國立中央大學資訊工程學系 101 學年度第二學期博士班資格考試題紙

科目： **演 算 法** (Algorithms)

1. We have the following definitions and theorem related to NP-completeness.

   Definition 1. Let X_1 and X_2 be tow problems. X_1 reduces to X_2 (written as $X_1 \propto X_2$) if and only if X_1 can be solved in polynomial time, by using a polynomial time algorithm which solves X_2.

   Definition 2. A problem is said to be a P (resp., NP) problem if it can be solved in polynomial time by a deterministic (resp., non-deterministic) algorithm.

   Definition 3. $\mathcal{NP}$ (resp., $\mathcal{P}$) is the set of all NP (resp., P) problems.

   Definition 4. A problem X is NP-complete if $X \in \mathcal{NP}$ and every NP problem reduces to X.

   Cook's Theorem. $\mathcal{NP}=\mathcal{P}$ if and only if the satisfiability (SAT) problem is a P problem.
   (This implies that every NP problem reduces to SAT.)

   We assume that SAT $\propto$ B, B $\propto$ C, C $\propto$ D and $D \in \mathcal{NP}$. By the above definitions and theorem, show that D is NP-complete. (20%)

2. Given n objects with positive integer values $P_1, P_2, \ldots, P_n$ as profits, and positive integer values $W_1, W_2, \ldots, W_n$ as weights, and a knapsack with positive integer value M as its capacity, the 0/1 knapsack problem is to find a subset of selected objects such that the total profit sum of the selected objects is maximized and the total weight sum of the selected objects is not larger than M. Note that we assume ($P_i/W_i \geq P_{i+1}/W_{i+1}$) and we put a minus sing to each of $P_1, P_2, \ldots, P_n$ to turn the original 0/1 knapsack problem into a minimization problem to minimize the total profit sum of the selected objects. We can use the branch-and-bound strategy to solve the 0/1 knapsack problem by finding a lower bound and an upper bound for each solution tree node to terminate some tree nodes. **Explain how we can find the upper bound of a node** (7%)**, and how we can find the lower bound of a node** (7%), **and how we can terminate a node** (7%).

3. An approximation algorithm for the problem has a ratio bound of $\rho(n)$ if for any input size n, the result C of the solution produced by the approximation algorithm is within a factor of $\rho(n)$ of the result C^* of the optimal solution. **What is the definition of the ratio bound $\rho(n)$?** (9%)

4. Given a set of n sticks of various lengths x_1, x_2, … , x_n, is it possible to join them end-to-end to form a square? For example if you are given 5 sticks of lengths 10, 20, 30, 40, 50, you should answer “no”; however, if the lengths are 10, 20, 30, 30, 30, the answer is “yes”. (a) Show that this problem is NP-complete. (10%) (b) If the sum of the lengths is small enough, it can be solved efficiently by a dynamic programming algorithm. Derive such an algorithm. (15%)

5. For an undirected graph G=(V, E) and a vertex v in V, let $G\backslash v$ denote the subgraph of G obtained by removing v and all the edges incident to v from G. If G is connected, then $G\backslash v$ can be connected or disconnected. (a) Prove that for any connected graph G, we can always find a vertex v in G such that $G\backslash v$ is connected. (10%) (b) Design an $O(|V|)$ time algorithm to find such a vertex. (15%)