ETH Zürich
Institute of Theoretical Computer Science
Dr. J. Lengler
H. Einarrson, R. Nenadov

FS 2013
Proposed solutions for sheet 0

Graphs and Algorithms

Exercise 1 (Connectivity)

(a) Take a minimum-length walk $W = u, v_1 \dots v_k, v$. 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 $u \dots v_i, v_{j+1} \dots v$ (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) The connected components are exactly $V(G)/\sim$.

Exercise 2 (Properties of Trees)

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

Let v_0, v_k be the end vertices of such a path $P = (v_0, v_1, \dots, v_k)$. 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_0 be a leaf of T and $T' := T - v_0$. Clearly, deleting v_0 cannot create a cycle and T' is thus cycle-free. It remains to show that T' is connected. Let $u, v \in V(T')$. Since T is connected there exists a u,v-path $P = (u, v_1, v_2, \dots, v_s, v)$ in T. Clearly, $\deg(v_i) \geq 2$ for all $1 \leq i \leq s$, hence v_0 cannot be in P and P is also a u,v-path in T'.

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

We need to show that G has $n-1$ edges. For $n = 1$ the statement is trivial. We proceed by induction on n. Since G is a tree we know from $(a)(i)$ that G contains a leaf v. From $(a)(ii)$ we know that $G - v$ is also connected and has no cycles. Hence by induction, $G - v$ has $|V(G - v)| - 1 = n - 2$ edges, and thus G has $n - 1$ edges.

$(ii) \Rightarrow (iii)$

We need to show that G has no cycles. Assume that this was not true 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 a path.)

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.

$(iii) \Rightarrow (i)$

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.

$(i) \Rightarrow (iv)$

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.

$(iv) \Rightarrow (i)$

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) 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 $T - e$ is thus not connected.

(d) 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 $C_1 = C_2$.

(e) Let G be a connected graph. We use the procedure already introduced in the proof of $(ii) \Rightarrow (iii)$ 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.

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) Consider this lemma

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.*

We use this lemma as follows to find a winning strategy. In each round of the game Player 1 can maintain the following invariant just before Player 2 moves. Let E_i denote the set of possible edges that Player 1 can choose in round i together with the i edges chosen in rounds $1, \dots, i$ (i.e. E_i contains the move by Player 1 in round i already). Then

- For each round i, just before Player 2 moves, there are two spanning trees $T_{1,i}$, $T_{2,i}$ with edges in E_i such that any edge in $E(T_{1,i}) \cap E(T_{2,i})$ is claimed by Player 1.

If this is the case then at the end of the game Player 1 must have claimed a spanning tree.

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 notes 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 $T_{i+1,1}$ and $T_{i+1,2}$ satisfy the invariant again.

It remains to prove the lemma. 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 component. 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.

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 Player 2 (blue) has a winning strategy, a strategy which guarantees her to claim a spanning tree. Then by the strategy stealing argument, Player 1 (red) can play an arbitrary edge in the first round, and after that do what Player 2 would do in her situation ignoring the edge she played on the first round. Note that now the *goal* of Player 2 is not to prevent Player 1 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!