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
Kasper Lindberg, Lucas Slot, Hongjie Chen, Manuel Wiedmer

9 December 2024

Algorithms & Data Structures — Exercise sheet 12 — HS 24

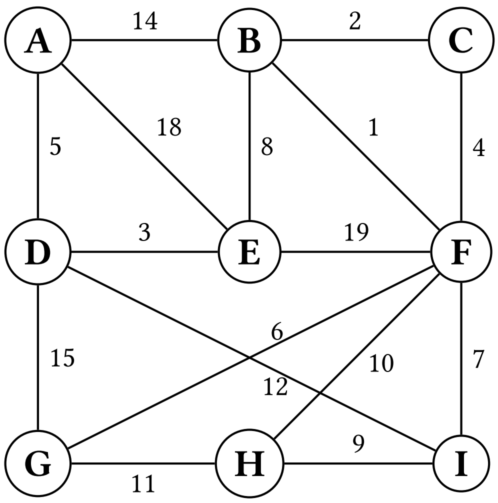
The solutions for this sheet are submitted on Moodle until 15 December 2024, 23:59.

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 *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 is 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 direct Fibre 2.0 links ($k < n$), 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 12.3 *Exploring connectivity of MSTs* **(1 point)**.

In this exercise, we explore connectivity properties of the set of spanning trees and MSTs of a graph using only 'local' changes. First we prove whats called the symmetric basis exchange property. Let $G = (V, E)$ be a connected graph and $w : E \to \mathbb{R}_{\geq 0}$ be a weight function.

(a) Let T_1 and T_2 be two different spanning trees of G and let $e \in T_1 \setminus T_2$. Show that there exists an edge $f \in T_2 \setminus T_1$ such that $(T_1 \setminus \{e\}) \cup \{f\}$ and $(T_2 \setminus \{f\}) \cup \{e\}$ are both spanning trees.

Now consider the graph $H = (\mathcal{B}, \mathcal{E})$ where each vertex of H corresponds to a spanning tree of G and we assign an edge between two vertices of H if their corresponding spanning trees differ by exactly two edges.

(b) Show that the graph H is connected.

    Hint: *Repeatedly apply part (a).*

(c) Consider the subgraph of H, H_{MST}, whose vertices are all MSTs of G and we keep an edge between two vertices if, again, the corresponding MSTs differ by two edges. Show that H_{MST} is connected.

    Hint: *Reuse the proof for (b) but also analyze the weights of the new spanning trees produced by (a).*

Exercise 12.4 *Maximum Spanning Trees and Trucking.*

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

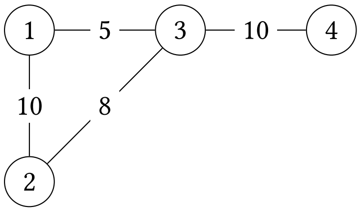
(a) How would you find the **maximum** spanning tree in a weighted graph $G = (V, E)$? Describe an algorithm with runtime $O((|V| + |E|) \log |V|)$.

(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: $OPT \le ALG$ and $OPT \ge ALG$.*
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.

Exercise 12.5 *Heavy and light edges* **(1 point)**.

Let $G = (V, E)$ be a connected, undirected, weighted graph with positive weights $w_e > 0$ for $e \in E$. We say an edge $e \in E$ is *heavy* if there exists a cycle $C \subseteq E$ so that $e \in C$ is the (strictly) heaviest edge in C, i.e.,

$$w_e > w_f \text{ for all } f \in C \text{ with } f \neq e.$$

We say an edge is *light* if there exists a minimum spanning tree $T \subseteq E$ of G which contains e.

(a) Show that a heavy edge cannot be light.

Hint: Assume for a contradiction that $T \subseteq E$ is an MST of G and that T contains a heavy edge e. Say e is the heaviest edge in a cycle $C \subseteq E$. Construct a strictly cheaper spanning tree of G by removing e from T, and replacing it by a different edge $f \in C$.

(b*) Show that an edge which is not heavy, must be light. Conclude that an edge is heavy if and only if it is not light.

Hint: You may use without proof that Kruskal's algorithm is correct regardless of the order in which edges of equal weight are processed.