Algorithmic Discrete Mathematics 3. Exercise Sheet

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Mathematics
PD Dr. Ulf Lorenz
Dipl.-Math. David Meffert

SS 2012
23. and 24. May 2012
Version of July 15, 2012

Groupwork

Exercise G1
Let the algorithm CLIQUE be defined by:

input : A graph G and a natural number k.

output : 'yes', if G contains a clique of cardinality k. Otherwise 'no'.

Let the algorithm IDEPENDENT SET (IS) be defined by:

input : A graph G and a natural number k.

output : 'yes', if G contains an independent set consisting of k vertices. Otherwise 'no'.

Show that CLIQUE $\leq_P$ IS.

Solution: First lets determine the polynomial function f which transforms the inputs $G = (V, E)$ and $k \in \mathbb{N}$ of CLIQUE to suitable inputs for IS. It is given by $f(\langle G, k\rangle) = \langle G', k\rangle$ with $G' = (V, E')$. The set E' is defined by $(v, w) \in E' \iff (v, w) \notin E$. G' is called the complementary graph (see H8).
claim: G contains a clique of cardinality $k \iff G'$ contains an independent set of cardinality k.
proof:

- $\Rightarrow$ Let $v_1, \ldots, v_k$ be a clique of cardinality k. Hence for every two different vertices v_i for $i \in \{1, \ldots, k\}$ they are directly connected by an edge. By definition this means they are not directly connected by an edge in G'. So the vertices $v_1, \ldots, v_k$ form an idenpendent set in G'.
- $\Leftarrow$ Assume that G contains no clique of cardinality k. Take an arbitary subset $S \subset V$ consisting of k elements $v_1, \ldots, v_k$. Because by assumption it is not a clique of G there exist $i, j \in \{1, \ldots, k\}$ with $i \neq j$ and $(v_i, v_j) \notin E$. By definition this means $(v_i, v_j) \in E'$ and for this reason $S = v_1, \ldots, v_k$ can't form an independent set in G'. Because S was chosen arbitrarily the second implication is proven.

Exercise G2 (Bipartite graphs)
Prove that a graph (V, E) is bipartite iff it contains no cycles of odd length.

Solution:

- $\Rightarrow$ Let G be bipartite with induced partion $V = S \dot\cup T$. Let $(v_1, v_2, \ldots v_n, v_1)$ be a cycle in G and w.l.o.g. let $v_1 \in S$. Since there are no edges inside the sets S and T, we get $v_{2k} \in T$ and $v_{2_{k+1}} \in S$ for $k \in \{1, \ldots \lfloor \frac{n}{2} \rfloor\}$. So it is necessary that n is even for v_n being an element of T. Hence there are no cycles of odd length.
- $\Leftarrow$ Let G be a graph without odd cycles. Without loss of generality we ca assume that G is connected. Otherwise do the same argumentation for each connected component of the graph. For a vertice $x_0 \in V$ let S be the set of all vertices in V with even distance[1] to the vertex x_0. Additionally we call its complement $T := V \setminus S$. We want to show that there are no edges within S and T. So assume there is an edge $(x, y) \in E$, such that $x, y \in S$. Let W_x and W_y be shortest paths from x_0 to x and to y respectively. By definition of S the paths W_x and W_y both have even length. Now let z be the last common vertice of the paths W_x and W_y (both beginning in x_0) and denote by W'_x and W'_y the remaining part of the paths from z to x and y respectively. Now define the cycle $x - W'_x \to z - W'_y \to y - (x, y) \to x$. Since W'_x and W'_y both have either odd or even length the given cycle has odd length. This contradicts G having

[1] The distance of two vertices x, y in a graph G is defined as the number of edges used in a shortest path from x to y.

no cycles of odd length, hence our assumption was wrong and for this reason there are no edges in the set S. The statement for the set T can be shown the same way. So we have $V = S \dot\cup T$ with suitable sets V and T. Hence the graph G is bipartite.

Exercise G3 (Eulerian graphs)
A path in a Graph $G = (V, E)$ is called *Eulerian path*, if it contains every edge $e \in E$ exactly once. An Eulerian path which is a cycle is called *Eulerian cycle*. A graph is called *eulerian* if it contains an Eulerian cycle.

(a) Which of the given graphs in **Figure 1** are Eulerian graphs?

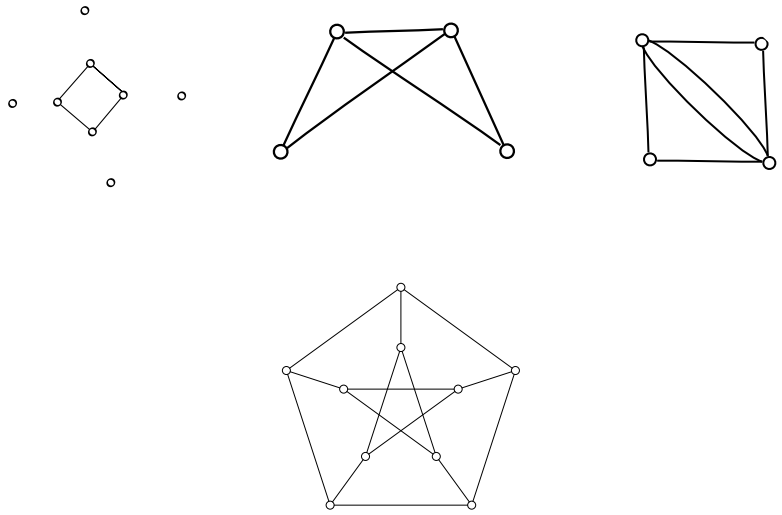

Figure 1: Eulerian graphs?

(b) Now let G be a connected graph. Name necessary conditions for G being eulerian.

(c) Are these conditions sufficient, too?

Solution:

(a) The solution is yes,no,yes,no. The best way to prove the nonexistence of an Eulerian cycle is by using one of the criterias of part (b).

(b) We look at the following two conditions and prove that they are necessary. Later we will show they are even equivalent.

i. E can be decomposed into cycles having disjoint edges.

ii. All vertices in V have an even degree.

Lets prove necessity of the conditions. So let $G = (V, E)$ be a graph and $T = [e_1, \ldots, e_n]$ be an Eulerian cycle in G.

We prove the decomposition into cycles first. So we take an arbitrary vertice $v \in V$ and follow the path given by T till we reach the vertex v again. If the cycle C constructed that way is the whole Eulerian cycle we are finished. Otherwise delete all the edges of C in the set E and delete all vertices in V which got isolated by this deletion process. The resulting graph $V' = (G', E')$ is eulerian again. So we repeat this procedure till the resulting graph is empty. By construction we get a decomposition of cycles having disjoint edges.

Now we can conclude the second condition (all vertices have even degree). Let $v \in V$ arbitrary. By doing the same procedure as for the cycle decomposition and always choosing v as the starting node we get exactly 2 new incident edges of v in every step till v becomes an isolated node. By construction they are all disjoint, hence $d(v) = 2k_v$ with k_v is the number of disjoint cycles containing v. Since $v \in V$ was arbitrarily chosen we are done.

(c) Yes the given conditions in part (b) are also sufficient. We we will prove this the following way. First we prove that a graph can be decomposed into cycles with disjoint edges, iff all vertices have even degree. After that we just have to prove the first condition.

So lets start with the equivalence. The first implication was already shown in part (b). So let G be a graph where all vertices have even degree. We use induction over the number of edges k in G to prove the decomposition.

Base case: For $k = 0$ the graph is trivially decomposed into cycles with disjoint edges.
Induction hypothesis: Assume that every graph with at most k edges can be decomposed into cycles with disjoint

edges.

Inductive step: Let G be a graph with $k+1$ edges, $v \in V$ an arbitrary vertex and $v_1 \in V$ with $(v, v_1) \in E$. Since $d(v_1) \geq 2$ there exists a vertex $v_2 \neq v$ with $(v_1, v_2) \in E$. By iterating this process, because G is a finite graph, we will come to a vertex v_l we already visited. So we get a cycle $C = [v_l, v_{l+1}, \ldots, v_l]$ in G. If C contains all edges in E we are done. Else we delete all these edges contained in the cycle and all vertice getting isolated by this deletion process (those with $d(v) = 2$). The resulting graph $G' = (V', E')$ contains less than k edges and by the induction hypothesis can be decomposed to a decomposition $\mathscr{E}$ of cycles with disjoint edges. The set $\mathscr{E} \cup \{C\}$ yields a decomposition of cycles with disjoint edges for the whole graph G.

Now we construct an Eulerian cycle out of the cycle decomposition with disjoint edges. Again we use induction over the number m of disjoint cycles.

Base case: For $m = 1$ the only cycle we have is of course an Eulerian cycle.
Induction hypothesis: Assume that every connected graph that can be decomposed into at most m disjoint cycles has an Eulerian cycle.
Inductive step: Let $\mathscr{E}$ be a decomposition into $m+1$ disjoint cycles. Choose a cycle $C \in \mathscr{E}$ and delete all edges in C from E and all vertices getting isolated by the deletion process. Since all connected components of the resulting graph $G' = (V', E')$ have a decomposition into at most m disjoint cycles, they all have an Eulerian cycle by using the induction hypothesis. With these cycles we construct an Eulerian cycle for G the following way. Start with an arbitrary vertex $v \in C$ and follow the cycle till the first node $v_1 \in V'$. Now follow the Eulerian cycle induced by connected component (V_1, K_1) containing v_1. Now keep on following the cycle C till the first node $v_1 \in V' \setminus \{V_1\}$.We repeat this procedure till we went through the whole cycle C and get to the starting vertex v again.(after going through all the Eulerian cycles of the connected components) Since the graph G is connnected we went through all the Eulerian cycles which are induced by the connected components of G' and the cycle C and for this reason have contructed an Eulerian cycle for G.

Exercise G4 (Primes and the class $\mathcal{NP}$)
The class of problems whose complement is in $\mathcal{NP}$ is called co-$\mathcal{NP}$. For the following exercise assume that the coding length of a natural number n is given by $\langle n \rangle = \lfloor \log_2 n \rfloor + 1$.[2]

(a) Show that the problem PRIMES, to determine if a given natural number is prime, is a co-$\mathcal{NP}$ problem.

(b) Why would it be much harder to show that this problem is also in the class $\mathcal{NP}$?

(c) Prove PRIMES $\in \mathcal{NP}$ under the assumption $\langle n \rangle = n$.

Solution:

(a) We have to prove that the problem to determine if a given number p is not a prime, is an $\mathcal{NP}$-problem.

A suitable object (certificate) for p not being a prime are two natural number a, b with $1 < a < b < p$ with $a \cdot b = p$. With this object one can easily check if the given answer 'yes' (p is not a prime) was correct. The algorithm to do this gets the inputs p and (a, b) and decides if (a, b) is a suitable certificate for p by checking if $a \cdot b = p$ and $1 < a < b < p$. Because of $a, b < p$ the binary representation of a, b is shorter than the one of p. So the runtime l for the mutiplication of a and b is $\max\{\langle a \rangle \langle b \rangle\} \cdot 1 \leq \langle p \rangle$. The runtime for the comparing of the numbers $a \cdot b$ and p is $\langle p \rangle$. Same holds for checking $1 < a < b < p$. So the algorithm is linear in $\langle p \rangle$, so in particular polynomial $\langle p \rangle$.

(b) This is much harder because one would need a certificate (probably a number or a set of numbers) which can be used (in polynomial time) to prove that p is prime. The coding length should also be polynomial in $\langle p \rangle$. It is not clear how such a certificate should look like. The set of all numbers smaller than p is not a good choice because the coding length of this set is equal to $\sum_{i=1}^{p-1} (\lfloor \log_2 i \rfloor + 1)$ which is not polynomial in $\langle p \rangle = \lfloor \log_2 p \rfloor + 1$.

Indeed such certificates do exist. The first one was found in 1975 by the australian mathematician Vaughan Pratt (born 1944). It consists of a number x which is coprime to p and has some other properties.

(c) Under this assumption a polynomial algorithm with a polynomial certificate is for example given by all natural numbers smaller than p (or $\sqrt{p}$) and checking if the division yields a natural number or not. Under the assumption $\langle p \rangle = p$ this certifate has polynomial coding length and the checking algorithm uses p divisions and for this reason has a runtime of $p \cdot \langle p \rangle = p^2$ which is of course polynomial in $\langle p \rangle = p$.

In fact solving this problem is also possible with the usual coding length (binary representation), but is was a open question for quite a long time. An algorithm was found in 2002 by the Indian mathematicians Manindra Agrawel, Neeraj Kayal and Nitin Saxena.

[2] This is the common binary representation of a natural number n. Hence you need $\lfloor \log_2 n \rfloor + 1$ digits.

Homework

Exercise H7 (Trees) (10 points)

Let $G = (V, E)$ be a graph with $n \geq 2$ vertices. Proof that the following statements are equivalent:

(a) G is a tree.

(b) G is connected and contains $n - 1$ edges.

(c) G contains $n - 1$ edges but no cycles.

(d) G is minimally connected. That means for every edge $e \in E$ the graph $G \setminus \{e\} = (V, E \setminus \{e\})$ is not connected.

(e) G contains no cycles and adding one edge generates exactly one cycle.

(f) For every two nodes $u, v \in V$ there is exactly one $[u, v]$-path in G.

Solution: There are several ways to prove the equivalences by proving different implications. We present one way to it.

$(a) \Rightarrow (b)$ and (c): We prove this by induction over the number of vertices n. For $n = 1$ we have only one vertex and no edges (otherwise there would be a cycle). So the graph has $n - 1 = 0$ edges and is of course connected and has no cycles. Now assume the statement we want to prove is right for all trees with a most n vertices. Let G be a tree with $n + 1$ vertices. Since $n + 1 > 1$ it is not hard to see, that we have at least one leaf v in the graph G. (by looking a simple path of maximal length in the tree) Delete this vertex v and its one incident edge. Since v was a leaf the resulting graph $G' = (V', E')$ is still connnected and has no cycles because G had not cycles in advance.(so G' is a tree with n vertices) By assumption G' has $n - 2$ edges is connected and has no cycles. Adding the deleted leaf with its edge to graph again (to get back G) we see T has $n - 1$ edges, is connected and has no cycles.

$(a) \Rightarrow (d)$: Let $e = (u, v)$ be an arbitrary edge in G. Assume $G \setminus \{e\}$ is connected. Then there exists a path from u to v which does not use the edge e. By adding e to this path we get a cycle which contradicts G being a tree.

$(a) \Rightarrow (f)$: Let $u, v \in V$. Since G is connected there is a least one path $p = (v = v_1, \ldots, v_k = u)$ in G. Assume there is another path $q = (v = q_1, \ldots, q_l = u)$. Let i be the smalllest index with $v_i \neq q_i$ and j the smallest with $v_j = q_j$ and $j > i$. Notice we have $i > 1$ ($v_1 = q_1$) and j exists since $v_k = q_l$. Taking the path $c = (v_{i-1}, v_i, \ldots, v_j = q_j, q_{j-1}, \ldots w_{i-1} = v_{i-1})$ we get a cycle which is a contradiction to G being a tree.

$(a) \Rightarrow (e)$: Assume we want to add the edge $e = (v, u)$ for $v, u \in V$. Because G was already connected there is a path from u to v not using e. Be adding the edge e to this path we get a cycle. Since the path from u to v was unique be part (f) this is the only cycle we get by adding e.

$(b) \Rightarrow (a)$: Assume we have a cycle C in the graph G. If we delete an arbitrary edge $e \in C$ the graph is still connected. This way we can remove all the cycles from G an get a connected graph G' without cycles. So G' is a tree and for this reason has to have $n - 1$ edges. This is a contradiction to G already having $n - 1$ edges, because we deleted at least one edge. So G already had no cycles.

$(c) \Rightarrow (a)$: Assume G is not connected and consists of the connected components $K_1, \ldots, K_r$. Then we can connect an arbitrary vertex $v_1 \in K_1$ with another vertice $v_2 \in K_2$ without generating a cycle. We do the same thing for all the other connected components and get a graph G' which is a connected and without cycles, hence a tree. So G' has $n - 1$ edges which is a contradiction to G already having $n - 1$ edges, because we at least added one edge.

$(d) \Rightarrow (a)$: We have to show that G has no cycles. Assume there is a cycle C in G. Then we can delete an arbitrary edge $e \in C$ and the resulting graph G' is still connected. This contradicts G being minimally connected.

$(e) \Rightarrow (a)$: We have to prove that G is connected. Assume it is not. Then there at least two connected components K_1 and K_2 of G and we can add an arbitrary edge between any vertices $v_1 \in K_1$ and $v_2 \in K_2$ without getting a cycle. This is a contradiction.

$(f) \Rightarrow (a)$: Since there is path from u to v for arbitrary vertices $u, v \in V$ the graph G is connected. If G would contain any cycle C there would be at least two different paths from u to v for arbitrary vertices $u, v \in C$. So G contains no cycles.

Exercise H8 (The complement graph) (10 points)

The complement graph $\overline{G}$ of $G = (V, E)$ is the graph were two vertices are adjacent iff they are not adjacent in G. So formally speaking we have $\overline{G} := (V, \binom{[n]}{2} \setminus E)$ with $\binom{[n]}{2} \setminus E := \{(i, j) \mid i \neq j \in \{1, \ldots, n\} (i, j) \notin E\}$.

Let G be an undirected graph. Prove that G or $\overline{G}$ is connected.

Solution: Lets assume G is not connected. (otherwise we are done). Without loss of generality one can assume that G has exactly two connected components K_1 and K_2. The general case can be proved by induction over the number of components.

Let V_1 consist of the vertices of K_1 and V_2 of the vertices of K_2. In the worst case $\overline{G}$ is a complete bipartite graph with partitions V_1 and V_2. For $v \in V_1$ and $w \in V_2$ the edge (v,w) is a $[v,w]$-path in $\overline{G}$. Let $v,w \in V_1$. Taking an arbitrary $u \in V_2$ the path (v,u,w) is a $[v,w]$-path. For $v,w \in V_2$ it works the same way. So $\overline{G}$ is connected.

Exercise H9 (10 points)

Prove SUBSETSUM $\leq_p$ PARTITION.

problem : SUBSETSUM

input : $a_1,...,a_n, b \in \mathbb{N}$

output : $T \subseteq \{1,...,n\}$ with $\sum_{k\in T} a_k = b$

problem : PARTITION

input : $a_1,...,a_l \in \mathbb{N}$

output : $T \subseteq \{1,...,l\}$ with $\sum_{k\in T} a_k = \sum_{k\notin T} a_k$

Solution: First lets determine the polynomial function f which transforms the inputs $a_1,\dots,a_n \in \mathbb{N}$ and $b \in \mathbb{N}$ of SUBSETSUM to suitable inputs for PARTITION. It is given by $f(\langle a_1,\dots,a_n,b\rangle) = \langle a_1,\dots,a_n,a_{n+1},a_{n+2}\rangle$ with $a_{n+1} = M - b + 1$ and $a_{n+2} = b+1$. The number M is defined by $M := \sum_{k=1}^{n} a_k$. This transformation can be done in polynomial time.

claim: $T \subseteq \{1,\dots,n\}$ is a solution for SUBSETSUM with inputs $a_1,\dots,a_n,b \in \mathbb{N} \iff K := T \cup \{n+1\} \subseteq \{1,\dots,n+2\}$ is a solution for PARTITION with inputs $a_1,\dots a_{n+2}$. (same argument can be done for $T \cup \{n+2\}$ but by construction a solution can't contain $n+1$ and $n+2$)

proof:

- $\Rightarrow$ Let $K \subseteq \{1,\dots,n+2\}$ be a solution for PARTITION of the form $K = T \cup \{n+1\}$ with inputs $a_1,\dots a_{n+2}$. We get

$$\begin{aligned}
&\textstyle\sum_{k\in K} a_k = \sum_{k\notin K} a_k \\
\Leftrightarrow\quad &\textstyle\sum_{k\in K\setminus\{n+1\}} a_k + M - b + 1 = \sum_{k\notin K\setminus\{n+2\}} a_k + b + 1 \\
\Leftrightarrow\quad &\textstyle\sum_{k\in K\setminus\{n+1\}} a_k + \sum_{k=1}^{n} a_k - \sum_{k\notin K\setminus\{n+2\}} a_k = 2b \\
\Leftrightarrow\quad &\textstyle\sum_{k\in K\setminus\{n+1\}} a_k + \sum_{k\in K\setminus\{n+1\}} a_k = 2b \\
\Leftrightarrow\quad &\textstyle 2\sum_{k\in K\setminus\{n+1\}} a_k = 2b.
\end{aligned}$$

  Hence $T = K \setminus \{n+1\}$ is a solution for SUBSETSUM.

- $\Leftarrow$ Let $T \subseteq \{1,\dots,a_n\}$ be a solution for SUBSETSUM. Then $K = T \cup \{n+1\}$ is a solution for PARTITION because of

$$\begin{aligned}
\sum_{k\in K} a_k &= M - b + 1 + \sum_{k\in T} a_k = 1 + \sum_{k=1}^{n} a_k \\
&= 1 + \sum_{k\in T} a_k + \sum_{k\notin T} a_k = 1 + b + \sum_{k\notin T} a_k \\
&= \sum_{k\notin K} a_k.
\end{aligned}$$

  Notice that $T \subseteq \{1,\dots,n\}$ and $K \subseteq \{1,\dots,n+2\}$.