浙江大学2018–2019学年秋冬学期 《数据结构基础》课程期末考试试卷

1-1 If N numbers are stored in a doubly linked list in increasing order, then the average time complexity for binary search is $O(logN)$. (2分)

○ T ◉ F

1-1 答案正确 (2 分)

1-2 Mergesort is stable. (2分)

◉ T ○ F

1-2 答案正确 (2 分)

1-3 If the preorder and the postorder traversal sequences of a binary tree have exactly the opposite orders, then none of the nodes in the tree has two subtrees. (2分)

◉ T ○ F

1-3 答案正确 (2 分)

1-4 Linear probing is equivalent to double hashing with a secondary hash function of $Hash_2(k) = 1$. (2分)

◉ T ○ F

1-4 答案正确 (2 分)

1-5 To sort N records, heap sort requires at least $O(N)$ extra space. (2分)

○ T ◉ F

1-5 答案正确 (2 分)

1-6 $logN^{20}$ is $O(N)$. (2分)

◉ T ○ F

1-6 答案正确 (2 分)

1-7 If keys are pushed onto a stack in the order `abcde`, then it's impossible to obtain the output sequence `cedab`. (2分)

◉ T ○ F

1-7 答案正确 (2 分)

1-8 Let P be the shortest path from S to T. If the weight of every edge in the graph is incremented by 1, P will still be the shortest path from S to T. (2分)

○ T ◉ F

1-8 答案正确 (2 分)

1-9 For a graph, if each vertex has an even degree, we can find an Euler circuit that visits every vertex exactly once. (2分)

◉ T ○ F

1-9 答案错误 ⓘ (0 分)

1-10 Given a binary search tree with 20 integer keys which include 5, 6, and 7, if 5 and 7 are on the same level, then 6 must be their parent. (2分)

○ T ◉ F

1-10 答案正确 (2 分)

2-1 In order to convert the infix expression `4 + 3 * (6 * 3 - 12)` to postfix expression using a stack S, then the minimum size of S must be: (3分)

○ A. 2
○ B. 3
◉ C. 4
○ D. 5

2-1 答案正确 (3 分)

2-2 Suppose that the size of a hash table is 11, and the hash function is H(key)=key%11. The following 4 elements have been inserted into the table as Addr(14)=3, Addr(38)=5, Addr(61)=6, Addr(86)=9. When open addressing with quadratic probing is used to solve collisions, the address of the element with key=49 will be . (3分)

○ A. 5
◉ B. 10
○ C. 7
○ D. 8

2-2 答案正确 (3 分)

2-3 Given input {15, 9, 7, 8, 20, -1, 4}. If the result of the 1st run of Shell sort is {15, -1, 4, 8, 20, 9, 7}, then the initial increment must be: (3分)

○ A. 1
○ B. 2
○ C. 3
◉ D. 4

2-3 答案正确 (3 分)

2-4 Which one of the following is a possible postorder traversal sequence of a binary search tree? (3分)

○ A. `3 1 6 2 5 8 10 9 7 4`
○ B. `3 1 2 6 5 8 10 9 7 4`
◉ C. `1 3 2 6 5 8 10 9 7 4`
○ D. `1 3 2 6 8 5 10 9 7 4`

2-4 答案正确 (3 分)

2-5 Given input {46, 79, 56, 38, 40, 84}. After the first partition (with the left most record as the pivot) of quick sort, the resulting sequence is: (3分)

○ A. {38,46,79,56,40,84}
○ B. {38,79,56,46,40,84}
○ C. {38,46,56,79,40,84}
◉ D. {40,38,46,56,79,84}

2-5 答案正确 (3 分)

2-6 Following is the C-like pseudo code of a function that takes a Queue as an argument.

```
void foo(Queue Q)
{
    Queue Q1 = CreateQueue();  // create an empty queue

    while (!IsEmpty(Q))
    {
        // dequeue an item from Q and enqueue it into Q1
        Enqueue(Q1, Dequeue(Q));
    }

    while (!IsEmpty(Q1))
    {
        // dequeue an item from Q1 and enqueue it into Q
        Enqueue(Q, Dequeue(Q1));
    }

    DisposeQueue(Q1);
}
```

What does the above function do? (3分)

- A. Removes the last item from Q
- B. Keeps Q unchanged
- C. Makes Q empty
- D. Reverses Q

2-6 答案正确 (3 分)

2-7 Let T be a tree of N nodes created by union-by-size without path compression, then the minimum depth of T may be (3分)

- A. 1
- B. $logN$
- C. $N-1$
- D. $N/2$

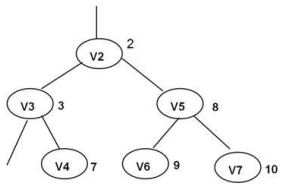
2-7 答案正确 (3 分)

2-8 For an in-order threaded binary tree, if the pre-order and in-order traversal sequences are `D A B C F E` and `B A C D E F` respectively, which pair of nodes' left links are both threads? (3分)

- A. D and A
- B. A and F
- C. C and F
- D. C and E

2-8 答案正确 (3 分)

2-9 The following is the part of depth-first search tree to find the articulation points, and the Num(v) value has been marked beside each vertex v. The back edges are not shown. Which of the following situation is impossible? (3分)



- A. low(v6) is greater than low(v7)
- B. low(v4) is 2
- C. low(v6) is 3
- D. low(v7) is 2

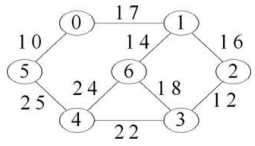
2-9 答案正确 (3 分)

2-10 Insert {18, 23, 4, 26, 31, 33, 17, 39} one by one into an initially empty hash table of size 13 with the hash function $H(Key) = Key\%13$, and linear probing is used to resolve collisions. What is the loading density when the first collision occurs? (3分)

- A. 0.54
- B. 0.63
- C. 0.31
- D. 0.62

2-10 答案正确 (3 分)

2-11 To find the minimum spanning tree with Prim's algorithm for the following graph, a sequence of vertexes 6, 1, 2, 3 was found during the algorithm's early steps. Which one vertex will be added in the next step? (3分)

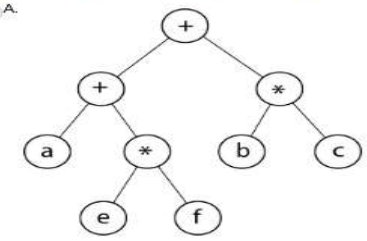


- A. 0
- B. 4
- C. 5
- D. the vertex serial is incorrect

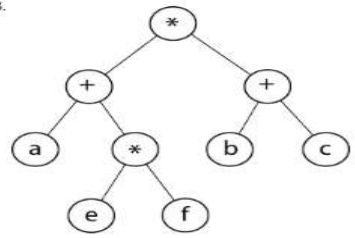
2-11 答案正确 (3 分)

2-12 Which one of the following is the expression tree corresponding to the postfix expression `aef*+bc*+` ? (3分)

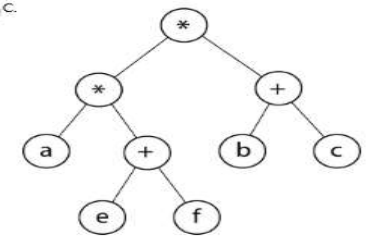
A.



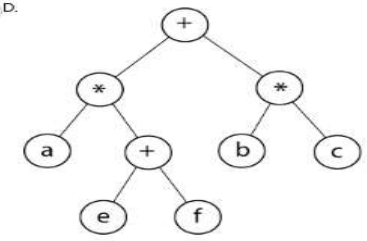
B.



C.



D.



2-12 答案正确 (3 分)

2-13 It is known that a 3-heap is a heap whose nodes have 3 children. Suppose that the level-order traversal sequence of a max-3-heap is {88, 76, 65, 82, 68, 46, 52, 44, 62, 33, 75, 60, 55, 28}. Use the linear algorithm to adjust this max-3-heap into a min-3-heap, and then run DeleteMin. As a result, there are __ nodes whose positions are not moved in the process. (3分)

A. 2
B. 3
C. 4
D. 5

2-13 答案正确 (3 分)

2-14 A graph with 100 vertices and 12 edges must have at most __ connected component(s). (3分)

A. 87
B. 88
C. 94
D. 95

2-14 答案正确 (3 分)

2-15 When inserting a new key `K` into a binary search tree `T` with 512 nodes, the worst-case number of comparisons between `K` and the keys already in `T` is in the range of: (3分)

A. [10, 511]
B. [9, 511]
C. [9, 512]
D. [10, 512]

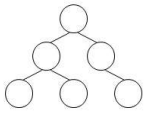
2-15 答案正确 (3 分)

2-16 Given a tree of degree 6. Suppose that the numbers of nodes of degrees 1, 2, 3, 4, 5, 6 are 3, 5, 1, 2, 4, 3, respectively. Then the number of leaf nodes must be: (3分)

A. 35
B. 39
C. 43
D. 45

2-16 答案正确 (3 分)

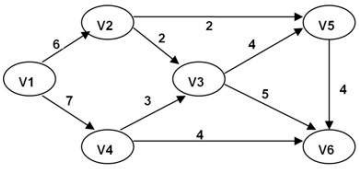
2-17 Given the structure of a binary search tree (as shown in the figure), which one of the following insertion sequences is impossible? (3分)



A. `83 67 91 98 20 75`
B. `83 67 75 91 20 98`
C. `83 91 75 67 20 98`
D. `83 91 98 67 75 20`

2-17 答案正确 (3 分)

2-18 The maximum flow from v1 to v6 is __: (3分)



- A. 11
- B. 12
- C. 13
- D. 0

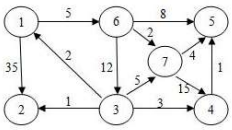
2-18 答案正确 (3 分)

2-19 The inorder and the postorder traversal sequences of a binary tree are `a b c d e f g` and `a c b g f e d`, respectively. Then the preorder traversal sequences is: (3分)

- A. `d b a c f e g`
- B. `d b a c e f g`
- C. `d a c b f e g`
- D. `d c a b e f g`

2-19 答案正确 (3 分)

2-20 Use Dijkstra algorithm to find the shortest paths from 1 to every other vertices. In which order that the destinations must be obtained? (3分)



- A. 6, 7, 5, 3, 2, 4
- B. 6, 2, 5, 7, 3, 4
- C. 2, 3, 4, 5, 6, 7
- D. 2, 4, 3, 6, 5, 7

2-20 答案正确 (3 分)

5-1 The function is to find the `K`-th largest element in a list `A` of `N` elements. The function `BuildMinHeap(H, K)` is to arrange elements `H[1]` ... `H[K]` into a min-heap. Please complete the following program.

```
ElementType FindKthLargest ( int A[], int N, int K )
{   /* it is assumed that K<=N */
    ElementType *H;
    int i, next, child;

    H = (ElementType *)malloc((K+1)*sizeof(ElementType));
    for ( i=1; i<=K; i++ ) H[i] = A[i-1];
    BuildMinHeap(H, K);

    for ( next=K; next<N; next++ ) {
        H[0] = A[next];
        if ( H[0] > H[1] ) {
            for ( i=1; i*2<=K; i=child ) {
                child = i*2;
                if ( child!=K && H[child]>H[child+1]      (3分) ) child++;
                if ( H[child]<H[0]                        (3分) )
                    H[i] = H[child];
                else break;
            }
            H[i] = H[0];
        }
    }
    return H[1];
}
```

5-1 答案正确 (6 分)

5-2 Given an array `a[]` of `n` integers, the function `MissingMin` is to find and return the minimum positive integer which is **NOT** in the array. For example, given { 3, -1, 8, 1, 0 }, 2 is the smallest positive integer which is missing.

```
int MissingMin( int a[], int n )
{
    int i, j, min, missing=1;

    for( i=0; i<n; i++ ){
        min = i;
        for( j = i+1; j < n; j++ )
            if (a[j]<a[min]                   (3分))  min = j;
        if ( min != i )  swap(a[i], a[min]);
        if ( a[i] == missing )  missing++;
        else if ( a[i] > missing )  break          (3分);
    }
    return missing;
}
```

5-2 答案正确 (6 分)

Write a program to test if a give sequence `Seq` is a topological order of a given graph `Graph`.

Format of functions:

```
bool IsTopSeq( LGraph Graph, Vertex Seq[] );
```

where `LGraph` is defined as the following:

```
typedef struct AdjVNode *PtrToAdjVNode;
struct AdjVNode{
    Vertex AdjV;
    PtrToAdjVNode Next;
};

typedef struct Vnode{
    PtrToAdjVNode FirstEdge;
} AdjList[MaxVertexNum];

typedef struct GNode *PtrToGNode;
struct GNode{
    int Nv;
    int Ne;
    AdjList G;
};
typedef PtrToGNode LGraph;
```

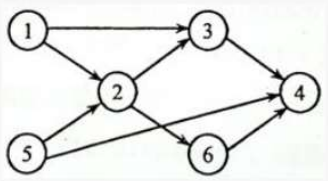
The function `IsTopSeq` must return `true` if `Seq` does correspond to a topological order; otherwise return `false`.

Note: Although the vertices are numbered from 1 to MaxVertexNum, they are **indexed from 0** in the LGraph structure.

Sample program of judge:

```
#include <stdio.h>
#include <stdlib.h>

typedef enum {false, true} bool;
#define MaxVertexNum 10  /* maximum number of vertices */
typedef int Vertex;      /* vertices are numbered from 1 to MaxVertexNum */

typedef struct AdjVNode *PtrToAdjVNode;
struct AdjVNode{
    Vertex AdjV;
    PtrToAdjVNode Next;
};

typedef struct Vnode{
    PtrToAdjVNode FirstEdge;
} AdjList[MaxVertexNum];

typedef struct GNode *PtrToGNode;
struct GNode{
    int Nv;
    int Ne;
    AdjList G;
};
typedef PtrToGNode LGraph;

LGraph ReadG(); /* details omitted */

bool IsTopSeq( LGraph Graph, Vertex Seq[] );

int main()
{
    int i, j, N;
    Vertex Seq[MaxVertexNum];
    LGraph G = ReadG();
    scanf("%d", &N);
    for (i=0; i<N; i++) {
        for (j=0; j<G->Nv; j++)
            scanf("%d", &Seq[j]);
        if ( IsTopSeq(G, Seq)==true ) printf("yes\n");
        else printf("no\n");
    }

    return 0;
}

/* Your function will be put here */
```

Sample Input (for the graph shown in the figure):




```
6 8
1 2
1 3
5 2
5 4
2 3
2 6
3 4
6 4
5
1 5 2 3 6 4
5 1 2 6 3 4
5 1 2 3 6 4
5 2 1 6 3 4
1 2 3 4 5 6
```

Sample Output:

```
yes
yes
yes
no
no
```

编译器 (1) C (gcc)

```
bool IsTopSeq( LGraph Graph, Vertex Seq[] ){
    int degree[MaxVertexNum];
    for(int i;i<Graph->Nv;i++){
        degree[i]=0;
    }
    int i;
    PtrToAdjVNode p;
    for(i=0;i<Graph->Nv;i++){
            for(p=Graph->G[i].FirstEdge;p;p=p->Next)
            {
                degree[p->AdjV]++;
            }
        }
    int j;
    for(i=0;i<Graph->Nv;i++){
        if(degree[Seq[i]-1]!=0)
        return 0;
        else{
            for(p=Graph->G[Seq[i]-1].FirstEdge;p;p=p->Next)
            degree[p->AdjV]--;
        }
    }
    return 1;
}
```