Eidgenössische
Technische Hochschule
Zürich

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

Departement of Computer Science
Johannes Lengler, David Steurer
Lucas Slot, Manuel Wiedmer, Hongjie Chen, Ding Jingqiu

11 December 2023

Algorithms & Data Structures — Exercise sheet 12 — HS 23

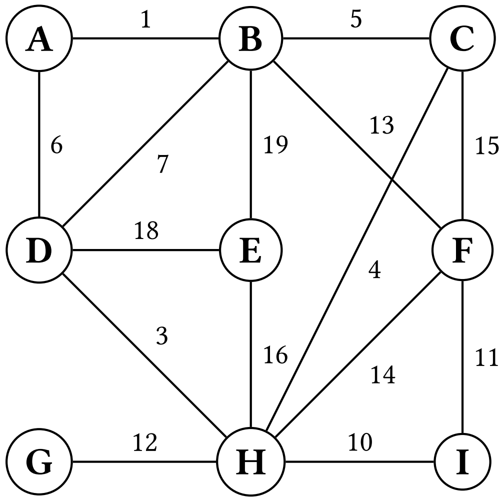
The solutions for this sheet are submitted at the beginning of the exercise class on 18 December 2023.

Exercises that are marked by * are challenge exercises. They do not count towards bonus points.

You can use results from previous parts without solving those parts.

Exercise 12.1 *MST practice* **(1 point)**.

Consider the following graph.



(a) Compute the minimum spanning tree (MST) using Boruvka's algorithm. For each step, provide the set of edges that are added to the MST.

(b) Provide the order in which Kruskal's algorithm adds the edges to the MST.

(c) Provide the order in which Prim's algorithm (starting at vertex G) adds the edges to the MST.

Exercise 12.2 *Uniqueness of MSTs* **(1 point)**.

The goal of this exercise is to understand when a graph has a unique minimum spanning tree.

(a) Give an example of a graph for which the minimum spanning tree is not unique. Show how to get two different minimum spanning trees of this graph using Kruskal's or Prim's algorithm. When there is a choice because several edges have the same weight, the algorithms are allowed to pick any of those edges.

It turns out that for a connected graph, if the weights of the edges are pairwise distinct, the minimum spanning tree is unique. To show this, let $G = (V, E)$ be a connected graph and $w : E \to \mathbb{R}_{\geq 0}$ be a weight function such that $w(e) \neq w(f)$ for $e, f \in E$ with $e \neq f$. We assume by contradiction that there are two different minimum spanning trees T_1 and T_2. Out of all edges that are in $T_1 \backslash T_2$ or $T_2 \backslash T_1$, let e be the edge of minimum weight (the edge of minimum weight is unique since by assumption the edge weights are pairwise distinct). By exchanging the roles of T_1 and T_2 if necessary, we can assume that $e \in T_1 \backslash T_2$.

(b) Show that $T_2 \cup \{e\}$ has a cycle and that there is an edge $f \in T_2 \backslash T_1$ on this cycle that has strictly larger weight than e.

(c) Show that the minimum spanning tree of G with the weight function w is unique.

Hint: *Use part (b) to construct a spanning tree of smaller weight than* T_2.

(d) Is the converse true as well? That is, if $G = (V, E)$ is a connected graph that has a unique minimum spanning tree, are the edge weights necessarily distinct? Give a proof or counterexample.

Exercise 12.3 *Constructing a Fiber Optic Network.*

The government of Atlantis put you in charge of installing a fiber optic network that connects all its n cities. There are two technologies of fibre optic that you can use:

- Fibre 1.0: It is a good reliable technology that is relatively cheap. There is a list of pairs of cities between which it is possible to install a direct Fibre 1.0 link. Furthermore, for each such pair, there is a corresponding positive integer cost.
- Fibre 2.0: It is an emerging technology that it extremely good and can directly connect any two cities. However, its cost is too high and the government cannot afford a single Fibre 2.0 link.

Note that all direct links are two-directional. The installed network should connect all the cities of Atlantis: Between any two cities, there should be a connected path of direct links in the network that connects them.

A philanthropist volunteered to donate the cost of exactly $k < n$ direct Fibre 2.0 links, and you can use them to connect any k pairs of cities. Your goal is to minimize the cost that is paid by the government for the Fibre 1.0 links that are needed to construct a connected network. Describe an algorithm that finds the network that costs the government the minimum amount of money.

Note that it is possible to construct a network connecting all the cities of Atlantis using only Fibre 1.0 links, but we would like to benefit from the k Fibre 2.0 links that were donated by the philantropist in order to minimize the cost that is paid by the government.

Hint: *Modify Kruskal's algorithm.*

Exercise 12.4 *TST and MST* **(1 point)**.

Let $G = (V, E)$ be a connected, weighted graph, with weights $w : E \to \mathbb{R}_{\geq 0}$. A *travelling salesperson tour* (TST) in G is a closed walk which visits each vertex $v \in V$ at least once. We write $\operatorname{tst}(G)$ for the length of a shortest TST in G, that is:

$$\operatorname{tst}(G) = \min_{\substack{P=(v_1,\ldots,v_\ell) \\ \text{is a TST in } G}} w(P), \qquad \text{where } w(P) := \sum_{i=1}^{\ell-1} w\big(\{v_i, v_{i+1}\}\big).$$

(a) Let $M \subseteq E$ be the edges of a minimum spanning tree of G, with weight $w(M) := \sum_{e \in M} w(e)$. Prove that $w(M) \leq \operatorname{tst}(G)$.

(b) Let $H = (V, M_{\text{double}})$ be the multigraph with vertex set V, and edge set M_{double} containing two copies of each edge $e \in M$. Prove that H has a Eulerian tour of length $2 \cdot w(M)$.

Hint: See Exercise 10.1. What can you say about the degree of a vertex in H?

(c) Describe an algorithm which outputs a TST in G of length at most $2 \cdot \mathrm{mst}(G)$, where $\mathrm{mst}(G)$ is the length of a minimum spanning tree of G. The runtime of your algorithm should be at most $O(|E| \log |E|)$. Prove that your algorithm is correct and achieves the desired runtime.

***Hint:** For a connected graph with n vertices and m edges, you may use the fact that there exists an algorithm to find a minimum spanning tree in time $O(m \log m)$, and a Eulerian tour (if one exists) in time $O(m)$.*

Exercise 12.5 *Maximum Spanning Trees and Trucking.*

We start with a few questions about **maximum spanning trees**.

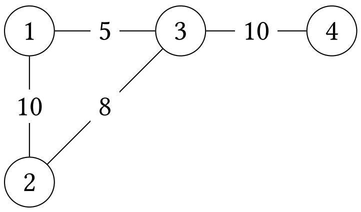
(a) How would you find the **maximum** spanning tree in a weighted graph G? Briefly explain an algorithm with runtime $O((|V| + |E|) \log |V|)$.

(b) Given a weighted graph $G = (V, E)$ with weights $w : E \to \mathbb{R}$, let $G_{\geq x} = (V, \{e \in E \mid w(e) \geq x\})$ be the subgraph where we only preserve edges of weight x or more. Prove that for every $s \in V$, $t \in V$, $x \in \mathbb{R}$, if s and t are connected in $G_{\geq x}$ then they will also be connected in $T_{\geq x}$, where T is the maximum spanning tree of G.

***Hint:** Use Kruskal's algorithm as inspiration for the proof.*
***Hint:** If it helps, you can assume all edges have distinct weight and only prove the claim for that case.*

Problem: You are starting a truck company in a graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$. Your headquarters are in vertex 1 and your goal is to deliver the maximum amount of cargo to a destination $t \in V$ in a single trip. Due to local laws, each road $e \in E$ has a maximum amount of cargo your truck can be loaded with while traversing e. Find the maximum amount of cargo you can deliver for each $t \in V$ with an algorithm that runs in $O((|V| + |E|) \log |V|)$ time. For the purpose of this exercise you can assume that your truck has unlimited capacity.

Example:



Output:
```
Max cargo to 1 is ∞
Max cargo to 2 is 10
Max cargo to 3 is 8
Max cargo to 4 is 8
```

Explanation:
The best path from the headquarters to 4 is $1 \to 2 \to 3 \to 4$, and the maximum cargo the truck can carry is $\min(10, 8, 10) = 8$.

(c) Prove that for every $t \in V$, the optimal route is to take the unique path in the **maximum** spanning tree of G.

***Hint:** Suppose that the largest amount of cargo we can carry from 1 to t in G (i.e., the correct result) is OPT and let ALG be the largest amount of cargo from 1 to t in the maximum spanning tree. We need to prove two directions: $OPT \leq ALG$ and $OPT \geq ALG$.*
***Hint:** One direction holds trivially as any spanning tree is a subgraph. For the other direction, use part (b).*

(d) Write the pseudocode of an algorithm that computes the output for all $t \in V$. The runtime of your algorithm should be $O((|V| + |E|) \log |V|)$. You can assume that you have access to a function that computes the maximum spanning tree from G and outputs it in any standard format. Briefly explain why the runtime bound holds.