國立成功大學

110學年度碩士班招生考試試題

編　號：111

系　所：工程科學系

科　目：資料結構

日　期：0203

節　次：第 1 節

備　註：不可使用計算機

編號： 111　　國立成功大學 110 學年度碩士班招生考試試題

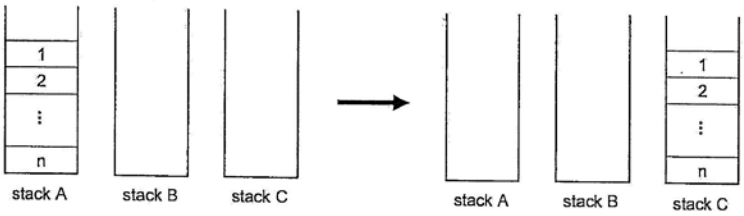
系　　所：工程科學系

考試科目：資料結構　　考試日期：0203，節次：1

※ 考生請注意：本試題不可使用計算機。　請於答案卷(卡)作答，於本試題紙上作答者，不予計分。

1. The well-known puzzle "Tower of Hanoi" can be stated as follows. Given three stacks, i.e., stack A, B and C, suppose that originally stack A contains *n* elements while stack B and C contain no element. The goal is to move all elements from stack A to stack C with the following constraints.

- Only one element can be moved between two stacks during a movement.
- The elements of each stack should be in ascending ordering in a top-down fashion at all times. (i.e., an element with a larger number cannot locate in top of an element with a smaller number within the same stack)



A. The number of movements to reach the desired goal can be formulated as T(*n*), i.e., a function of *n*. Then, what is the value of T(3)?　(5%)

B. By analyzing this puzzle carefully, we can have T(*n*) = 2*T(*n*-1) + 1. Please explain this equation in details to illustrate *why* this equation holds and *how* this equation reflects a recursive function. (10%)

C. What is the general form of T(*n*)?　(5%)

2. Suppose you are converting an array **double A[n]** into a linked list.

A. Please define a structure (such as **struct** in C or C++) of the linked list used to store the n values of the original array. (5%)

B. Please construct the algorithms for performing subscription, i.e. indexing, for both *lvalue* and *rvalue* of an element, that is **A[i] = value** and **value = A[k]**. (5%)

3. For the binary tree defined as

```
typedef struct node *tree-pointer;
typedef struct node {
        int data;
        tree_pointer left_child, right_child;
        };
```

Please using recursive algorithm to write down pre-order and post-order traversals. (10%)

國立成功大學 110 學年度碩士班招生考試試題

系　　所：工程科學系

考試科目：資料結構

考試日期：0203，節次：1

4. Given the adjacent matrix of a graph as follows, please find (you don't have to write a program)

   A. The minimum spanning tree. (10%)
   B. The shortest paths from A to all other nodes. (10%)

	A	B	C	D	E	F	G	Z
A	0	2	0	0	0	1	0	0
B	2	0	2	2	4	0	0	0
C	0	2	0	0	3	0	0	1
D	0	2	0	0	4	3	0	0
E	0	4	3	4	0	0	7	0
F	1	0	0	3	0	0	5	0
G	0	0	0	0	7	5	0	6
Z	0	0	1	0	0	0	6	0

5. Compete the following recursive code for pre-order traversal. (15%)

```
void preorder(tree_pointer ptr)
/* preorder tree traversal */
{
    if (ptr) {
        printf("%d", ptr->data);
```

6. For the unsorted sequence: 27, 6, 38, 2, 60, 12, 58, 16, 47, 18

   A. Please write down every sequence of each step while applying quicksort. (10%)
   B. Show the mergesort for it by binary tree. (15%)