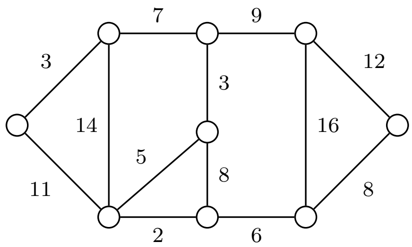
Universität Siegen
Lehrstuhl Theoretische Informatik
Markus Lohrey

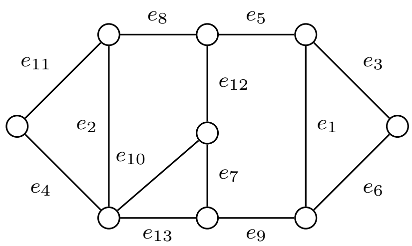
Algorithmics 1
WS 2020/21

Exercise 5

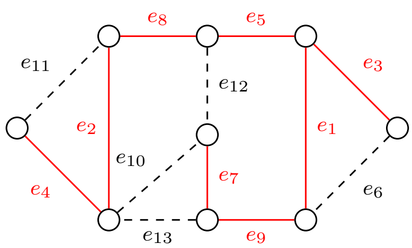
Task 1
Compute a spanning subtree of maximal weight using Kruskal's algorithm for the following graph:



Solution
We first sort the edges by their weights in decreasing order. To illustrate better what it yields, we show the graph one more time:



Kruskal's algorithm now takes greedily any heavy edge into the set F, such that (V, F) has no cycles.
In the end the spanning subtree has the following edges: $F = \{e_1, e_2, e_3, e_4, e_5, e_7, e_8, e_9\}$.



Task 2

(a) Show that for each tree $T = (V, E)$ with $|V| > 0$ we have $|E| = |V| - 1$.

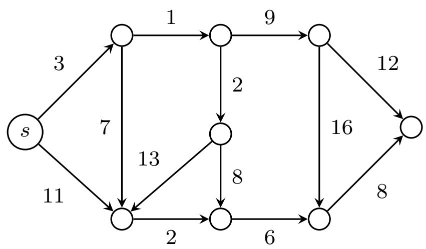
(b) Show that every connected graph has a spanning subtree.

Solution

(a) We do an induction on $|V|$. In case $|V| = 1$ it is clear that $|E| = 0$. Now let $|V| > 1$. Since T has no cycles, there is a leaf in T, meaning there is a $v \in V$ with $|v_E| = 1$, where $v_E = \{\{v, u\} \in V^2 \mid \{v, u\} \in E\}$. So $|v_E| = 1$ means that v borders only one edge, which means that the node u with $\{v, u\} \in E$ is the parent node of v. Let $T' = (V', E')$ with $V' \setminus \{v\}$ and $E' = E \setminus v_E$. This is a tree, since it is connected, because T is connected and v is a leaf, and it also has no cycles, since $E' \subseteq E$ and T has no cycles. Furthermore, $|V'| = |V| - 1$, so by induction hypothesis we obtain $|E'| = |V'| - 1$. We have now proven that $|E| = |E'| + 1 = |V'| - 1 + 1 = |V'| = |V| - 1$.

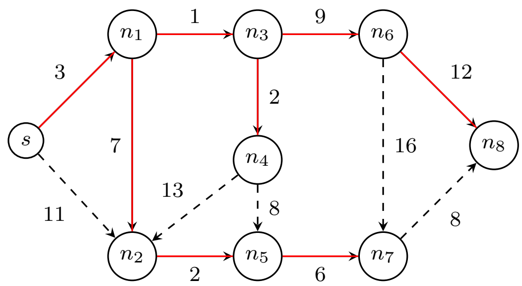
(b) Let $G = (V, E)$ be a connected graph. If G is a tree, G is a spanning tree of G. Otherwise, choose an edge $e \in E$ that is on a cycle in G and let $E' = E \setminus \{e\}$. Now $G' = (V, E')$ is still connected and $E' \subset E$. We set $G = G'$ and iterate the above step. This algorithm terminates because G is finite and we remove one edge in each step. Repeatedly removing edges on cycles in a finite graph eventually leads to a graph that has no cycles and is therefore a (spanning) subtree.

Task 3
Use Dijkstra's algorithm to compute all shortest paths starting at node s.



Solution
For Dijkstra's algorithm it is useful to draw a table and indicate the tree nodes, the boundary nodes and the unknown nodes. The latter ones have distance ∞ to the treenodes (where distance is **bold**), since they cannot be reached in one step. The shortest paths are resulting in a tree, highlighted in red.

Node	s	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
Step 0	**0**	3	11	∞	∞	∞	∞	∞	∞
Step 1	**0**	**3**	10	4	∞	∞	∞	∞	∞
Step 2	**0**	**3**	10	**4**	6	∞	13	∞	∞
Step 3	**0**	**3**	10	**4**	**6**	14	13	∞	∞
Step 4	**0**	**3**	**10**	**4**	**6**	12	13	∞	∞
Step 5	**0**	**3**	**10**	**4**	**6**	**12**	13	18	∞
Step 6	**0**	**3**	**10**	**4**	**6**	**12**	**13**	18	25
Step 7	**0**	**3**	**10**	**4**	**6**	**12**	**13**	**18**	25
Step 8	**0**	**3**	**10**	**4**	**6**	**12**	**13**	**18**	**25**



Task 4
Let F_n be the n-th Fibonacci number ($F_1 = F_2 = 1$ and $F_{n+1} = F_n + F_{n-1}$). Show that

$$\sum_{i=1}^{n} F_i^2 = F_n \cdot F_{n+1}$$

and

$$\sum_{i=1}^{2n+1} (-1)^{i-1} F_i = F_{2n} + 1.$$

Solution
We can show the first statement by induction in n. For $n = 1$ we have $1^2 = 1^2$. Assume we already showed the equation for n and want to prove it for $n + 1$. Then

$$\begin{aligned}\sum_{i=1}^{n+1} F_i^2 &= \sum_{i=1}^{n} F_i^2 + F_{n+1}^2 \\ &= F_n \cdot F_{n+1} + F_{n+1}^2 \\ &= (F_n + F_{n+1}) \cdot F_{n+1} \\ &= F_{n+2} \cdot F_{n+1}.\end{aligned}$$

For the second statement, we also do an induction in n. The case $n = 1$ yields $1 - 1 + 2 =$

$2 = 1 + 1$. Going to $n + 1$ we get

$$\begin{aligned}\sum_{i=1}^{2n+3} (-1)^{i-1} F_i &= \sum_{i=1}^{2n+1} (-1)^{i-1} F_i - F_{2n+2} + F_{2n+3} \\ &= (F_{2n} + 1) - F_{2n+2} + (F_{2n+2} + F_{2n+1}) \\ &= F_{2n} + F_{2n+1} + 1 \\ &= F_{2n+2} + 1.\end{aligned}$$

Remark: Both inductions work with $n = 0$ as well (assuming $F_0 = 0$).