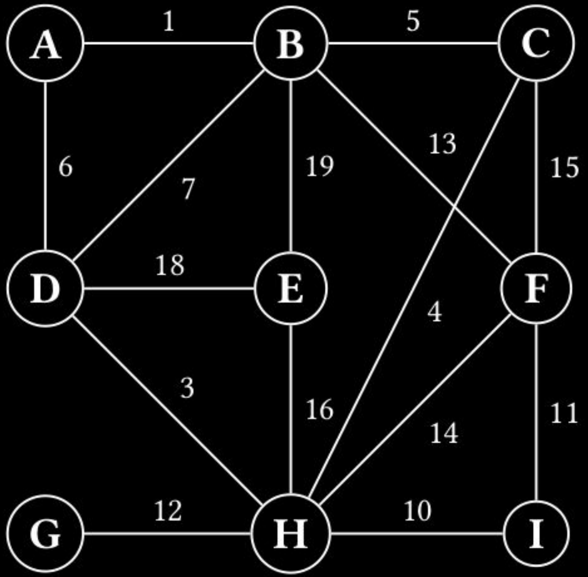
Algorithms and Data Structures

Exercise Session 12

https://n.ethz.ch/~ahmala/and

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

Consider the following graph.



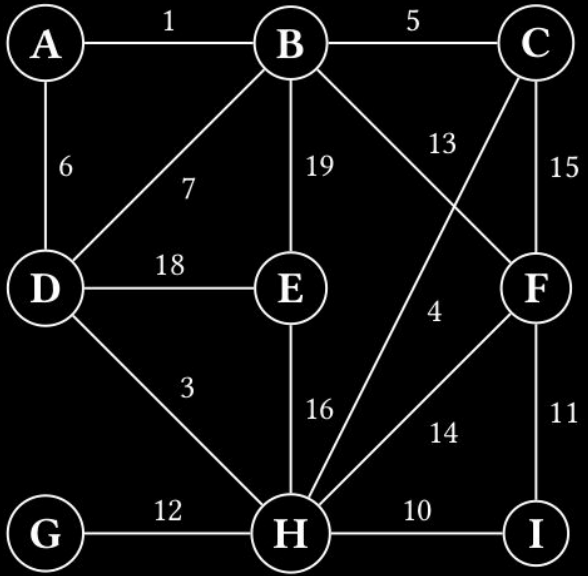
(a) Compute the minimum spanning tree (MST) using Boruvka's algorithm. For each step, provide the set of edges that are added to the MST.

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

Consider the following graph.



Boruvka

1) Input is a connected, weighted and un-directed graph.
2) Initialize all vertices as individual components (or sets).
3) Initialize MST as empty.
4) While there are more than one components, do following for each component.
   a) Find the closest weight edge that connects this component to any other component.
   b) Add this closest edge to MST if not already added.
5) Return MST.

(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.

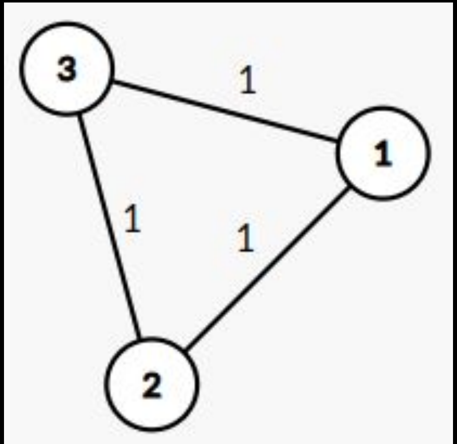
It turns out that for a connected graph, if the weights of the edges are pairwise distinct, the minimum spanning tree is unique. To show this, let $G = (V, E)$ be a connected graph and $w : E \to \mathbb{R}_{\geq 0}$ be a

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



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 $T_1 \setminus T_2$ or $T_2 \setminus T_1$, 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 $e \in T_1 \setminus T_2$.

(b) Show that $T_2 \cup \{e\}$ has a cycle and that there is an edge $f \in T_2 \setminus T_1$ 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 12.4 *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 $\mathrm{tst}(G)$ for the length of a shortest TST in G, that is:

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

(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 $w(M) \leq \mathrm{tst}(G)$.

(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 $O(m)$.*

HS-21 Exam

Peer Grading

Exercise 12.2

Last bonus

Exercise Sheet 13 won’t be graded

https://docs.google.com/spreadsheets/d/1owPsJsd9THBWInwFcVjKCc0f_r6n4pGwwDKMDdwaCjM/edit?usp=sharing


WELL THAT'S THE END