《数据结构基础》

2021-2022 学年第一学期期中考试试卷

一、判断题（10 题）

R1-1 ADT is the abbreviation for Abstract Data Type in the textbook of data structure （ ）

R1-2 The major task of algorithm analysis is to analyze the time complexity and the space complexity ( )

R1-3 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( )

R1-4 If the preorder and inorder traversal sequences of a binary tree are the same ,then none of the nodes in the tree has a left child. ( )

R1-5 The number of leaf nodes in a complete binary tree with 124 nodes is definite.( )

R1-6 N^3logN and N$\log N^3$ have the same speed of growth. ( )

R1-7 The time complexity of Binary Search will be the same no matter we store the element in an array or a linked list. ( )

R1-8 To find 63 from a binary search tree,one possible searching sequence is{39,101,25,80,70,59,63}. ( )

R1-9 The inorder traversal sequence of any min-heap must be in sorted order.( )

二、单选题(20 题)

R2-1 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 enqueue in the order {a,b,c,d,e},the impossible dequeue sequence is:

A、e c b a d　　B、b a c d e　　C、d b a c e　　D、d b c a e

R2-2 For a non-empty doubly linked circular list,with h and t pointing to its head and tail nodes,respectively,the TRUE statement is:

A. t->next==h　　B. h->next==t　　C. h->pre==NULL　　D. t->next==h->next

R2-3 Given the pushing sequence of a stack as {6,5,4,3,2,1}.Among the following,the impossible popping sequence is:

A. 3 4 6 5 2 1　　B. 5 4 3 6 1 2

C. 2 3 4 1 5 6　　D. 4 5 3 1 2 6

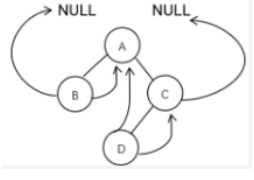
R2-4 What kind of tree has the property that the nodes along the path from the root to any node are in sorted order?

A. complete binary tree　　B. binary search tree

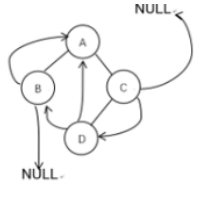
C. None of the tree　　D. heap

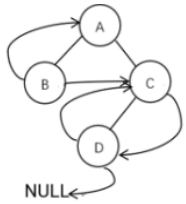
R2-5 Among the following threader binary trees(the threads are represented by arrows).which one is the pre-orderr threader tree?

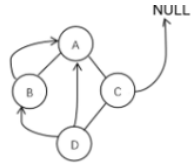
A.



B、



C、 

D、 

R2-6 What is the major difference among lists,stacks,and queues?

A、Lists use pointers ,and stacks and queues use arrays

B、Lists and queues can be implemented using circularly linked lists,but stacks cannot

C、Stacks and queues are lists with insertion/deletion constraints

D、Lists are linear structure while stacks and queues are not

R2-7 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.

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

Given a tri-diagonal matrix(三对角矩阵） M of order 100.Compass the matrix by storing its tri-diagonal entries $m_{i,j}$(1<=i<=100,1<=j<=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:

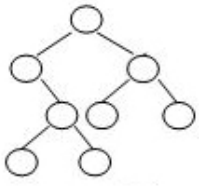
A、87

B、88

C、86

D、89

R2-8 Given the shape of a binary tree shown by the figure below.If its preorder traversal sequence is{E,D,A,F,H，C,B,G},then the node on the same level of F must be:



A. H B. A and G C. B D. C and G

R2-9 For a sequentially stored linear list of length N ,which of the following operations has time complexity O（1）?

A. visit the i-th(1<=i<=N)node and find the immediate predecessor of the i-th(2<=i<=n)node

B. insert a new node after the i-th(1<=i<=N)node.

C. sort the N nodes in inserting order

D. delete the i-th(1<=i<=N)node

R2-10 Suppose that the level-order traversal sequence of a max-heap is(48,27,32,12,18,20,15).Use the linear algorithm to adjust this max-heap into a min-heap,and then call DeleteMin.The post-order traversal sequence of the resulting tree is:

A. 15,18,20,27,48,32

B. 27,48,18,32,20,15

C. 32,48,27,20,15,18

D. 48,18,27,20,32,15

R2-11 For the following function (where n>0)

```
int func(int n)
{   for(i=0;i<n;i++)
        for(j=i;i>0;j/=2)
            printf("%d\n",j);
}
```

The most accurate time complexity bound is:

A. O(logn) B. O(n) C. $O(n^2)$ D. O(nlogn)

R2-12 The array representation of the disjoint sets is given by{3,3,-5,2,1,-3,-1,6,6}.Keep in mind that the elements are numbered from from 1 to 9.After invoking Union (Find(4),Find(8)) with union-by-size and path compression,how many elements will be change in the resulting array?

A. 3 B. 4 C. 2 D. 1

R2-13 In a complete binary tree with 1102 nodes,there must be __leaf nodes.

A. 79 B. 551 C. 1063 D. cannot be determined

R2-14 Given a binary search tree with its post-order 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?

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

B、It is possible that the resulting tree may have 3 leaves

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

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

三、程序填空题

R5-1 The function BuildTree is to build and return a binary tree from its inorder and postorder traveral sequences.

The tree structure is defined as the the following :

```
typedef struct Node *PtrToNode:

struct Node{

    int Data;

    PtrToNode left,Right;

    };
```



```
    Typedef PtrToNode Tree;
```

Please fill in the blank.

```
Tree BuildTree(int in[],int post[],int N)
{   //in[] stores the in-order traversal sequence
    //and post[] stores the post-order traversal sequence
    //N is number of nodes in the tree
    Tree T;
    int i;
    If(!N){
        Return NULL;
        }
    T =(Tree)malloc(sizeof(struct Node));
    T->data=_____①_____;
    for(i=0;i<N;i++)
        If(in[i]==T->Data)break;
    T->left=BuildTree(_____②_____);
    T->Right=BuildTree(_____③_____);
    return T;
    }
```

四、函数题

R6-1 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->1->2->3->4->5->6.

The list structure is defined as the following:

```
  Typedef struct Node *PtrToNode;
    Struct Node{
        Int data;
        PtrToNode Next;
        };
    Typedef PtrToNode List;
```

函数接口定义：

```
List ListConcat(List L1,List L2);
```

裁判测试程序样例：

```
#include <stdio.h>
#include<stdio.h>
Typedef struct Node *PtrToNode;
    Struct Node{
        Int data;
        PtrToNode Next;
        };
    Typedef PtrToNode List;
List.Read();
void Print(List L);
List ListConcat(List L1,List L2);
int main()
{   List L1,L2,L;
    L1=Read();
    L2=Read();
    L=ListConcat(L1,L2);
    Print(L);
    Return 0;
}
List Read()
{   int n,i,a;
    List   head,p,q,t;
    scanf("%d",&n);
    head=NULL;
    for(i=0;i<n;i++)
    {   p=(List)malloc(sizeof(struct Node));
        scanf("%d",&a);
        p->Data=NULL;
        if(head==NULL)
            head=p;
        else
            q->next=p;
        q=p;
        }
        return head;
    }
```

```
void Print(List L)
{    List p=L;
     if(p==NULL)
          Printf(“NULL”);
     While(p)
     {    printf(“%d”,p->Data);
          p=p->Next;
     }
}
```

/*请在这里填写答案*/

输入样例：

在这里给出一组输入，例如：

3

4 5 6

3

1 2 3

♁结尾无空行

输出样例：

在这里给出相应的输出，例如：

4 5 6 1 2 3

♁结尾无空行

2021-2022 学年第一学期期中考试试卷参考答案

一、判断题（10 题）

1、【正解】T

【解析】ADT 是《数据结构》教材中抽象数据类型的缩写。

【考点延伸】抽象数据类型

2、【正解】T

【解析】算法分析的主要任务是分析算法的时间复杂度和空间复杂度

【考点延伸】算法分析的任务

3、【正解】T

【解析】在顺序表中删除最后一个元素不需要移动元素，因此时间复杂度为 O(1)。在顺序表中的第一个位置插入元素的时间复杂度为 O(n)。

【考点延伸】时间复杂度

4、【正解】T

【解析】前序遍历是根左右，中序遍历是左根右。若前序遍历和中序遍历的顺序相同，则该树没有左孩子。

【考点延伸】二叉树的遍历

5、【正解】T

【解析】完全二叉树的叶子结点数公式：n0=(n+1)/2.当 n 为偶数时，n0=n/2;当 n 为奇数时，n0=(n+1)/2。因此 n=124 时，n0=62

【考点延伸】完全二叉树

6、【正解】F

【解析】N^3logN 的增长速度大于 $NlogN^3$

【考点延伸】时间复杂度

7、【正解】F

【解析】二分法查找也称折半查找，它适合于按键值排序的存储结构。在二分法查找时，每次取中间一个数据元素进行判断，若找到，则停止查找，否则决定取其前一半或后一半数据元素继续查找。因此二分法查找只能用于已经排序的顺序存储结构。因此二分查找不适应链式存储。

【考点延伸】二分查找

8、【正解】F

【解析】二叉搜索树：若它的左子树不空，则左子树上所有结点的值均小于它的根结点的值；若它的右子树不空，则右子树上所有结点的值均大于它的根结点的值；该序列不符合二叉搜索树。

【考点延伸】二叉搜索树

9、【正解】F

【解析】 任何最小堆中从根结点到任一叶结点路径上的所有结点是有序的

【考点延伸】堆

二、单选题(20 题)

1、【正解】D

【解析】由题意知，队列可由两端进，一端出，因此 D 选项是不能实现的。

【考点延伸】队列

2、【正解】A

【解析】循环双链表的尾结点的指针指向头结点，故 A 正确。B 选项的情况是只头结点和尾结点，不是涵盖全部情况。C 应该为 h->pre==t。

【考点延伸】循环双链表

3、【正解】A
【解析】栈的性质是先进后出。A 的正确顺序是 345621

【考点延伸】栈

4、【正解】D

【解析】任何最小（大）堆中从根结点到任一叶结点路径上的所有结点是有序的

【考点延伸】堆

5、【正解】C

【解析】二叉树中每个结点的空左孩子指向前驱，空右孩子指向后继；若无前驱/后继则引出为 NIL。

【考点延伸】二叉搜索树

6、【正解】C

【解析】序列，堆栈和队列的主要不同： 堆栈和队列是带有插入/删除约束的列表

【考点延伸】栈，对

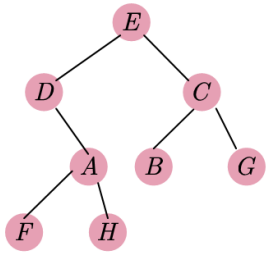
7、【正解】A

【解析】 $m_{30,30}$ 是在对角线上的，因此计算方式为 2+28*3+2=88,下标从 0 开始，故选 87

【考点延伸】三对角阵

8、【正解】A

【解析】该二叉树如下，因此选 A



【考点延伸】二叉树的遍历

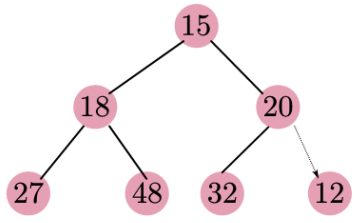
9、【正解】A

【解析】BD 的时间复杂度为 O(n)。C 的时间复杂度取决于使用的排序算法。

【考点延伸】时间复杂度

10、【正解】B

【解析】最后调整后的堆为：



【考点延伸】堆

11、【正解】D

【解析】第二个 for 循环是从 i 开始的，所以小于 $O(n^2)$，因此时间复杂度为 O(nlogn)。

【考点延伸】时间复杂度

12、【正解】A

【解析】并查集里的 find 函数里可以进行路径压缩，是为了更快速的查找一个点的根节点。对于一个集合树来说，它的根节点下面可以依附着许多的节点，因此，我们可以尝试在 find 的过程中，从底向上，如果此时访问的节点不是根节点的话，那么我们可以把这个节点尽量的往上挪一挪，减少数的层数，这个过程就叫做路径压缩。

【考点延伸】并查集

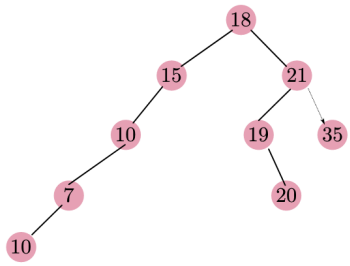
13、【正解】B

【解析】完全二叉树的叶子结点数公式：n0=(n+1)/2.当 n 为偶数时，n0=n/2;当 n 为奇数时，n0=(n+1)/2。n=1102，因此叶子结点为 551.

【考点延伸】完全二叉树

14、【正解】D

【解析】该二叉树可能为：



因此 D 选项是不能实现的。

【考点延伸】二叉树

三、程序填空题

1、【解析】①post[N-1]　②in,post,i　③in+i+1,post+i+1，N-i,

【考点延伸】二叉树的遍历

四、函数题

1、【解析】List listConcat(List L1,List L2)

```
{
    if(L1==NULL)return L2;
    else if(L2==NULL)return L1;
    List head=L1;
    while(L1->Next!=NULL) L1=L1->Next;
    L1->next=L2;
    return head;
}
```

【考点延伸】链表