Graphs and Algorithms

Exercise 1 (Connectivity)

(a) Take a minimum-length walk \( W = u, v_1 \ldots v_k, v \). It is a path: assume it is not, then there are internal vertices \( v_i \) and \( v_j, j > i \) of \( W \) with \( v_i = v_j \). We can construct a new shorter walk \( u \ldots v_i, v_{j+1} \ldots v \) (i.e., “leave out the loop between \( v_i \) and \( v_j \)”), which contradicts the choice of \( W \).

(b) \(-s \sim s\) by \( s = s \).

\(-s \sim t \implies t \sim s\) by reversing the path (walking the vertices in reverse order).

\(-s \sim t \) and \( t \sim u \) \( \implies s \sim u\): if either \( s = t \) or \( t = u \) there is nothing to prove. Otherwise there are paths \( s-t \) and \( t-u \). Concatenating them yields an \( s-u \)-walk, and by (a) there is a \( s-u \)-path.

(c) Let us assume that \( G \) is not connected, and let \( C_1, \ldots, C_k \subseteq G \) be the connected components of \( G \). Then for every two vertices \( x \in V(C_i) \) and \( y \in V(C_j) \), for \( i \neq j \), we have \( \{x, y\} \in E(\overline{G}) \). On the other hand, if \( x, y \in C_i \) then let \( j \in \{1, \ldots, k\} \setminus \{i\} \) be arbitrarily chosen (observe that \( k \geq 2 \), thus \( j \) is well defined). By the previous observation we have that \( \{x, z\}, \{y, z\} \in E(\overline{G}) \) for every \( z \in V(C_j) \), and so there exists a path of length at most two between \( x \) and \( y \) in \( \overline{G} \). This proves that \( \overline{G} \) is connected, and furthermore that the distance between any two vertices is at most two.

(d) Let \( P = v_1, \ldots, v_t \) be a longest path in \( G \). Then \( \Gamma(v_1) \subseteq \{v_2, \ldots, v_1\} \), as otherwise we could extend \( P \) by a vertex which is a contradiction with the assumption that \( P \) is a longest path. Since \( \deg(v_1) \geq k \) it follows that \( t \geq k + 1 \) and furthermore \( \Gamma(v_1) \setminus \{v_2, \ldots, v_k\} \neq \emptyset \). Therefore, there exists \( k' \in \{k + 1, \ldots, t\} \) such that \( v_{k'} \in \Gamma(v_1) \) and observe that \( C = v_1, \ldots, v_{k'} \), \( v_1 \) is a cycle of length at least \( k + 1 \).

Exercise 2 (Properties of Trees)

(a) (i) Since \( T \) is a connected graph with at least 2 vertices, there are edges in \( T \). Hence, any maximal path in \( T \) has length at least 1, and thus two distinct end vertices.

Let \( v_0, v_k \) be the end vertices of such a path \( P = (v_0, v_1, \ldots, v_k) \). As \( T \) is acyclic, the only neighbor of \( v_0 \) on \( P \) is \( v_1 \) and the only neighbor of \( v_k \) on \( P \) is \( v_{k-1} \). Because \( P \) was chosen to be maximal, both \( v_0 \) and \( v_k \) have no neighbors outside of \( P \). This shows that \( v_0 \) and \( v_k \) are two distinct leaves.
(ii) Let $v$ be a vertex with degree $\Delta(T)$. We say that a path $P$ is a $v$-path, if $P = v, a_1, \ldots, a_k$. Let $\mathcal{F}$ be a maximal, with respect to the number of edges, family of edge-disjoint $v$-paths. Let $F_1, F_2 \in \mathcal{F}$ be two arbitrary paths from the family. The only vertex in the intersection of $F_1$ and $F_2$ is $v$, as otherwise we would have a cycle.

We will prove that end of each path in $\mathcal{F}$ is a leaf. Let $F_1 = v, a_1, \ldots, a_k$ be an arbitrary path in $\mathcal{F}$. If $a_k$ is not a leaf, then there is a vertex $t \notin F_1$, which is connected to $a_k$. It can not be that $t \in F_1$, for some $F_1 \in \mathcal{F}$, as we would have a cycle. Therefore, we can extend $F_1$ with the edge $\{a_k, t\}$, which contradicts the maximality of $\mathcal{F}$.

The only thing that is left to do is to prove that $|\mathcal{F}| \geq \Delta(T)$. Assume that this is not the case. As the degree of a vertex $v$ in any $v$-path is 1, there exists an edge $e$ incident to $v$ which is not part of any $F \in \mathcal{F}$. This contradicts the maximality of $\mathcal{F}$.

(iii) Let us by $L, D$ and $R$ denote the set of vertices with degree 1, 2 and at least 3, respectively. We have

$$\sum_{v \in L} 1 + \sum_{v \in D} 2 + \sum_{v \in R} \text{deg}(v) = \sum_{v \in T} \text{deg}(v) = 2n - 2.$$

By the definition of $R$ and previous observation, we have $2n - 2 \geq |L| + 2|D| + 3|R|$. Clearly, $|D| = n - |L| - |R|$. By plugging in this into previous inequality, we get

$$2n - 2 \geq |L| + 2(n - |L| - |R|) + 3|R| = 2n - |L| + R.$$

It follows easily that $|R| \leq |L| - 2$.

(b) (i) $\Rightarrow$ (ii)

We need to show that $G$ has $n - 1$ edges. We proceed by induction on $n$. If $n = 1$ the statement is trivial, thus we can assume that $n \geq 2$. Since $G$ is a tree we know from (a)(i) that $G$ contains a leaf $v$. Let $G' = G - v$, a graph obtained from $G$ by removing the vertex $v$ and the edge incident to it. If $G'$ is a tree then by induction we have that $G'$ contains $|V(G')| - 1 = n - 2$ edges. Since $v$ has degree exactly 1, this implies that $G$ contains $n - 1$ edges.

It remains to prove that $G'$ is a tree. If $G'$ is not connected, then there exist two vertices $u, w \in V(G')$ such that there is no path between them in $G'$. However, since $G$ is connected there exists a path from $u$ to $w$ in $G$, and thus this path has to contain vertex $v$. As every vertex in path, except the endpoints, has the degree two, this contradicts the fact that $v$ is a leaf. Finally, since $G'$ is a subgraph of $G$, every cycle in $G'$ is also a cycle in $G$. Therefore, $G'$ is connected and has no cycles – thus it is a tree.

(ii) $\Rightarrow$ (iii)

We need to show that $G$ has no cycles. Assume the opposite and let $e \in E(G)$ be an edge contained in a cycle in $G$. Then the graph $G - e$, in
which $e$ is removed, is still connected. If it wasn’t, the end vertices $u$ and $v$ of $e$ would have to lie in different components of $G - e$, but they are still connected by the remaining part of the cycle.

As long as there are cycles in $G$, we repeatedly remove an edge contained in a cycle. The resulting graph $G'$ is then a connected and acyclic graph. By the statement proven above, $G'$ has $n - 1$ edges. As this was also the number of edges of the original graph $G$, there were no edges removed and hence also no cycles in $G$.

$(iii) \implies (i)$

We need to show that $G$ is connected. Let $G_1, G_2, \ldots, G_k$ be the connected components of $G$. Then each $G_i$ is connected and acyclic and therefore, by the statement proven above, has $|V(G_i)| - 1$ edges. Hence, we get that

$$n - 1 = |E(G)| = \sum_{i=1}^{k} |E(G_i)| = \sum_{i=1}^{k} (|V(G_i)| - 1) = n - k.$$  

As $k$ is the number of connected components of $G$, this proves that $G$ is connected.

$(i) \implies (iv)$

Since $G$ is connected there exists at least one $u, v$-path in $G$. Now assume that there are two different $u, v$-paths. Set $a_0 := b_0 := u$ and $a_s := b_t := v$ and let $(a_0, a_1, \ldots, a_s)$ and $(b_0, b_1, \ldots, b_t)$ denote two different $u, v$-paths for suitable $s, t$. Clearly, there exists a minimum index $i$ such that $a_i \neq b_i$ and a minimum index $j > i$ such that $a_j = b_k$ for some $k > i$. Then $(a_{i-1}, a_i, \ldots, a_j = b_k, b_{k-1}, \ldots, b_{i-1})$ forms a cycle in $G$, which is a contradiction. Hence, there are no two different $u, v$-paths in $G$.

$(iv) \implies (i)$

It is easy to see that $G$ is connected. Now assume that $G$ has a cycle $(v_1, \ldots, v_s, v_1)$ for some $s$. Then $(v_1, v_s)$ and $(v_1, v_2, \ldots, v_s)$ are two different $v_1, v_s$-paths in $G$, which is a contradiction. Hence, $G$ contains no cycles.

(c) Look at (b), direction from $(i)$ to $(ii)$.

(d) Let $T$ be a tree and $e = \{u, v\} \in E(T)$. By (b), $(u, v)$ is the only $u, v$-path in $T$. Hence, there is no $u, v$-path in $T - e$ and thus $T - e$ is not connected.

(e) Let $T$ be a tree on $n$ vertices and $e = \{u, v\} \in \binom{V(T)}{2} \setminus E(T)$ be an edge which is not in $T$ and set $T^+ := (V(T), E(T) \cup \{e\})$. Since $T^+$ is connected and contains more than $n - 1$ edges it must contain at least one cycle by (b). Since $T$ is cycle-free, the edge $\{u, v\}$ must be contained in every cycle of $T^+$. Let $C_1 = (v, u, a_1, \ldots, a_s, v)$ and $C_2 = (v, u, b_1, \ldots, b_t, v)$ be two cycles in $T^+$. Then $(u, a_1, \ldots, a_s, v)$ and $(u, b_1, \ldots, b_t, v)$ are $u, v$-paths in $T$. By (b) this path is unique and hence $C_1 = C_2$.

(f) Let $G$ be a connected graph. We use the procedure already introduced in the proof of $(ii) \implies (iii)$ in (a). As long as $G$ contains cycles, remove an arbitrary edge that is contained in a cycle. At the end of the process, we
obtain an acyclic connected subgraph of $G$ that still contains all vertices, i.e. a spanning tree.

\[ (g) \] First, notice that the graph from the statement is well-defined. More precisely, $|E(T_1) \setminus E(T_2)| = 1$ is a proper edge definition, because if $|E(T_1) \setminus E(T_2)| = 1$, then $|E(T_2) \setminus E(T_1)| = 1$.

Let $F$ and $H$ be two arbitrary different trees on the vertex set $V$. By $D$ we denote the edges contained in $H$ that are not in $F$, i.e $D = E(H) \setminus E(F)$. Let $F' = F + e_1$, where $e_1$ is an arbitrary edge from $D$. As $F$ is a tree, by (e) we have that $F'$ has exactly one cycle. There must exist one edge in this cycle that is not contained in $H$, as otherwise $H$ wouldn’t be a tree.

Let us call this edge $f$ and define $F_2 := F' - f$. Observe that $F_2$ has no cycles and $n - 1$ edges and hence $F_2$ is also a tree by (b)(iii). Notice that the tree $F_2$ has one more edge in common with $H$ than $F$ and $\{F, F_2\} \in \mathcal{E}$. More formally, $D_2 := E(H) \setminus E(F_2)$ and $|D_2| = |D| - 1$. Clearly, we can similarly find a tree $F_3$ with $|E(H) \setminus E(F_3)| = |D| - 2$ and $\{F_2, F_3\} \in \mathcal{E}$. By repeating this argument until $E(H) = E(F_i)$, we have proved that there is a sequence of trees $F, F_2, \ldots, F_i = H$, such that each two consecutive trees form an edge from $\mathcal{E}$.

Exercise 3 (Bridg-It)

(a) This is a very hard question to prove formally. One possible proof idea is the following. Clearly not both players can win. If this were the case then any winning path from left to right would intersect all winning paths from top to bottom. However by the rules of the game this is not possible (edges from different players do not cross).

Assume that the player going from left to right does not create a path from one side to the other. Let $C$ denote the connected component containing the left side of the board. Clearly it does not contain the right side or a path to it. The boundary of this component must be claimed by the other player (otherwise we could extend $C$ or there are not-chosen edges).

However this boundary must go from the top of the board to the bottom as $C$ contains the entire left side.

The same holds symmetrically for the top to bottom player as well, therefore one must always win.

(b) Let us consider the following lemma.

\textbf{Lemma.} Let $G$ be a graph and $T_1, T_2$ two spanning trees of $G$. Then for all $e \in E(T_1) \setminus E(T_2)$ there exists an edge $e' \in E(T_2)$ such that $(V(G), E(T_1) \setminus e) \cup e'$ is a spanning tree.

\textit{Sketch of the proof.} Assume $e = \{v, w\}$ cuts $T_1$ into two components $T'_1$ and $T''_1$. Note that these must both be spanning trees on the vertices of their respective components. As $e \notin E(T_2)$ the tree $T_2$ must contain a $v$-$w$ path which does not use the edge $e$. This path also contains a cut-edge in $T_2$ between the vertex-sets $V(T'_1)$ and $V(T''_1)$. Choosing $e'$ as such a cut-edge proves the lemma. □
We use this lemma as follows to find a winning strategy. Let $E_i$ denote the set of possible edges that Player 1 can choose in round $i$ together with the $i$ edges chosen by Player 1 in rounds $1, \ldots, i$ (note that $E_i$ contains the move by Player 1 in round $i$). Then the Player 1 can maintain the following invariant:

(*) for each round $i$, just before Player 2 moves, there are two (not necessarily different) spanning trees $T_{1,i}$, $T_{2,i}$ of $G$, with edges in $E_i$, such that any edge in $E(T_{1,i}) \cap E(T_{2,i})$ is claimed by Player 1.

Let $t$ be the last round of the game. Then $E_t$ is the set of all edges claimed by the Player 1 together with one additional edge $e$. It is not possible that $e \in E(T_{1,t}) \cap E(T_{2,t})$, as then by the invariant we have that $e$ is claimed by the Player 1. Therefore, without loss of generality we may assume that $e \not\in E(T_{1,t})$ and thus all edge edges of $T_{1,t}$ are claimed by the Player 1.

Maintaining the invariant before the first move of Player 2 is easy: the board of the game, viewed as a graph, contains two spanning trees $T_{1,1}$, $T_{2,1}$ which have only one edge in common. See the lectures slides for an example of two such trees. Player 1 claims this common edge as her first move.

Now assume that the invariant holds in round $i$ and Player 2 makes a move and then Player 1 plays again for round $i+1$. Player 2 must choose an edge $e$ which cuts across one edge of $T_{1,i}$ (or $T_{2,i}$, but without loss of generality assume it cuts $T_{1,i}$) which is not contained in $T_{2,i}$. Player 1 in her next move claims an edge $e'$ of $T_{2,i}$ which fixes the cut in $T_{1,i}$, which by the lemma above is always possible. She then sets $T_{1,i+1} = (T_{1,i} \setminus e) \cup e'$ and $T_{i+1,2} = T_{i,2}$. These new trees $T_{i+1,1}$ and $T_{i+1,2}$ satisfy the invariant again.

**Exercise 4 (Strategy Stealing)**

Player 2 might not be able to ignore Player 1’s first move! If Player 2 plays according to Player 1’s strategy, this strategy may tell him at some point to select the edge which Player 1 claimed in his first move. (This case will indeed occur if Player 1 uses a winning strategy.)

In the original strategy stealing argument (where Player 1 steals Player 2’s strategy) the strategy may tell Player 1 to claim an edge that Player 1 already claimed, but never an edge which was already claimed by Player 2!

**Exercise 5 (Bridg-It on Graphs)**

Assume that B has a winning strategy, a strategy which guarantees her to claim a spanning tree. Then by the strategy stealing argument, R can play an arbitrary edge in the first round, and after that do what B would do in her situation ignoring the edge she played on the first round. Note that now the goal of R is not to prevent B from building a spanning tree, but rather claiming one for herself. Similarly as for the Bridg-It game, it can be shown that having more edges doesn’t harm, thus at the end of the game both red and blue edges contain a spanning tree. However, this is a contradiction with the assumption that $G$ doesn’t contain two edge-disjoint spanning trees!