國立臺灣師範大學九十六學年度碩士班考試入學招生試題

軟體基礎 科試題（資訊工程學系用，本試題共 4 頁）

注意：1.依次序作答，只要標明題號，不必抄題。
2.答案必須寫在答案卷上，否則不予計分。

1. Answer the following questions with supporting explanations if needed.

(a) Obtain an addressing formula for the element $A[i_1][i_2],\cdots,[i_n]$ in an array declared as $A[l_1..u_1][l_2..u_2],\cdots,[l_n..u_n]$. Assume a column major representation of the array with one word per element and α the address of $A[l_1][l_2],\cdots,[l_n]$. (5 分)

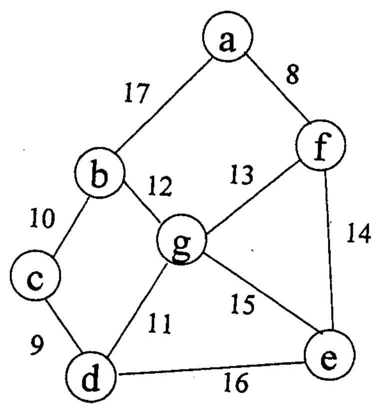
(b) Write out the postfix form of the expression:

! (A && ! ((B < C) || (C>D))) || (B < E) . (5 分)

(c) Use the link-list representation to represent the sparse matrix:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (5 分)$$

2. Given a weighted, undirected graph G shown as below:



(a) Write out the costs of the edges selected in order when using Kruskal's algorithm to construct the minimum-cost spanning tree of G. (5 分)

07-3

(b) Write out the costs of the edges selected in order when using Prim's algorithm to construct the minimum-cost spanning tree of G and starting from vertex a. (5 分)

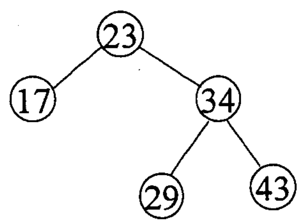
(c) Compare Kruskal's algorithm and Prim's algorithm. (5 分)

3. Suppose we have the preorder sequence " j i g c b f h e a d " and the inorder sequence " c g b i f j a e h d " of the same binary tree T.

(a) Draw the binary tree T. (5 分)

(b) Write out the level-order traversal for T. (5 分)

4. Given an AVL (Adelson-Velskii and Landis) tree T shown as below:



(a) Draw the corresponding new AVL tree T after 52, 25 and 48 are inserted. (5 分)

(b) Draw the corresponding new AVL tree T after 26, 7 and 28 are further inserted based on the result of (i). (5 分)

5. What is the major difference between $\theta(n^3)$ and $\omega(n^3)$? (5 分)

6. What is the major difference between P problems and NP problems? (5 分)

7. Given a set S containing n real numbers.

(a) Please design an O(n) time algorithm to find a number that is *not* in the set. (5 分)

(b) Prove that $\Omega(n)$ is a lower bound on the number of steps required to solve the problem. (5 分)

8. A thief breaks into a jewelry store carrying a knapsack. The knapsack will break if the total weight of the items stolen exceeds some maximum weight *W*. Each item has a value and a weight. The thief wants to maximize the total value of the items while not making the total weight exceed *W*. This problem is called the 0–1 Knapsack problem. We assume that all the values are positive integers.

(a) A brute-force solution is to consider all subsets of the *n* items; discard those subsets whose total weight exceeds *W*; and, of those remaining, take one with maximum total profit. Please analyze the time complexity of the brute-force algorithm. (5 分)

(b) An obvious greedy strategy is to steal the items with the largest profit first; that is, steal them in nonincreasing order according to profit. Please give an example to show that this strategy would not work very well. (5 分)

(c) Another obvious greedy strategy is to steal the lightest items first. Please give an example to show that this strategy would also not work very well. (5 分)

(d) To avoid the drawbacks of the previous two greedy algorithms, a more sophisticated greedy strategy is to steal the items with the largest profit per unit weight first. That is, we order the items in nonincreasing order according to profit per unit weight, and select them in sequence. An item is put in the knapsack if its weight does not bring the total weight above W. Please give an example to show that this strategy would still not work very well. (5 分)

(e) We can solve the 0–1 Knapsack problem using dynamic programming. Let A be an optimal subset of the n items. There are two cases: either A contains $item_n$ or it does not. If A does not contain $item_n$, A is equal to an optimal subset of the first n - 1 items. If A does not contain $item_n$, the total profit of the items in A is equal to p_n, the value of $item_n$, plus the optimal profit obtained when the items can be chosen from the first n - 1 items under the restriction that the total weight cannot exceed $W - w_n$, where w_n is the weight of $item_n$. For i>0 and w>0, we let $P[i][w]$ be the optimal profit obtained when choosing items only from the first i items under the restriction that the total weight cannot exceed w. From the above statements, please derive a recursive formula for computing $P[i][w]$. (5 分)

(f) The branch-and-bound design strategy can also be applied to the 0–1 Knapsack problem. Please briefly describe how to do it. (5 分)