you are given N cities, and M roads. For each road you are given the non-negative price of the road and whether FEDRO office should pay for that road or not.

Answer the following questions:

- How do you model the graph? What are the vertices, edges, edge weights? How many vertices and edges does the graph contain? (Everything if possible in words.)
- How does your algorithm work?
- What is the running time of your algorithm (as concisely as possible in Θ notation)?

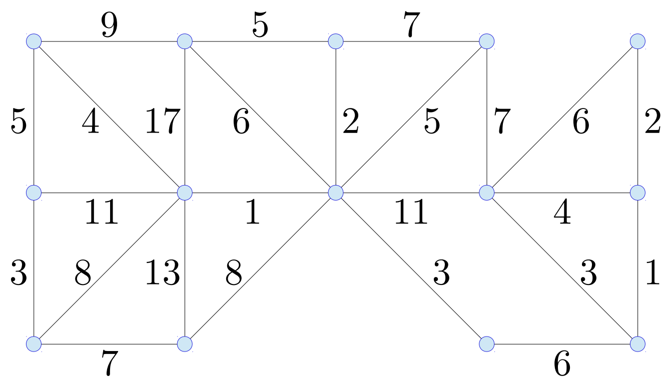
Solution.

- The graph contains N vertices (one for each city) and M edges (there is an edge between two vertices, whenever a street connects the two cities). The edge weights are as follows: For every edge (street) that needs to be paid by the FEDRO, the edge weight is equal to the cost to build the street. For every edge, that gets paid by a canton, the edge weight is 0.
- Run Kruskal's or Prim's algorithm on the graph.
- Kruskal: Sorting all edges takes $\Theta(M \log M)$ time. Running Kruskal takes $\Theta(M \log N)$ time.

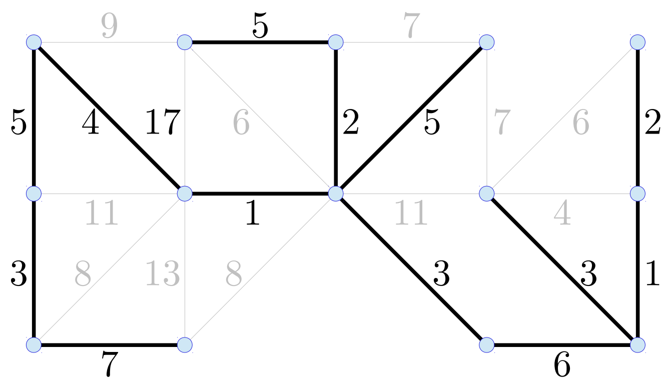
  Prim: The time complexity is $\Theta(M + N \log N)$ when implemented efficiently.

Exercise 12.3 *Compute minimum spanning tree I.*

Run the algorithm of Kruskal on the following graph to compute a minimum spanning tree. Mark the edges that are contained in the solution.

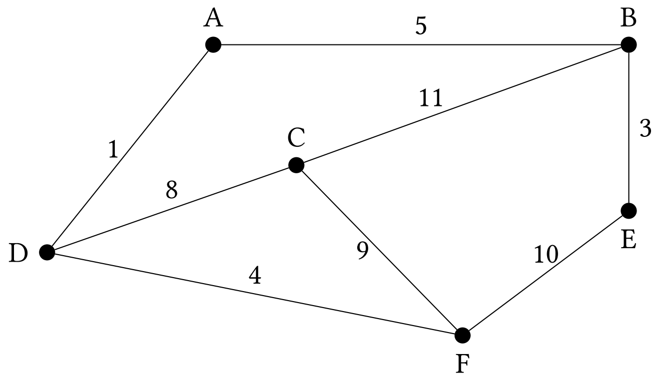


Solution. Kruskal finds the following spanning tree:



Exercise 12.4 *Compute minimum spanning tree II.*

Given the following graph, answer the questions below.



a) Which are the first three edges that Prim's algorithm (also discovered by Jarnik and Dijkstra) includes in the minimum spanning tree, when *starting in vertex A*?

b) Which is the first edge, that gets rejected by the algorithm of Kruskal?

Solution.

a) (A, D), (A, B) und (D, F)

b) (C, F)

Exercise 12.5 *Properties of minimum spanning trees.*

1. Let $G = (V, E)$ be a graph with n nodes and edge weights $w\colon E \to \mathbb{N}$. Let $e_1, \ldots, e_m$ be an enumeration of the edges of G. Consider modified edge weights such that $w'(e_i) = n{\cdot}m{\cdot}w(e_i){+}i$. Show that a minimum spanning tree with respect to w' is also a minimum spanning tree with respect to w.

   Note that by modifying the weights in this way, all weights in w' become unique. This implies that for computing minimum spanning trees (i.e., for running a minimum spanning tree algorithm), we can in general assume that all edge weights are unique. However, for some structural properties, it matters if the weights are unique or not, as we will see in some of the following exercises.

2. Given two different minimum spanning trees T_1 and T_2 of a graph G. Show that the number of edges with weight w is always the same for both trees: If T_1 contains k edges of weight w, then also T_2 does, and vice versa.

3. Show that if all edge weights in a graph are unique, then also the minimum spanning tree is unique.

4. Prove or disprove: If the minimum spanning tree of a graph G is unique, then G does not contain two edges with the same weight.

5. A *Shortest-Path-Tree* (*Kürzeste-Wege-Baum*) in a graph G with edge weights is a spanning tree B with root r, such that for every vertex v in the tree it holds that the unique path from r to v in B is a shortest r-v-path in G. Prove or disprove:

(a) every minimum spanning tree is a shortest-path tree (for some root r).

(b) Every shortest-path-tree is a minimum spanning tree.

Solution.

1. We will show that any two spanning trees T and T' with $w(T) < w(T')$ satisfy $w'(T) < w'(T')$. Indeed, $w'(T') - w'(T) = n \cdot m \cdot (w(T') - w(T)) + \sum_{e_i \in T'} i - \sum_{e_j \in T} j \geq n \cdot m - \sum_{e_j \in T} j \geq n \cdot m - (n-1) \cdot m > 0$.

2. Before we show the actual claim, let us first show the following subclaim:

   For two different MSTs S and T of a connected graph G there are always two edges $s \in E(S) \backslash E(T)$ and $t \in E(T) \backslash E(S)$, so that $S + t - s$ is another MST of G.

   We mean by $S + t - s$ the tree that is obtained when we remove s from S and add t instead.

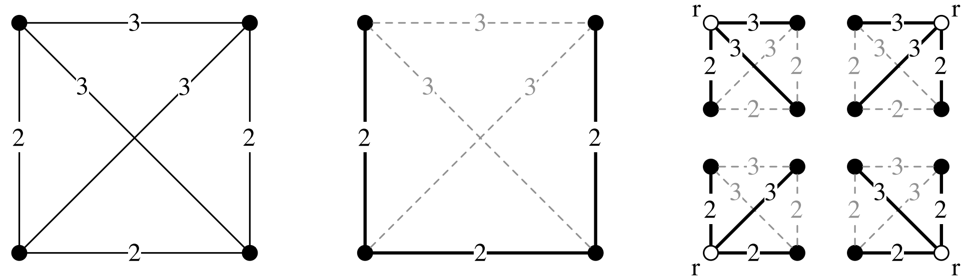
   Proof (Subclaim)**:** Let t be an arbitrary edge in T that is not in S (i.e., $t \in E(T) \backslash E(S)$). When adding t to S, we get a cycle. But if we remove any other edge e from $E(S) \backslash E(T)$ of this cycle, then we get another spanning tree. Note that such an edge must exist since T does not contain a cycle. Among these edges in the cycle there must be an edge s in $E(S) \backslash E(T)$ that connects the two connected components of $T \backslash t$. Hence, with this choice of s and t we could also remove t from T and add s to obtain a new spanning tree. This implies that it is not possible that $w_s < w_t$, as otherwise $T - t + s$ was a spanning tree with smaller wight than T, and similarly, it is not possible that $w_s > w_t$, as otherwise $S + t - s$ had smaller weight than S. Therefore, it must hold that $w_s = w_t$ and thus $S + t - s$ and $T - t + s$ both are minimum spanning trees of G. □

   We now start with the actual proof. Assume towards a contradiction that S and T are two MSTs of a graph $G = (V, E)$ such that not all edge weights appear the same number of times. By the above subclaim, it must hold that there are two edges $s \in E(S) \backslash E(T)$ und $t \in E(T) \backslash E(S)$, so that $S' := S + t - s$ is an MST of G. Observe that in S' all the weights appear the same number of times as in S since $w_s = w_t$. Now, if $S' \neq T$, then repeat the subclaim-argument for S' and T to obtain an MST S''. We can repeat this step until we obtain a new MST S^* that is equal to T and contains the same weights as S. This clearly contradicts the initial claim. Note that the described procedure converges after at most $|V| - 1$ repetition, since in each step, we add one edge of T. □

3. Let $G = (V, E, w)$ be an undirected graph such that no two edges $e, e' \in E$ have the same weight. Assume towards a contradiction that G has two MSTs $T_1 = (V, E_1)$ and $T_2 = (V, E_2)$. The edges in $E_1 \cap E_2$ appear in both T_1 and T_2, while the edges in $E_1' = E_1 \backslash E_2$ and $E_2' = E_2 \backslash E_1$ are only contained in T_1 resp. T_2. Since the weights of the edges are pairwise distinct, there must be an edge $e^{\perp}$ in $E_1' \cup E_2'$ with minimum weight. Assume w.l.o.g. that $e^{\perp}$ is in T_1. If $e^{\perp}$ is added to T_2, then T_2 contains a cycle with t least one edge $e' \in E_2'$ (otherwise T_1 would have contained a cycle). When we remove e' from T_2 and add $e^{\perp}$ instead, we get a spanning tree. Since $w(e^{\perp}) < w(e')$, the new spanning tree has smaller weight than T_2, which is a contradiction to the assumption that T_2 is a minimum spanning tree. □

4. The claim is wrong: Consider the complete undirected graph with 3 vertices such that 2 edges have weight 1 and the third edge has weight 2. There is a unique minimum spanning tree of this graph although it contains two edges with equal weight.

5. Both claims are wrong as illustrated in the following counterexample. In particular, the graph G has a unique minimum spanning tree that for none of the possible roots is a shortest-path tree.



G MST Kürzeste-Wege-Bäume

Submission: On Monday, 17.12.2018, hand in your solution to your TA *before* the exercise class starts.