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

FS 2013
Proposed solutions for sheet 1

Graphs and Algorithms

Notation

We briefly list the notation we use. If no confusion can arise, we use the convention $G = (V, E)$ for all graphs G. For a set of vertices S, we denote with $G[S]$ the subgraph *induced* with vertices from S,

$$G[S] = (S, \{\{a, b\} \in G \mid a, b \in S)\}).$$

By slight abuse of notation, we define $G \setminus \{x\}$ to be *(i)* the subgraph of G induced by $V \setminus \{x\}$ if x is a vertex, and *(ii)* the graph $(V, E \setminus \{x\})$ if x is an edge.

We denote with $\deg_G(v)$, where $v \in G$, degree of v (number of edges incident with v) in G, and $\delta(G) = \min\{\deg_G(v) \mid v \in G\}$. If it is clear in which graph we are interested, we simply write $\deg(v)$ instead of $\deg_G(v)$. We call an edge $e \in G$ a *bridge* if number of connected components of $G \setminus \{e\}$ is strictly larger than the number of connected components of G.

Solutions

Exercise 1 (Tree embedding)

We show by induction on d that we can embed any tree T on $d+1$ vertices into a graph G for which $\delta(G) \geq d$.

For $d = 0$ the claim trivially holds. Let us assume that it holds for all $d' < d$, for some $d > 0$, and consider any graph G such that $\delta(G) \geq d$. By deleting an arbitrary leaf v from T, we get a tree $T' = T \setminus \{v\}$ on $d-1$ vertices (exercise 2.a, sheet 0). By induction we have $T' \subseteq G$, that is, there exists a homomorphism $f' : T' \to G$. We now try to extend this homomorphism to T. Let w be the unique neighbour of v in T. Since T' has d vertices, $\deg_{T'}(w)$ is at most $d-1$. On the other hand, we have that the degree of $f(w)$ in G is at least d. Therefore, there exists at least one neighbour of $f'(w)$, say $v' \in G$, such that $v' \neq f(t)$ for all $t \in T'$. This proves that $f : T \to G$,

$$f(v) = \begin{cases} v' & \text{if } t = v', \\ f'(t) & \text{otherwise,} \end{cases}$$

is a homomorphism, and thus T is a subgraph of G.

Exercise 2 (Bridges)

We prove that a bridge in a graph $G = (V, E)$ cannot belong to a cycle. Note that this implies the claim of the exercise. If there are more than $n-1$ bridges then consider the subgraph of G where we remove all non-bridges. By the pigeonhole principle, this subgraph must have a component with at least as many edges as vertices. Thus, by the characterization of trees, the component contains a cycle. Altogether we have shown that there exists a cycle in G which consist only of bridges – contradiction!

It remains to prove that a bridge in G cannot belong to a cycle. Let $e = \{u, v\}$ be an edge which belongs to a cycle in G, and consider a shortest such cycle $C = \{u, c_1, \ldots, c_t, v\}$, with $\{u, c_1\}, \{c_{i-1}, c_i\}, \{c_t, v\} \in E$ for $i \in \{2, t\}$. Since C is a cycle, we have $t \geq 1$. Further, let $a, b \in V$ be any pair of vertices which belong to the same component of G. We prove that there exists a path from a to b in $G \setminus \{e\}$. Consider any path from a to b in G. If the path does not contain e, then we are done. Otherwise, without loss of generality, we may assume that the path traverses e from u to v. This in particular implies that there exists a path from a to u and v to b in $G \setminus \{e\}$. However, by merging three paths, the path from a to u, a path from u to v along the cycle C and the path from v to b, we get a walk from a to b in $G \setminus \{e\}$. Therefore, there also has to exists a path from a to b in $G \setminus \{e\}$.

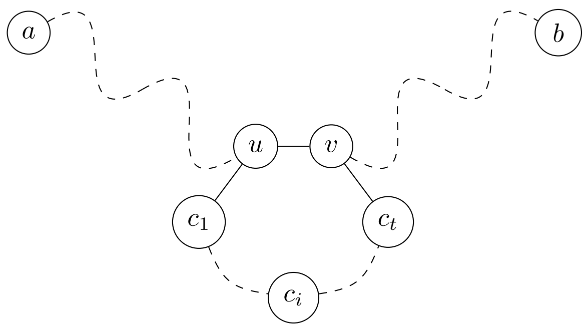

Figure 1: Walk from a to b

We have shown that the connected components of G and $G \setminus \{e\}$ coincide. Therefore, e is not a bridge.

Remark: As an alternative, it is possible to prove that removing a bridge from a graph does not destroy other bridges. Therefore, we can remove the bridges one by one, each time increasing the number of components by 1. Since we start with at least one component, and end up with at most n components, we can remove at most $n-1$ bridges.

Exercise 3 (Maximal graphs)

The statement is false. Consider a graph on $2 \cdot (2k)$ vertices, where the first $2k$ vertices form a clique (graph with all edges) and the rest $2k$ vertices are paired such that each vertex appears in exactly one pair. Clearly, by interpreting any such pair as an edge, we get a graph that is maximal and has $\binom{2k}{2} + k =$

$k(2k-1)+k = 2k^2$ edges, which is less than $\binom{2k+1}{2} = k(2k+1) = 2k^2+k$. The reason why the given proof is incorrect is that it relies on the wrong fact that every maximal graph on $2n$ vertices contains a maximal graph on $2(n-1)$ vertices as a subgraph.

Exercise 4 (Cops and robber)

Let G be a graph with girth at least 5. We want to show that the robber can avoid being caught forever if there are at most $p = \delta(G) - 1$ cops in the play.

Let us denote the positions of cops and the robber with $c_1, \ldots, c_p$ and r, respectively. We will show that the robber can maintain the following invariant: right before the cops turn, the shortest path from robber to any cop is at least 2. We denote the shortest path between vertices u and v with $d(u, v)$.

We first prove the following claim: for any vertex $r \notin \{c_1, \ldots, c_p\}$, there exists a vertex $r' \in \Gamma(r)$ such that $d(r', c_i) \geq 2$ for every $i \in \{1, \ldots, p\}$. Let $\{w_1, \ldots, w_t\} = \Gamma(r)$, and by the assumption on the minimal degree of G we have $t \geq p+1$. Further, let $\Gamma^+(v) = \Gamma(v) \cup \{v\}$ for all $v \in V$. The crucial observation is that $\Gamma^+(w_i) \cap \Gamma^+(w_j) = \{r\}$ for any $i, j \in \{1, \ldots, p+1\}$ and $i \neq j$, as otherwise we would have either a cycle of length 3 or 4 which is a contradiction with the fact that girth of G is at least 5. On the other hand, no cop occupies r. Therefore, each cop can occupy a position in at most one of the $\Gamma^+(w_i)$. Since the number of cops is at most p, there exists an $i \in \{1, \ldots, p+1\}$ such that no cop is in the set $\Gamma^+(w_i)$. Therefore, setting $r' = w_i$ satisfies the claim.

We will now use this claim to prove the exercise. In the initialisation round, after cops have chosen their position, there has to exists at least one vertex r such that $r \notin \{c_1, \ldots, c_p\}$. This simply follows from the fact that there are at least $\delta(G) \geq p+1$ vertices in G. Then by the claim there exists a vertex r' such that $d(r', c_i) \geq 2$ for all i. Choosing r' to be the robber's initial position satisfies the invariant. Further, note that after each move of cops we have $d(r, c_i) \geq 1$ for all i. In particular, we have $r \notin \{c_1, \ldots, c_p\}$, and thus by the claim there exists $r' \in \Gamma(r)$ such that $d(r', c_i) \geq 2$. Therefore, moving the robber from r to r' maintains the invariant! To summarize, we have proven that the robber has a strategy to avoid cops forever if there are at most $\delta(G) - 1$ cops.

Exercise 5 (Spanning trees)

Note that removing any two links between kites (we will call them simply *links*) separates the graph into two components. Thus at least $m-1$ links need to belong to a spanning tree.

Let us first consider the case when there are exactly $m-1$ links in a spanning tree, and let T be any such tree. Let $K = \{k_1, k_2, k_3, k_4\}$ be vertices of a kite, connected as in the figure 2.

If $T[K]$ is not connected, T cannot be a spanning tree, which is a contradiction to our choice of T. To see this, observe that any path from k_i to k_j, $i \neq j$ which is not completely contained in $T[K]$ has to use least one link. However, then such path has to use all links, but since exactly one link is missing in T, this cannot happen. Further, if T' is any spanning tree of a kite K, then

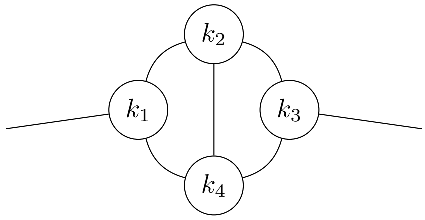

Figure 2: Kite K

$(T \setminus T[K]) \cup T'$ (edge-wise set operations) is again a spanning tree. Therefore, we can construct a spanning tree with exactly $m-1$ links by first choosing which link we omit, and then independently choose a spanning tree of every kite. By simple enumeration we see that there are exactly 8 possible spanning trees of a kite, thus in total we have

$$m \cdot 8^m$$

spanning trees with exactly $m - 1$ links.

Next we consider the case when all links are in a spanning tree T. Then there has to exists a kite K such that $T[K]$ is not connected subgraph, as otherwise we would have that T contains a cycle. On the other hand, if there are two or more kites, say K_1 and K_2, such that both $T[K_1]$ and $T[K_2]$ are disconnected subgraphs, then T itself is also disconnected. Therefore, there exists exactly one kite K which is disconnected. However, we have to be a bit careful here. The subgraph $T[K]$ indeed has to be disconnected, but k_2 and k_4 have to belong to the same component with either k_1 or k_3. Otherwise, even adding all other edges outside of K to a tree T wouldn't "cover" k_2 (or k_4). We can now count the number of spanning trees with exactly m links by first choosing a disconnected kite K, edges of $T[K]$ and edges of every other kite (which now has to form a spanning tree of a kite). From previous consideration, we have that there are exactly 8 possible spanning trees of a connected kite. By simple enumeration we have that there are also 8 possible choices for edges of $T[K]$ such that $T[K]$ is disconnected and k_i belongs to the same component as either k_1 or k_3, for $i \in \{2, 4\}$. Thus, there are

$$m \cdot 8^m$$

spanning trees with m links. In total, there are $2m \cdot 8^m$ spanning trees of the given graph.