COL 351

TUTORIAL SHEET 12

1. The directed Hamiltonian Cycle Problem is as follows: given a directed graph G, is there a cycle which contains all the vertices ? Suppose you have a polynomial time algorithm for this problem. Show that you can also find such a cycle (if it exists) in polynomial time.

   Solution: Suppose G is Hamiltonian. Let C be any Hamiltonian cycle in G. Then, if we remove any edge not in C, the resulting graph will still be Hamiltonian. Thus, we get the following algorithm (let $\mathcal{A}$ denote the algorithm which given a graph, decides whether it is Hamiltonian or not): first run $\mathcal{A}$ on G to check if G is Hamiltonian or not. Assume G is Hamiltonian. While G has more than n edges, find an edge e in G such that $\mathcal{A}(G - e)$ returns true. As we argued above, there must exist such an edge – so we can try each edge in G and see if $\mathcal{A}(G - e)$ is true or not. Let e be such an edge. Then, we remove e, and repeat this process. Finally, when G has only n edges, these must form a Hamiltonian cycle.

2. The undirected Hamiltonian Cycle Problem can be defined similarly as above. The undirected Hamiltonian Path problem is as follows: given an undirected graph G, is there a path which contains all the vertices ? Show that the undirected Hamiltonian path is polynomial time reducible to the undirected Hamiltonian Cycle problem.

   Solution: Let $\mathcal{I}$ be an input to the Hamiltonian path problem. Note that $\mathcal{I}$ consists of an undirected graph G. We need to produce a graph G' such that G has a Hamiltonian path if and only if G' has a Hamiltonian cycle. We proceed as follows: add a new vertex v to the graph G and add edges between v and every vertex in G – call this graph G'. Now if P is a Hamiltonian path in G starting at vertex s and ending at t, then v, s, P, t, v is a Hamiltonian cycle in G'. Conversely, if C is a Hamiltonian cycle in G', then removing the vertex v from C gives a Hamiltonian path in G.

3. [**KT-Chapter8**] Consider a set $A = \{a_1, ..., a_n\}$ and a collection $B_1, B_2, ..., B_m$ of subsets of A. (That is, $B_i \subseteq A$ for each i.) We say that a set $H \subseteq A$ is a hitting set for the collection $B_1, B_2, ..., B_m$ if H contains at least one element from each B_i ? that is, if $H \cap B_i$ is not empty for each i. We now define the Hitting Set problem as follows. We are given a set $A = \{a_1, ..., a_n\}$, a collection $B_1, B_2, ..., B_m$ of subsets of A, and a number k. We are asked: is there a hitting set $H \subseteq A$ for $B_1, B_2, ..., B_m$ so that the size of H is at most k? Prove that Hitting Set is NP-complete.

   Solution: It is easy to show that Hitting Set is in NP. A solution just needs to exhibit the set H – one can easily verify in polynomial time whether H is of size k and intersects each of the sets $B_1, \ldots, B_m$.

   We reduce from vertex cover. Consider an instance of the vertex cover problem – graph $G = (V, E)$ and a positive integer k. We map it to an instance of the hitting set problem as follows. The set A is the set of vertices V. For every edge $e \in E$, we have a set S_e consisting of the two end-points of e. It is easy to see that a set of vertices S is a vertex cover of G iff the corresponding elements form a hitting set in the hitting set instance.

4. [**KT-Chapter8**] You have a set of friends F whom you're considering to invite, and you're aware of a set of k project groups, $S_1, \ldots, S_k$, among these friends (these sets need not be disjoint). The problem is to decide if there is a set of n of your friends whom you could invite so that not all members of any one group are invited. Prove that this problem is NP-complete.

   Solution: It is in NP because a solution can just exhibit such a set of friends, and the verifying algorithm has to just check that for each group this set does not contain all the elements from that set – a task which can be easily done in polynomial time.

   To prove NP-completeness, we reduce from independent set. Consider a graph G and a number k. We map it to an instance of this problem as follows: set of friends F correspond to the set of vertices in G. For every edge $e = (u, v)$ in G, we have a set S_e consisting of $\{u, v\}$ only. It is easy to check that G has an independent set of size k iff the instance of this problem has a set of friends of size k with the desired property.

5. [**KT-Chapter9**] Give an algorithm for the Hamiltonian path problem in a directed graph whose running time is $O(2^n p(n))$, where $p(n)$ is a polynomial in n (here, n denotes the number of vertices in the graph).

   Solution: This is done by dynamic programming. For every subset S of vertices, let $G[S]$ denote the subgraph of G induced by S, i.e., those edges of G which have both end-points in S. For every set of vertices S and vertices $v \in S$, we have a table entry $T[S, v]$ which is supposed to store a Hamiltonian path in $G[S]$ which starts with v (if there exists such a path, otherwise it stores NO). Now the recursive definition of T. Let N_v be the set of neighbors of v in the set S. Then $T[S, v]$ is false if $T[S - v, w]$ is false for all $w \in N_v$. Otherwise suppose $T[S - v, w]$ is a path P. Then $T[S, v]$ stores the path v followed by P. It is easy to check that the time taken by the algorithm is $O^(2^n n^2)$.

6. [**KT-Chapter8**] Consider the following problem. You are given positive integers $x_1, \ldots, x_n$, and numbers k and B. You want to know whether it is possible to partition the numbers $\{x_i\}$ into k sets $S_1, \ldots, S_k$ so that the squared sums of the sets add up to at most B:

   $$\sum_{i=1}^{k} \left(\sum_{x_j \in S_i} x_j \right)^2 \leq B.$$

   Show that this problem is NP-complete.

   Solution: A solution just needs to exhibit the partition. Since addition and multiplication take polynomial time, we can easily verify a solution in polynomial time. We reduce PARTITION to this problem. Recall that an input to partition consists of a set of positive numbers $x_1, \ldots, x_n$, and we need to check if they can be divided into two parts, each of which has the same sum. We map this to our problem as follows:

the set of numbers is $x_1, \ldots, x_n$ again. The parameter $k = 2$, and $B = \Sigma^2/2$, where Σ is the sum of these n numbers. Now suppose there were a partition of $x_1, \ldots, x_n$ into two sets A and B of equal sum. Then, the same partition in our problem will have the sum $\Sigma^2/2$.

Now we show the converse. Suppose it is possible to partition the numbers into parts A and B such that $\Sigma_A^2 + \Sigma_B^2 \leq \Sigma^2/2$, where Σ_A denotes the total sum of numbers in A (and similarly for Σ_B). But note that $\Sigma_A + \Sigma_B = \Sigma$. It is an easy exercise to show that $\Sigma_A^2 + \Sigma_B^2 \geq \Sigma^2/2$ with equality if and only if $\Sigma_A = \Sigma_B$. Thus, $\Sigma_A = \Sigma_B$.

7. [**KT-Chapter8**] Given an undirected graph $G = (V, E)$, a feedback set is a set $X \subseteq V$ with the property that $G - X$ has no cycles. The undirected feedback set problem asks: given G and k, does G contain a feedback set of size at most k? Prove that the undirected feedback set problem is NP-complete.

   Solution: Easy to check that the problem is in NP: a solution needs to show a set of vertices S. The verifying algorithm first removes S, and then checks that the resulting graph does not have a cycle (can be done by any graph traversal algorithm).

   We reduce from vertex cover. Let G, k be an instance of vertex cover. We produce a graph $G' = (V', E')$ from $G = (V, E)$ as follows: for every vertex v in G, we have a vertex v in G' as well (G' will have some more vertices). For an edge $e = (u, v)$ in G, we create a new vertex v_e in G', and add edges $(u, v_e), (v_e, v), (u, v)$ in G'. Now, suppose G have a vertex cover of size k. Let these vertices be S. We claim that S will be a feedback set in G'. Indeed, any cycle in G' must contain a pair u, v, where (u, v) is an edge in G. Since S contains at least one of u and v, S must intersect this cycle. Thus after removing S from G', we will not have any cycle.

   Conversely, consider a feedback set S of size k in G'. First we claim that S need not contain any of the vertices v_e – indeed, if $e = (x, y)$, then any cycle containing v_e must also contain x. Thus we can replace v_e by x. Finally, any feedback set in S' must contain a vertex from the cycle x, y, v_e in G', where $e = (x, y)$ is an edge in E. We just argued that S must contain either x or y. Thus, S is a vertex cover in G.