Exercise 1

For the proof of the first part of the exercise, we need the following lemma.

**Lemma 1.** Let $T = (V, E)$ and $T' = (V, E')$ be two spanning trees of a graph $G$ and let $e \in E \setminus E'$ be an edge that belongs to $T$ but not to $T'$. Then there is an edge $e' \in E \setminus E$ such that $(V, E \setminus \{e\} \cup \{e'\})$ is a spanning tree of $G$.

**Proof.** Removing $e$ separates the tree $T$ into two subtrees $T_1$ and $T_2$. There must be at least one edge $e'$ of $T'$ which connects a node in $T_1$ with a node in $T_2$. The tree $(V, E \setminus \{e\} \cup \{e'\})$ is a spanning tree of $G$. \qed

- We can now show how $B$ can win the game if $G$ contains two edge-disjoint spanning trees. Once there is a blue spanning tree, $B$ clearly wins the game. $B$ can obtain a blue spanning tree by applying the following strategy. At all times, $B$ maintains the invariant that there are two spanning trees $T_0$ and $T_1$ consisting of blue and uncolored edges of $G$ such that the uncolored parts of $T_0$ and $T_1$ are edge-disjoint. At the beginning, $B$ can find two such spanning trees because we assume that $G$ contains two edge-disjoint spanning trees. If $R$ colors an edge $e$ of $T_i$, by the above lemma, there is an edge $e'$ of $T_{1-i} \setminus T_i$ such that $T_i \setminus \{e\} \cup \{e'\}$ is a spanning tree of $G$. By coloring $e'$ blue, $B$ can maintain the invariant that there are two spanning trees consisting of blue and uncolored edges such that the uncolored parts are edge-disjoint. The two trees are now $T_i \setminus \{e\} \cup \{e'\}$ and $T_{1-i}$. If $R$ colors an edge that does not belong to $T_0$ or $T_1$, $B$ can color an arbitrary edge to maintain the invariant. Because in the end, all edges are colored, the invariant implies that there will be a blue spanning tree and $B$ therefore wins the game.

- Yes, $R$ can always win if $G$ does not contain two edge-disjoint spanning trees. For the sake of contradiction assume that $G$ does not contain two edge-disjoint spanning trees and that $B$ has a winning strategy. $B$ then has a strategy to construct a blue spanning tree of $G$. Because $R$ starts the game, $R$ can achieve whatever $B$ can achieve. In particular, $R$ can start by coloring an arbitrary edge and then play $B$’s strategy from the second step on. To win, $B$ has to construct a blue spanning tree of $G$. If $B$ can win, $R$ therefore can construct a red spanning tree. However, if $R$ constructs a red spanning tree, $B$ cannot construct a blue spanning tree because $G$ does not contain two edge-disjoint spanning trees.

**Remark:** In the literature, the described kind of games are also known as Maker-Breaker games. There are two players Maker and Breaker which alternately claim edges of a graph $G$. Maker tries to claim a set of edges which satisfies a certain property, Breaker tries to avoid this. The game of the exercise is known as “Connectivity Game.”
Exercise 2

Let the weight of a subtree be the sum of the weights of its vertices. We first show the following lemma.

**Lemma 2.** For every $t \geq 1$, it is possible to remove a vertex $v$ from $T$ such that one of the remaining subtrees has weight at most $Wt/(t+1)$ and all other subtrees have weight less than $W/(t+1)$.

**Proof.** Let $r$ be an arbitrary vertex of $T$ and assume that $T$ is rooted at $r$. We want to find a vertex $v$ with the following properties. The total weight in the subtree rooted at $v$ is at least $W/(t+1)$ and the weight of each of the subtrees rooted a child vertex of $v$ is less than $W/(t+1)$. Such a vertex $v$ satisfies the conditions of the lemma. When removing $v$, one of the remaining subtrees has weight at most $Wt/(t+1)$ (all nodes that are not in the subtree rooted at $v$) and all other subtrees have weight less than $W/(t+1)$ (the subtrees rooted at the children of $v$). We can find such a $v$ with the following walk on $T$. We start at the root $r$ and set $v := r$. As long as $v$ has a child vertex $v'$ such that the subtree rooted at $v'$ has weight at least $W/(t+1)$, we set $v := v'$. As soon as the weights of the subtrees of all children of $v$ are less than $W/(t+1)$, we have found a vertex $v$ with the desired properties. □

We can now use Lemma 2 to show both parts of the exercise.

- Using $t = 1$ in Lemma 2 directly gives what we need.
- We apply induction on $k$. For $k = 1$, the statement is equivalent to the first part of the exercise. For $k > 1$, we use Lemma 2 with $t = k$ to separate $T$ into one subtree $T'$ of weight at most $W' = Wk/(k+1)$ and subtrees of weight at most $W/(k+1)$. By the induction hypothesis, we can find $k-1$ vertices to separate $T'$ into subtrees of weight at most

$$\frac{W'}{k-1+1} = W \cdot \frac{k}{k+1} \cdot \frac{1}{k} = \frac{W}{k+1}.$$