Lab13-Solution

CS363-Computability Theory, Xiaofeng Gao, Spring 2016

1. Find the certificate and certifier for the decision version of the following problems.

   (a) Clique: Given an undirected graph, find a subset S that there is an edge connecting every pair of nodes in S with maximum nodes.

   (b) Metric k-center: Given n cities with specified distances for each pair of cities as d_{ij}, one wants to build k warehouses in different cities and minimize the maximum distance of a city to a warehouse.

   (c) Set Packing: Given a set U of n elements, a collection $S_1, \cdots, S_m$ of subsets of U, find the maximum subsets such that no two of them intersect.

   (d) minimum k-cut: Given a weighted graph $G = (V, E)$, we want to find a minimum weighted set of edges whose removal would partition the graph to k connected components.

   Solution.

   (a) Decision version: Given an undirected graph, does there exists a subset S that there is an edge connecting every pair of nodes in S with k nodes.

   - Certificate: A subgraph $S = (V, E)$ of the original graph with $|V| = k$
   - Certifier: Check the following conditions: S is a subgraph $S = (V, E)$ of the original graph with $|V| = k$; Check if $\forall u, v \in V$ have a corresponding edge $\{u, v\} \in E$. If both conditions hold, return true; otherwise false.

   (b) Decision version: Given n cities with specified distances for each pair of cities as d_{ij}, is it possible to build k warehouses in different cities and the maximum distance of a city to a warehouse is less than or equal to D.

   - Certificate: A subset S of the n cities and $|S| = k$
   - Certifier: Check the following conditions: S is a subset of the n cities with cardinality k; Find the maximum distance d_{max} of a city to S and check $d_{max} \leq D$. If both conditions hold, return true; otherwise false.

   (c) Decision version: Given a set U of n elements, a collection $S_1, \cdots, S_m$ of subsets of U, is there a subset C of $\{S_1, \cdots, S_m\}$ such that $|C| \geq k$ and S_i do not intersect with S_j for any $S_i, S_j \in C \land S_i \neq S_j$.

   - Certificate: A subset C of $\{S_1, \cdots, S_m\}$.
   - Certifier: Check the following conditions: C is a subset of $\{S_1, \cdots, S_m\}$; S_i do not intersect with S_j for any $S_i, S_j \in C \land S_i \neq S_j$; $|C| \geq k$. If all conditions hold, return true; otherwise false.

   (d) Decision version: Given a weighted graph $G = (V, E)$, does there exists a m weighted set of edges whose removal would partition the graph to k connected components.

   - Certificate: A subset S of edges in E.
   - Certifier: Check the following conditions: If the total weight of the edges in S is m; Check if the remaining graph contains exactly k connected components (There are many polynomial algorithms that can do it, like DFS). If both conditions hold, return true; otherwise false.

$\square$

2. The knapsack problem is a well-known optimization problem. Given a set of n items, each item i with a weight w_i and a value v_i, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit W and the total value is as large as possible.

   Prove that the knapsack problem is NP-complete. (Hint: One solution is reducing the Subset Sum problem to it.)

   Solution.

   - Firstly, we show the problem is NP. We can find the decision version, certificate and certifier as following:
     - Given a set of n items, each item i with a weight w_i and a value v_i, is there a collection C with total value equal or larger than V while total weight is less than or equal to a given limit W.
     - Certificate: A collection C of items.
     - Certifier: Check the following conditions: The total value of items in C is equal or larger than V; The total weight of items in C is equal or less than W; If all conditions hold, return true; otherwise false.
   - Then we show how we construct a instance of knapsack problem from a instance from the subset sum problem. Consider a subset sum problem with set $S = \{e_1, \ldots, e_n\}$ and the target T. Assume $\forall i(e_i \leq e_m)$.
     - For each element $e_i \in S$, we construct an two item: i_0 with $w_{i0} = v_{i0} = e_m * (2^i + 2^n) + e_i$, i_1 with $w_{i1} = v_{i1} = e_m * (2^i + 2^n)$.
     - We assign the weight threshold and the value threshold as $W = V = e_m * (n2^n + 2^{n-1} + \cdots + 1) + T$.

     Correctness Proof

     - $\Rightarrow$: If there is a subset S whose sum is exactly T. Then for each element e_i, if it is in S, we select i_{e_0}, else select i_{e_1}. Then we get a collection whose weight and value are exactly $e_m * (n2^n + 2^{n-1} + \cdots + 1) + T$.
     - $\Leftarrow$: If there is a collection whose weight and value are exactly $e_m * (n2^n + 2^{n-1} + \cdots + 1) + T$. The $e_m * (n2^n + 2^{n-1} + \cdots + 1)$ entry bounds that for each i, either i_{e_0} and i_{e_1} should be selected once. Then we can easily transformed the collection to a subset S.

   $\square$

3. We know that $\mathbf{P} \subseteq \mathbf{NP} \cap \textbf{co-NP}$. Please give an example that belongs to following set. If you can, briefly explain your reason. (Should be examples different from the course slides).

   (a) **Co-NP**.

   (b) **Co-NP** $\cap$ **NP-hard**.

   (c) **Co-NP** $\cap$ **NP**, but not known to be in **P**.

   Solution.

   (a)
   - The complement of Subset Zero problem: given a finite set of integers, does every non-empty subset have a non-zero sum?
   - The complement of Set Cover problem: for every combination of X subsets, is it impossible to cover all elements?

(b) If we can find such a problem that is both NP-hard and co-NP, then we can prove that NP=co-NP. However, the statement is still an open problem.

(c)
- Factoring : Given an integer, find its factors.
- Parity Games : Deciding which of the two players has a winning strategy in parity games.
- Stochastic Games : The problem of deciding which player has the greatest chance of winning a stochastic game
- Lattice Problems : The problems of approximating the shortest and closest vector in a lattice to within a factor of $\sqrt{n}$

$\square$