Exercise III

Remarks: All the graphs here are without self loops and parallel edges. In all the algorithms, always explain their correctness and analyze their complexity. The complexity should be as small as possible. A correct algorithm with large complexity, may not get full credit.

- **Question 1:** Let $G(V, E)$ be a graph and let $T_1(V, E_1)$ and $T_2(V, E_2)$, with $E_1 \subseteq E$, $E_2 \subseteq E$, be two subtrees of $G$. Show that there are two edges $e_1 \in E_1 \setminus E_2$ and $e_2 \in E_2 \setminus E_1$ such that the two following are trees: $T_1 \setminus \{e_1\} \cup \{e_2\}$, $T_2 \setminus \{e_2\} \cup \{e_1\}$.

**Answer:** Since $T_1 \neq T_2$, there is an edge $e_1 \in T_1 \setminus T_2$. When removing $e_1$ from $T_1$, $T_1$ decomposes into two trees, $T'$ and $T''$. These two trees define a cut, $c(T', T'')$. Now, add $e_1$ to $T_2$. A unique cycle $C$ is closed. Let $e_2$ be any edge in that cycle, $e_2 \neq e_1$. Clearly, $T_2 \setminus \{e_2\} \cup \{e_1\}$ is still a legal tree.

Now, use the cut-cycle lemma. We know that $C$ and $c(T', T'')$ share an even number of edges. But, they clearly share $e_1$, by definition. Hence, there is an edge $e_2 \in C$ that is a cut edge, i.e., goes from $T'$ to $T''$. This edge, re-connects $T_1 \setminus \{e_1\}$. Hence, $T_1 \setminus \{e_1\} \cup \{e_2\}$ is a tree, as well, as required.

- **Question 2:** Let $G$ be a graph with all edge-weights distinct. Show that the minimum spanning tree is unique.

**Answer:** Say for the sake of contradiction, that there are two distinct minimum spanning trees $T_1 \neq T_2$. Let $e_1$ and $e_2$ be as above. Namely, $T_1 \setminus \{e_1\} \cup \{e_2\}$, $T_2 \setminus \{e_2\} \cup \{e_1\}$ are trees.

Say without loss of generality that $w(e_1) < w(e_2)$ (recall, all the edges are distinct). Now, by question 1, $T_2 \setminus \{e_2\} \cup \{e_1\}$ is a tree. However, its weight is $w(T_2) - w(e_2) + w(e_1) < w(T_2)$. This contradicts the fact that $w(T_2)$ is a minimum spanning tree.

- **Question 3:** Let $G$ be a graph and $T$ some given minimum spanning tree of $G$. Show that there is a run of Prim’s algorithm that gives $T$ as a result (in this run, whenever there are many choices for the next edge, we will chose the “correct” edge, so that the result will be $T$).

**Answer:** Let $T_i$ be the tree constructed by Prim’s algorithm, after $i$ edges were added. Thus, $T_0 = \emptyset$. Let $T$ be the “desired” optimum tree. We prove by induction for every $i \leq n - 1$, that we can chose to add edges, so that $T_i \subseteq T$. For $i = 0$, this is obvious. Now, assume that $T_i \subseteq T$. Now, we add the minimum edge $e$, touching $T_i$. Let $V_i$ the
vertices in $T_i$ and consider the cut: $c(V_i, V \setminus V_i)$. Clearly, $e$ above belongs to the cut. Now, if $e \in T$, we get that $T_{i+1} \subseteq T$, as required. Assume now that $e \notin T$.

Add $e$ to $T$. A unique cycle $C$ is closed. By the cut-cycle lemma, $C$ intersects $c(V_i, V \setminus V_i)$ by an even number of edges. By definition, $e$ belongs to $C \cap c(V_i, V \setminus V_i)$. Hence, there must be at least one more edge $e' \in T$, such that $e' \in C \cap c(V_i, V \setminus V_i)$. Now, we claim that $w(e') \leq w(e)$. Otherwise, the tree $T \setminus \{e'\} \cup \{e\}$ is a legal tree, with weight less than the optimum, $w(T)$, which is a contradiction. (This actually implies that $w(e') = w(e)$.) Now, replace $T_{i+1} = T_i \cup \{e\}$, by $T'_{i+1} = T_i \cup \{e'\}$. Clearly, $T'_{i+1} \subseteq T$, and $e'$ is legal edge to chose, since $e'$ is one of the minimum edges touching $T$. This gives the required result, since clearly $T_{n-1} = T$.

- **Question 4:** Let $T$ be a tree, and let $d_1, \ldots, d_n$ be the degrees of the vertices. Assume without loss of generality that $d_1 = 1$. Define: $d_1^* = d_1 + 2, d_2^* = d_2, \ldots, d_{n-1}^* = d_{n-1}, d_n^* = d_n$.

  1. Prove that the average of $d_i^*$, $\sum_{i=1}^{n} d_i^* / n = 2$.

  **Answer:** The sum of degrees is twice the number of edges. Since in a tree $|E| = |V| - 1$, we get that $\sum_i d_i = 2 \cdot n - 2$. It follows by definition that $\sum_i d_i^* = 2n$. Hence, $\sum_i d_i^* / n = 2$.

  2. Let $B = \{i \mid d_i \neq 2\}$. Prove that the average of the degrees in $B$ is 2, namely, $\sum_{i \in B} d_i^* / |B| = 2$.

  **Answer:** Since the average is 2, removing vertices with $d_i^*$ exactly 2, clearly does not change the average. More precisely, $(\sum_i d_i^* - 2)/(n - 1) = 2$, still, so removing one element of value 2 has no affect on the average. And so on.

  3. Let $C = \{i \mid d_i \geq 3\}$ and $L$ be the set of leaves (degree 1 vertices). Show that: $|L| = 2 + \sum_{i \in C} (d_i - 2)$. This gives a relation between the number of leaves and the degrees larger than 2 in $T$.

  **answer:** By the above we have:

  \[
  \frac{2 + \sum_{v \in L} 1 + \sum_{v \in B} d(v)}{|L| + |B|} = 2
  \]

  Rearranging, we get that $|L| = 2 + \sum_{v \in B} d(v) - 2 \cdot |B|$. Since $\sum_{v \in B} d(v) - 2 \cdot |B| = \sum_{v \in B} (d(v) - 2)$, the required is derived.

- **Question 5:** Give an algorithm for finding a tree $T(V, E')$ with minimum maximum weight, namely, if $w_{\text{max}} = \max_{e \in E'} w(e)$, the algorithm finds the a tree $T$ that minimizes $w_{\text{max}}$. 

2
Answer: The point to observe is that any minimum spanning tree has minimum $w_{\text{max}}$. For otherwise, let $T^*$ be a minimum spanning tree. Let $w^* = w(e^*)$ be its largest weight. Let $T$ be a spanning tree, with maximum weight $w_{\text{max}} < w^*$. Let $c(V, \bar{V})$ be the cut resulting in $T^*$, if we remove the edge $e^*$. Because $T$ is connected, there must be an edge $e \in T$ that crosses the cut $c$. Thus, $w(e) \leq w_{\text{max}} < w^*$. But the tree $T^* \setminus \{e^*\} \cup \{e\}$ has weight less than the optimum. This is a contradiction.