数据结构基础-期中考试试卷

开始时间	2021/11/15 15:55:00	**结束时间**	2021/11/15 16:40:00	**答题时长**	45分钟
答卷类型	标准答案	**总分**	100		

判断题

得分：暂无 总分：27

1-2 An algorithm may or may not require input, but each algorithm is expected to produce at least one result as the output. (3分)

● T ○ F

1-3 $N^3 \log N$ and $N \log N^3$ have the same speed of growth. (3分)

○ T ● F

1-4 $N \log N^2$ and $N \log N^3$ have the same speed of growth. (3分)

● T ○ F

1-5 ADT is the abbreviation for Abstract Data Type in the textbook of data structures. (3分)

● T ○ F

1-13 There are more NULL pointers than the actual pointers in the linked representation of any binary tree. (3分)

● T ○ F

1-11 The number of degree 3 nodes in a ternary tree (三叉树) is only related to the number of degree 2 nodes and that of leaf nodes, i.e it has nothing to do with the number of degree 1 nodes. (3分)

● T ○ F

1-6 For a sequentially stored linear list of length N, the time complexities for deleting the last element and inserting the first element are $O(1)$ and $O(N)$, respectively. (3分)

● T ○ F

1-8 The time comlexity of Selection Sort will be the same no matter we store the elements in an array or a linked list. (3分)

● T ○ F

1-15 The preorder traversal sequence of any min-heap must be in sorted (non-decreasing) order. (3分)

○ T ● F

单选题

得分：暂无 总分：55

2-4 For the following function (where $n > 0$) (5分)

```
int func ( int n )
{   int i = 1, sum = 0;
    while ( n > sum )  { i *= 2; sum += i; }
    return i;
}
```

the most accurate time complexity bound is:

○ A. $O(2^n)$
○ B. $O(n)$
○ C. $O(n\log n)$
● D. $O(\log n)$

2-21 Suppose that enqueue is allowed to happen at both ends of a queue, but dequeue can only be done at one end. If elements are enqueued in the order {a, b, c, d, e}, the impossible dequeue sequence is: (5分)

○ A. b a c d e
○ B. d b a c e
○ C. e c b a d
● D. d b c a e

2-22 A tri-diagonal matrix is a square matrix with nonzero elements only on the diagonal and slots horizontally or vertically adjacent the diagonal, as shown in the figure. (5分)

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 & \cdots & 0 & 0 \\ a_{21} & a_{22} & a_{23} & \ddots & \ddots & 0 & 0 \\ 0 & a_{32} & a_{33} & \ddots & \ddots & a_{n-2,n-1} & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & \cdots & \cdots & \cdots & a_{n,n-1} & a_{n\,n} \end{bmatrix}.$$

Given a tri-diagonal matrix (三对角矩阵) M of order 100. Compress the matrix by storing its tri-diagonal entries $m_{i,j}$ ($1 \le i \le 100$，$1 \le j \le 100$) row by row into a one dimensional array N with indices starting from 0. Then the index of $m_{30,30}$ in N is:

○ A. 86

● B. 87

○ C. 88

○ D. 89

2-18 What kind of tree has the property that the nodes along the path from the root to any node are in sorted order? (5分)

○ A. binary search tree

○ B. complete binary tree

● C. heap

○ D. full binary tree

2-1 Given the popping sequence of a stack as { a, b, c, d, e, f }. Among the following, the impossible pushing sequence is: (5分)

○ A. c b a f e d

● B. d f e a c b

○ C. f e a b c d

○ D. f e d a b c

2-7 For a non-empty doubly linked circular list, with `h` and `t` pointing to its head and tail nodes, respectively, the TRUE statement is: (5分)

● A. `t->next == h`

○ B. `h->pre == NULL`

○ C. `t->next == h->next`

○ D. `h->next == t`

2-8 Suppose that the level-order traversal sequence of a min-heap is { 2, 17, 5, 46, 22, 8, 10 }. Use the linear algorithm to adjust this min-heap into a max-heap, and then call DeleteMax. The postorder traversal sequence of the resulting tree is: (5分)

○ A. 2, 8, 17, 5, 10, 22

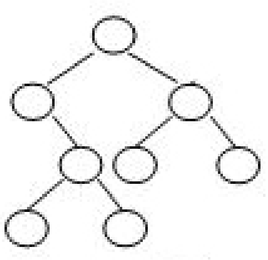
○ B. 22, 17, 5, 2, 10, 8

● C. 5, 2, 17, 8, 10, 22

○ D. 2, 8, 10, 5, 17, 22

2-11 Given the shape of a binary tree shown by the figure below. If its inorder (5分)

traversal sequence is { D, E, A, B, F, H, C, G }, then the node on the same level of H must be:



○ A. E and G

○ B. B

○ C. E

● D. A and G

2-26 Given a binary search tree with its postorder traversal sequence { 2, 7, 15, 10, 20, 19, 35, 21, 18 }. If 18 is deleted from the tree, which one of the following statements is FALSE? (5分)

○

A. One possible preprder traversal sequence of the resulting tree may be { 15, 10, 7, 2, 21, 19, 20, 35 }

●

B. One possible preprder traversal sequence of the resulting tree may be { 20, 10, 7, 2, 15, 21, 19, 35 }

○

C. One possible preprder traversal sequence of the resulting tree may be { 19, 10, 7, 2, 15, 21, 20, 35 }

○ D. It is possible that the resulting tree may have 3 leaves

2-17 In a complete binary tree with 1534 nodes, there must be ____ leaf nodes. (5分)

○ A. 510

○ B. 511

○ C. 766

● D. 767

2-5 What is the major difference among lists, stacks, and queues? (5分)

○ A. Lists use pointers, and stacks and queues use arrays

● B. Stacks and queues are lists with insertion/deletion constraints

○

C. Lists and queues can be implemented using circularly linked lists, but stacks cannot

○ D. Lists are linear structures while stacks and queues are not

程序填空题

得分：暂无 总分：18

5-2 The function `BuildTree` is to build and return a binary tree from its inorder and postorder traversal sequences.

The tree structure is defined as the following:

```
typedef struct Node *PtrToNode;
struct Node{
    int Data;
    PtrToNode Left, Right;
};
typedef PtrToNode Tree;
```

Please fill in the blanks.

```
Tree BuildTree( int in[], int post[], int N )
{ //in[] stores the inorder traversal sequence
  //and post[] stores the postorder traversal sequence
  //N is the number of nodes in the tree
     Tree T;
     int i;

     if (!N) {
        return NULL;
     }
     T = (Tree)malloc(sizeof(struct Node));
     T->Data = post[N-1] (3分);
     for (i=0; i<N; i++)
         if (in[i]==T->Data) break;
     T->Left = BuildTree( in, post, i (3分));
     T->Right = BuildTree( in+i+1, post+i, N-i-1 (3分));
     return T;
}
```

5-6 Concatenation of lists is an operation where the elements of one list are added at the end of another list. For example, if we have a linked list `L1` →1→2→3 and another one `L2` →4→5→6. The function `ListConcat` is to return the head pointer of the list L→4→5→6→1→2→3.

The list structure is defined as the following:

```
typedef struct Node *PtrToNode;
struct Node{
    int Data;
    PtrToNode Next;
};
typedef PtrToNode List;
```

Please fill in the blanks.

```
List ListConcat( List L1, List L2 )
{
    List Tmp = L2;
    if ( !L2 ) return L1;
    while ( Tmp->Next )
        Tmp = Tmp->Next (3分);
    Tmp->Next = L1 (3分);
    return L2 (3分);
}
```