Universität
des
Saarlandes
FR 6.2 Informatik

Prof. Dr. Benjamin Doerr, Dr. Danny Hermelin, Dr. Reto Spöhel

Summer 2011

Solution for Exercise 1

Exercise 1+2 (*oral homework, total 8 points via test*)

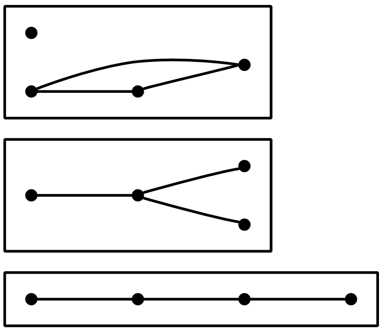
a) If U is independent in G, there are no edges between vertices in U. The graph $\overline{G}$ contains an edge exactly if it does not exist in G.

   Hence there is an edge between any two vertices of U in $\overline{G}$.

b) False. A graph can have up to $\binom{n}{2}$ edges. That is larger than n for $n \geq 4$, for example in complete graphs.

c) Statement $\exists v, u \in V : \{v, u\} \notin E \wedge (\forall w \in V \backslash \{u\} : \{v, w\} \in E)$;
   negation $\forall v, u \in V : \{v, u\} \in E \vee (\exists w \in V \setminus \{u\} : \{v, w\} \notin E)$.

d) The following graphs are non-isomorphic.



e) Assume that all vertices of G have different degrees. Then necessarily all $|V|$ degrees between 0 and $|V| - 1$ are present in the graph. This leads to a contradiction, as the existence of a $v \in V$ with $\deg(v) = |V| - 1$ excludes the existence of isolated vertices, since v must be connected to all other vertices in V.

f) For $n = 2k$, $k \in \mathbb{N}$, we construct a graph G as follows.

$$G = (\{v_1, \ldots, v_k, w_1, \ldots, w_k\}, \{v_i w_j | i \leqslant j, v_i \neq w_j\} \cup \{v_i v_j \,|\, i \neq j\})$$

Then we have $\deg(w_i) = i$ and $\deg(v_i) = 2k - i$. Therefore, only v_k and w_k have the same degree in our construction. For $n = 2k + 1$ we use the same construction but add an isolated vertex.

g) Note: graph as defined in the exercise is commonly called a *d-dimensional hypercube*. It has 2^d vertices, each vertex has d neighbours, as there are d possible positions in which a node can differ from another node in exactly one bit. Since

$$|E| = \frac{1}{2} \sum_{v \in V} \deg(v),$$

in all graphs, we have in particular

$$|E| = \frac{1}{2} \sum_{v \in V} d = \frac{1}{2} 2^d \cdot d = 2^{d-1} \cdot d.$$

Exercise 3 (*written homework, 4 points*)

Let $G = (V, E)$ be a finite directed graph without isolated vertices.

For a node $v \in V$, let $\text{indeg}(v)$ be $|\{(w, v) | (w, v) \in E\}|$ and let $\text{outdeg}(v)$ be $|\{(v, w) | (v, w) \in E\}|$.

Claim: G has an Eulerian tour if and only if it is connected and for every vertex v we have $\text{indeg}(v) = \text{outdeg}(v)$.

Proof: First we show that if G is Eulerian we have $\text{indeg}(v) = \text{outdeg}(v)$ for all $v \in V$ and G is connected. Let $T = v_1, e_1, v_2, \ldots, v_k, e_k, v_1$ be an Eulerian tour of G.

Since there are no isolated vertices, each vertex is contained in the Eulerian tour. Consequently, there is a walk between any two vertices. Hence the graph is connected. Let $w \neq v_1$ be a vertex in G. All occurences of w in T have the form $(x, w), w, (w, y)$ and therefore each occurence contributes one to both $\text{indeg}(w)$ as well as $\text{outdeg}(w)$. As all incident edges to w appear in the tour, $\text{indeg}(w) = \text{outdeg}(w)$. The same is true for occurences of v_1, except of course for the first occurence that only contributes one outgoing edge and the last occurence that only contributes one incoming edge. Therefore $\text{indeg}(v_1) = \text{outdeg}(v_1)$.

Now we show that a longest edge-simple walk in a directed connected graph G without isolated vertices and $\text{indeg}(v) = \text{outdeg}(v)$ for all $v \in V$ must necessarily be an Eulerian tour. Let $W = v_1, e_1, v_2, \ldots, v_k, e_k, v_{k+1}$ be a longest edge-simple walk in G. We show that W is closed. Assume for the sake of contradiction that $v_1 \neq v_{k+1}$. Then e_k is an incoming edge to v_{k+1} that is not matched by an outgoing edge. As $\text{indeg}(v_{k+1}) = \text{outdeg}(v_{k+1})$, there must be another edge $\hat{e} = (v_{k+1}, u)$. Then $\hat{W} = v_1, e_1, v_2, \ldots, v_k, e_k, v_{k+1}, \hat{e}, u$ is a longer walk than W.

Suppose W is not a Eulerian tour and thus there is an edge that is not contained in W. As G is connected, there must either be an edge $e' = (v_i, w)$, $e' \notin E(W)$, that is incident to a vertex v_i on the tour and points away from the tour, or there is an edge $e'' = (u, v_i)$, $e'' \notin E(W)$, that points towards the tour. Note that if G contains loops, it might happen that $u = v_i = w$ and hence $e' = e''$.

In both cases W is not a longest walk, as we can construct a longer one using either e' or e''. Either we extend W at the end using e' to get the walk

$$W' = v_i, e_i, \dots, v_{i-1}, e_{i-1}, v_i, e', w,$$

or we extend W at the beginning by starting from u along the edge e'' to construct

$$W'' = u, e'', v_i, e_i, v_{i+1}, \dots, e_k, v_1$$

This contradicts the maximality of W.

Exercise 4 (*written homework, 4 points*)

We start by giving an example for sequences of length 3. In this case "0001011100" is a minimal length string that contains all such sequences.

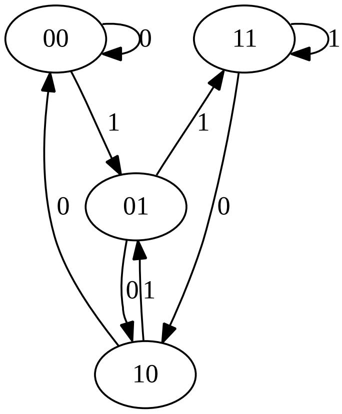
0	0	0	1	0	1	1	1	0	0
0	0	0							
	0	0	1						
		0	1	0					
			1	0	1				
				0	1	1			
					1	1	1		
						1	1	0	
							1	0	0

Claim: The shortest bitstring that contains all bitstrings of length k has length $2^k + k - 1$.

Proof: The length of such a sequence must be minimal: In a sequence of length n, there are $n - (k - 1)$ possible positions for a length k substring. To have all bitstrings of length k, we need a string of length $2^k + k - 1$.

We proceed to show how the existence of these strings. As we are supposed to use exercise 3, we start by constructing a graph in which a Eulerian tour corresponds to a minimal length bitstring as constructed above for $k = 3$.

Let G_k be a directed graph with $V = \{(b_1, b_2, \dots, b_k) | b_i \in \{0, 1\}\}$ and $E = \{((b_1, b_2, \dots, b_k), (b_2, b_3, \dots, b_{k+1})$ $\{0, 1\}\}$. Let an edge $e = (u, v)$ be labeled with the last bit of v. For $k = 2$ we get the following graph:



To use exercise 3 for concluding that G_k is indeed Eulerian for all k, we need to show that $\text{indeg}(v) = \text{outdeg}(v)$ for all $v \in V$ and that G_k is connected. As for a vertex $v = (b_1, b_2, \ldots, b_{k-1}, b_k)$ there are incoming edges from $(0, b_1, \ldots, b_{k-1})$ and $(1, b_1, \ldots, b_{k-1})$ and outgoing edges to $(b_2, \ldots, b_k, 0)$ and $(b_2, \ldots, b_k, 1)$, the restriction on the degrees is satisfied. The graph is also connected, as for each node $v = (b_1, \ldots, b_k)$ there is a path to $(0, 0, \ldots, 0)$. Therefore G is Eulerian.

Note that $|E(G_k)| = 2^{k+1}$.

We show how to construct a string as required by the exercise from a Eulerian tour on G_k.

Let $v_1, e_1, v_2, \ldots e_{2^{k+1}} v_1$ be a Eulerian tour in G_k. Then $v_1, e_1, e_2, \ldots, e_{2^{k+1}}$ is a string (formed from the labels) that contains all bitstrings of length $k+1$. Because of how we defined the edges, whenever an edge $e = ((a_1, \ldots, a_k), (b_1, \ldots, b_k))$ occurs in the sequence, the sequence contains the bitstring $(a_1, a_2, \ldots, a_k, b_k)$. As there are 2^{k+1} edges and no such bitstring occurs twice, all bitstrings of length $k+1$ are contained in the sequence.