國立中央大學100學年度碩士班考試入學試題卷

所別：資訊工程學系碩士班 不分組(一般生)
資訊工程學系軟體工程碩士班 不分組(一般生)

科目：資料結構與演算法 共 2 頁 第 1 頁

本科考試禁用計算器

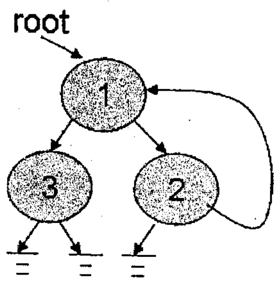
＊請在試卷答案卷（卡）內作答

1. A data structure ***Node*** is defined as follows:

```
Struct Node{
   int key;  // the key value of the item
   Struct Node *left;
   Struct Node *right;
}
```

1.1 (8%) The data structure ***Node*** can be used to form a binary tree. Use pseudocode to design a function ***Search*** that takes two arguments as the input--- a particular key value ***x*** and a pointer ***root*** which points to the root of a binary tree. Then, the program searches for ***x*** from ***root***. If ***x*** is found, return a pointer that points to the node; Otherwise, return Null.

1.2 (10%) The data structure ***Node*** can be used to create a graph that contains cycles. The following figure shows a graph with a cycle 1→2→1. Please use pseudo code to design a function ***IsCycleExist*** that takes one argument as the input --- a pointer ***root*** which points to a node of a graph. The function should return ***TRUE*** if a cycle is found, and return ***FALSE*** otherwise.



2. (5%) A small grid system consists of two identical machines, X and Y. A set of computing jobs, shown in the following table, are submitted to the system for parallel execution. Each job can be identified by its ID, required service time, and arrival time. Each machine can only handle one job at a time. The system uses **a stack** to handle a job of which the required service time is smaller than and equal to 4 seconds; it uses **a queue** otherwise. If a job is waiting in a queue and another one is waiting in a stack, the job in the stack is always selected for execution first.

What is the completion order of the jobs?

Job	1	2	3	4	5	6	7	8	9
Required service time (Sec)	3	4	8	7	4	4	4	7	4
Arrival time (Sec)	t	t+1	t+2	t+3	t+4	t+5	t+6	t+6	t+7

3. A sequence of integers is listed as follows:
2 3 4 5 6 9 8 7

3.1 (5%) Illustrate the binary search tree after each insertion using the above sequence.

3.2 (5%) Illustrate the AVL tree after each insertion using the above sequence.

注意：背面有試題

國立中央大學100學年度碩士班考試入學試題卷

所別：資訊工程學系碩士班 不分組(一般生)
資訊工程學系軟體工程碩士班 不分組(一般生)

科目：資料結構與演算法 共 2 頁 第 2 頁

本科考試禁用計算器

*請在試卷答案卷（卡）內作答

4. Prove or disprove the following statements. Note that the functions should be positive function when n is large enough. So be careful when you construct counter-examples.

4.1 (7%) $f(n) = \Theta(g(n))$ implies $h(f(n)) = O(h(g(n)))$ if $h(n)$ is an increasing function (i.e., $h(n_1) > h(n_2)$ for $\forall n_1 > n_2$);

4.2 (7%) $f(n) + g(n) = \Theta(max\{f(n), g(n)\})$.

5. Let V be a set of n points in the plane. Let $G = (V, E)$ be the complete graph over V, and the weight of each edge $e \in E$ is the length of this edge. The Euclidean Traveling Salesman Problem (ETSP) of V is to find the cycle C^* such that C^* visits each node exactly once, and it has the minimum weight among all such cycles. Let T^* be a minimum spanning tree of G

5.1 (10%) Show that $\omega(T^*) \le \omega(C^*)$, where $\omega(X)$ is the weight of a subgraph X of G.

5.2 (10%) Given T^*, design an algorithm to compute a cycle C that is a 2-approximation of the optimal ETSP cycle C^*. Namely, $\omega(C^*) \le \omega(C) \le 2\omega(C^*)$. *(Hint: first show that* $\omega(T^*) \le \omega(C^*) \le 2\omega(T^*)$.*)*

6. Below are the inputs of an instance of the 0/1 knapsack problem with n=5 items and the knapsack capacity M=38. Note that P_i is the profit of item i and W_i is the weight of item i, i=1,…,5. The 0/1 knapsack problem is to maximize the total profit of items that are wholly put into the knapsack under the constraint that their total weight should not exceed the knapsack capacity.

i	1	2	3	4	5
P_i	6	10	4	5	6
W_i	10	19	8	10	12

$(P_i/W_i \ge P_{i+1}/W_{i+1})$

參考用

The problem can be solved by the branch-and-bound strategy on the basis of the best-first search scheme by deriving the lower bound and upper bound for each node in the search tree.

6.1 (12%) Derive the upper bound and the lower bound for the root node and explain how you obtain the bounds.

6.2 (6%) Explain how the bounds can be used in the branch-and-bound strategy. (6%)

7. Given a chain $A_1, A_2, \ldots, A_n$ of n matrices, where matrix A_i has dimension $d_{i-1} \times d_i$ for i=1,…,n, the matrix-chain multiplication problem is to find a way to fully parenthesize the product $A_1A_2\ldots A_n$ so that the number of scalar multiplications required to compute the product is minimized. A product of matrices is fully parenthesized if it is either a single matrix, or a product of two fully parenthesized matrix product, surrounded by parentheses. For example, $A_1A_2A_3$, where A_1 is a 10×100 matrix, A_2 is a 100×5 matrix and A_3 is a 5×50 matrix, is optimally and fully parenthesized as $((A_1A_2)A_3)$, which requires $10\times100\times5+10\times5\times50=7500$ scalar multiplications.

7.1 (10%) Write an algorithm to take a chain $A_1, A_2, \ldots, A_n$ of n matrices as inputs and to output the minimized number of scalar multiplications required to compute the product of the n matrices on the basis of the dynamic programming strategy.

7.2 (5%) Use your algorithm to calculate the minimized number of scalar multiplications for computing the product $A_1A_2A_3A_4A_5$, where A_1 is a 3×100 matrix, A_2 is a 100×5 matrix, A_3 is a 5×50 matrix, A_4 is a 50×20 and A_5 is a 20×6 matrix.

注意：背面有試題