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
Johannes Lengler, Gleb Novikov, Chris Wendler, Ulysse Schaller

07. December 2020

Algorithms & Data Structures — Exercise sheet 12 — HS 20

Exercise Class (Room & TA): ____________________

Submitted by: ____________________

Peer Feedback by: ____________________

Points: _____

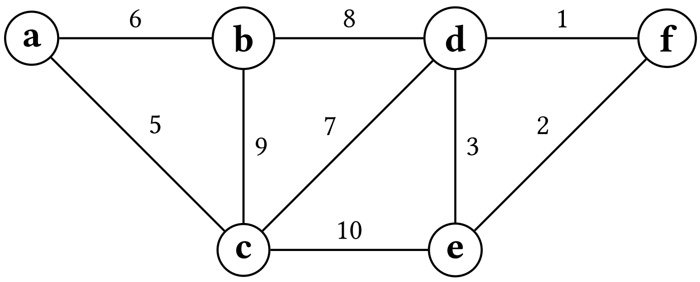
Submission: On Monday, 14. December 2020, hand in your solution to your TA *before* the exercise class starts. Exercises that are marked by * are challenge exercises. **Execeptionnally, they will also count toward bonus points.**

Remark. Let $G = (V, E)$ be a weighted graph with nonnegative weights ($w(e) \geq 0 \quad \forall e \in E$) such that all edge-weights are different ($\forall e \neq e'$ in E, $w(e) \neq w(e')$). Then the minimum spanning tree of G is unique.

You can use this fact without further justification for solving this exercise sheet.

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 a) adds the edges to the MST.

Exercise 12.2 *Ancient Kingdom of Macedon.*

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*, which was much faster than all the alternatives in the Ancient Macedon Kingdom. However, the Tesla Carriage required *asphalt roads* to operate, and such 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.

*Exercise 12.3 *Spanning Forest with 2 components* ***(2 points)***.

Let $G = (V, E)$ be a connected edge-weighted graph in which all weights of the edges are different and positive. Consider the following two algorithms, which take G, the weights of all edges, and two different vertices $u, v \in V$ as input.

Algorithm 1

Run Kruskal's algorithm to get a minimum spanning tree $T = (V, E_T)$.
Find the unique path π from u to v in T.
Find the edge e of maximal weight among edges in π.
Remove e from T to get a graph $H = (V, E_T \setminus \{e\})$.
return H

Algorithm 2

$M \leftarrow \{u, v\}$
$E_M \leftarrow \emptyset$
while $M \neq V$ **do**
    $\Delta M = \{\{w_0, w_1\} \in E : w_0 \in M, w_1 \in V \setminus M\}$
    Find the edge $\{w_0, w_1\} \in \Delta M$ of minimal weight.
    $E_M \leftarrow E_M \cup \{\{w_0, w_1\}\}$
    $M \leftarrow M \cup \{w_1\}$
return $H = (M, E_M)$

Prove that the two algorithms return the same graph.

Hint: *Consider the graph G' which is obtained from G by adding an edge e_0 of weight 0 between u and v (if the edge $\{u, v\}$ already exists, then simply decrease its weight to 0). Try to relate the two given algorithms on G to algorithms that you know from the lecture on G'.*

Exercise 12.4 *Counting walks.*

Recall that a *walk of length* k in a graph $G = (V, E)$ is a sequence of vertices $v_0, v_1, \ldots, v_k$ such that for $1 \leq i \leq k$ there is an edge $\{v_{i-1}, v_i\} \in E$.

Consider the adjacency matrix representation of a graph $G = (V, E)$, with $V = \{v_1, \ldots, v_n\}$. The adjacency matrix A contains number of walks of length 1 (= edges), i.e., $A_{ij} = 1$ if $\{v_i, v_j\} \in E$ and $A_{ij} = 0$ otherwise. Similarly, A^k contains the number of walks of length k in its entries (see Exercise 9.5.).

a) Provide an efficient algorithm that given an adjacency matrix A and an integer $k \geq 1$ computes A^k using only matrix multiplications.

b) Determine the number of matrix multiplications required by your algorithm in Θ notation. Justify your answer.