浙江大学2018-19秋冬《数据结构基础》期末模拟练习

开始时间 1/1/2016, 12:00:00 AM **结束时间** 1/18/2038, 12:00:00 AM **答题时长** 120分钟
考生 孙恺元 **得分** 91 **总分** 100

判断题

总分：20 得分：20

1-1 For a graph, if each vertex has an even degree or only two vertexes have odd degree, we can find a cycle that visits every edge exactly once (2分)

○ T ◉ F

评测结果：答案正确（2 分）

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

○ T ◉ F

评测结果：答案正确（2 分）

1-3 If N numbers are stored in a singly linked list in increasing order, then the average time complexity for binary search is $O(logN)$. (2分)

○ T ◉ F

评测结果：答案正确（2 分）

1-4 $(logN)^3$ is $O(N)$. (2分)

◉ T ○ F

评测结果：答案正确（2 分）

1-5 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

评测结果：答案正确（2 分）

1-6 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

评测结果：答案正确（2 分）

1-7 Mergesort is stable. (2分)

(•) T ( ) F

评测结果：答案正确（2 分）

1-8 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

评测结果：答案正确（2 分）

1-9 Let M be the minimum spanning tree of a weighted graph G. Then the path in M between V1 and V2 must be the shortest path between them in G. (2分)

( ) T (•) F

评测结果：答案正确（2 分）

1-10 To sort N records by quick sort, the worst-case time complexity is $O(NlogN)$. (2分)

( ) T (•) F

评测结果：答案正确（2 分）

单选题

总分：60 得分：57

2-1 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]

评测结果：答案错误（0 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

评测结果：答案正确（3 分）

2-3 The inorder and the postorder traversal sequences of a binary tree are `a b c d e f g` and `a c b f g 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 g f`
- C. `d a c b f e g`
- D. `d c a b e g f`

评测结果：答案正确（3 分）

2-4 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

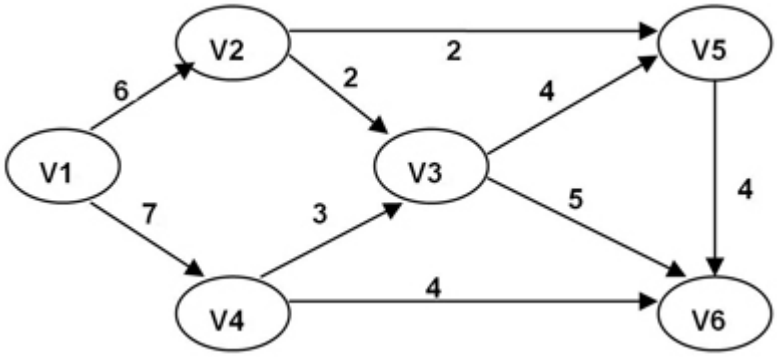
评测结果：答案正确（3 分）

2-5 Given an initially empty hash table `HT` with length 7, together with a hash function $H(k) = k\%7$. Let us use linear probing to solve collisions. What is the average search length for successful searches after inserting 22, 43, 15 one by one into `HT`? (3分)

- A. 1.5
- B. 1.6
- C. 2
- D. 3

评测结果：答案正确（3 分）

2-6 The maximum flow from v1 to v6 is __: (3分)



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

评测结果：答案正确（3 分）

2-7 Which one of the following is a possible postorder traversal sequence of a binary search tree? (3分)

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

评测结果：答案正确（3 分）

2-8 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}

评测结果：答案正确（3 分）

2-9 A graph with 50 vertices and 17 edges must have at most __ connected component(s). (3分)

- A. 32
- B. 33
- C. 44
- D. 45

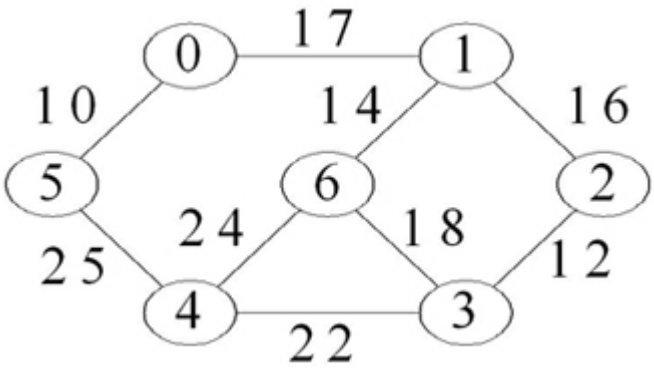
评测结果：答案正确（3 分）

2-10 Given input { 46, 79, 56, 38, 40, 84 }. Which one of the following is the initial heap built by heap sort? (3分)

- A. 79, 46, 56, 38, 40, 80
- B. 84, 79, 56, 46, 40, 38
- C. 84, 56, 79, 40, 46, 38
- D. 84, 79, 56, 38, 40, 46

评测结果：答案正确（3 分）

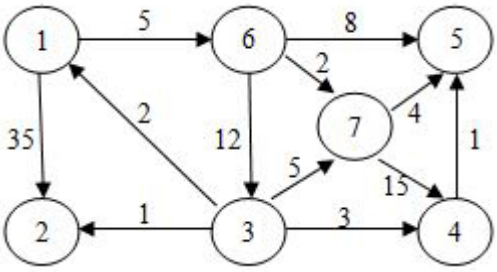
2-11 To find the minimum spanning tree with Prim's algorithm for the following graph, a sequence of vertexes 6, 1, 2, 3 was found during the algorithm's early steps. Which one vertex will be added in the next step? (3分)



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

评测结果：答案正确（3 分）

2-12 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

评测结果：答案正确（3 分）

2-13 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

评测结果：答案正确（3 分）

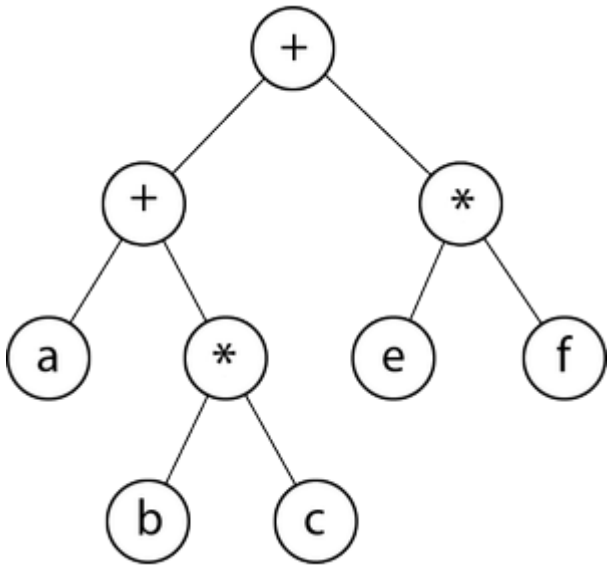
2-14 It is known that a 3-heap is a heap whose nodes have 3 children. Suppose that the level-order traversal sequence of a max-3-heap is {88, 76, 65, 82, 68, 46, 52, 44, 62, 33, 75, 60, 55, 28}. Use the linear algorithm to adjust this max-3-heap into a min-3-heap, and then run DeleteMin. As a result, there are __ nodes whose positions are not moved in the process. (3分)

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

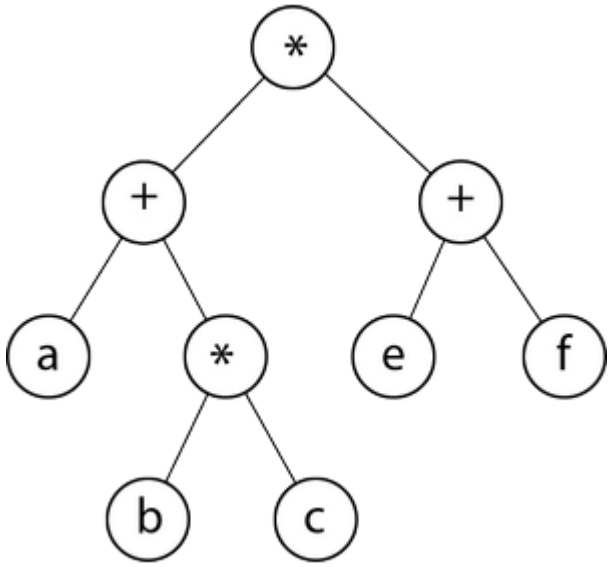
评测结果：答案正确（3 分）

2-15 Which one of the following is the expression tree corresponding to the postfix expression `abc*+ef*+`? (3分)

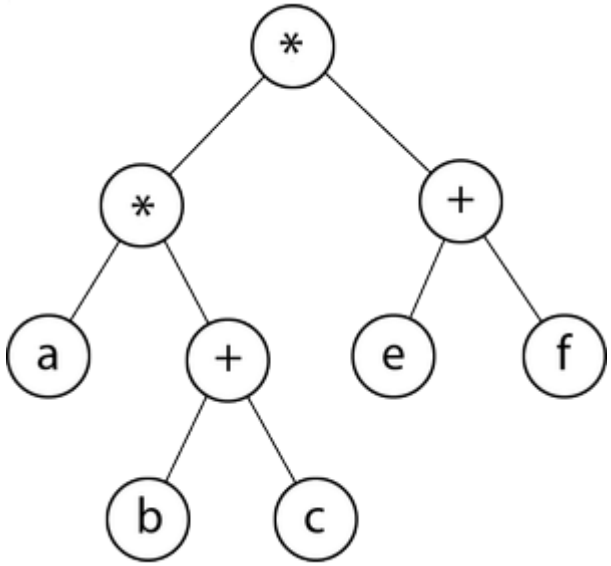
A.



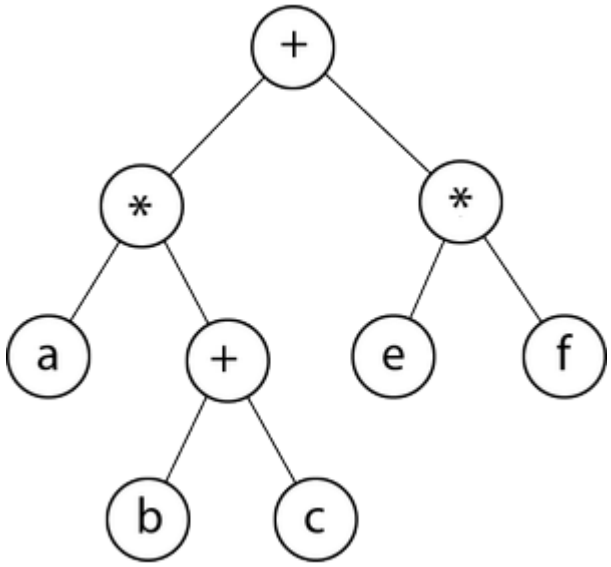
B.



C.



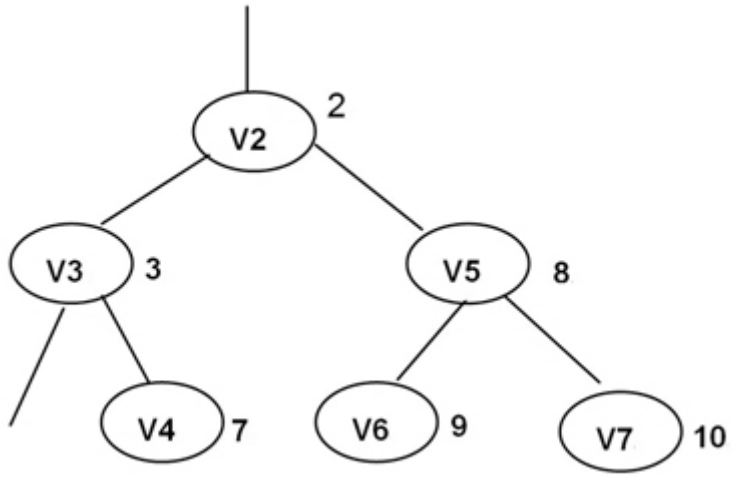
D.



评测结果：答案正确（3 分）

2-16 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

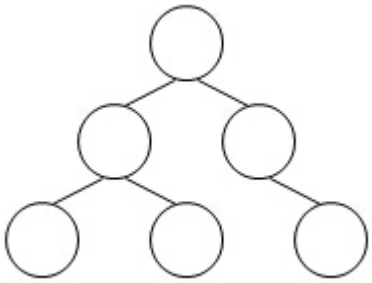
评测结果：答案正确（3 分）

2-17 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' right links are both threads? (3分)

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

评测结果：答案正确（3 分）

2-18 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`

评测结果：答案正确（3 分）

2-19 Following is the C-like pseudo code of a function that takes a Queue as an argument.

```
void foo(Queue Q)
{
    Stack S = CreateStack();  // create an empty stack

    while (!IsEmpty(Q))
    {
        // dequeue an item from Q and push it into S
        Push(S, Dequeue(Q));
    }

    while (!IsEmpty(S))
    {
        // pop an item from S and enqueue it into Q
        Enqueue(Q, Pop(S));
    }

    DisposeStack(S);
}
```

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

评测结果：答案正确（3 分）

2-20 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. $N/2$

评测结果：答案正确（3 分）

程序填空题

总分：12 得分：6

5-1 The function is to find the `K`-th largest element in a list `A` of `N` elements. The initial function call is `Qselect(A, K, 0, N-1)`. Please complete the following program.

```
ElementType Qselect( ElementType A[], int K, int Left, int Right )
{
    ElementType Pivot = A[Left];
    int L = Left, R = Right+1;

    while (1) {
        while ( A[++L] > Pivot ) ;
        while(A[--R]<Pivot)                                    (3分);
        if ( L < R ) Swap( &A[L], &A[R] );
        else break;
    }
    Swap( &A[Left], &A[R] );
    if ( K < (L-Left) )
        return Qselect(A, K, Left, R-1);
    else if ( K > (L-Left) )
        return Qselect(A,K-(L-Left),L+1,Right)                 (3分);
    else
        return Pivot;
}
```

评测结果：部分正确（3 分）

序号	结果	得分
0	答案正确	3
1	段错误	0

5-2 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+1]>H[child] (3分) ) child++;
                if ( H[child]<H[0] (3分) )
                    H[i] = H[child];
                else break;
            }
            H[i] = H[0];
        }
    }
    return H[1];
}
```

评测结果：部分正确（3 分）

序号	结果	得分
0	答案错误	0
1	答案正确	3

函数题

总分：8　得分：8

6-1 Check Topological Order

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

Format of functions:

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

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 N_v;
    int N_e;
```

```
    AdjList G;
};
typedef PtrToGNode LGraph;
```

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

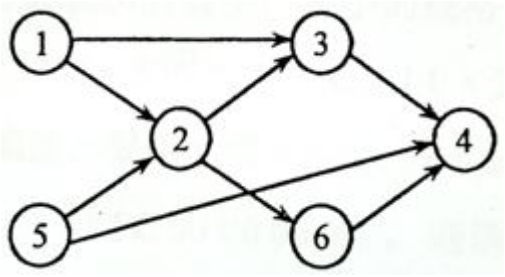
Note: Although the vertices are numbered from 1 to MaxVertexNum, they are **indexed from 0** in the LGraph structure.

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 N_v;
    int N_e;
    AdjList G;
};
typedef PtrToGNode LGraph;

LGraph ReadG(); /* details omitted */

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

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->N_v; j++)
            scanf("%d", &Seq[j]);
```

```
        if ( IsTopSeq(Seq, G)==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
```

代码

```
bool IsTopSeq( Vertex Seq[], LGraph Graph )
{
    int Indegree[MaxVertexNum] = {0};
    for (int i = 0; i < Graph->N_v; ++i)
```

```
    {
        if (Graph->G[i].FirstEdge)
        {
            PtrToAdjVNode p = Graph->G[i].FirstEdge;
            while (p)
            {
                Indegree[p->AdjV]++;
                p = p->Next;
            }
        }
    }

    for (int i = 0; i < Graph->N_v; ++i)
    {
        if (Indegree[Seq[i] - 1] == 0)
        {
            int v = Seq[i] - 1;
            PtrToAdjVNode p = Graph->G[v].FirstEdge;
            while (p)
            {
                Indegree[p->AdjV]--;
                p = p->Next;
            }
        }
        else
            return false;
    }
    return true;
}
```

评测结果：答案正确（8 分）

测试点	结果	得分	耗时	内存
0	答案正确	4	4 ms	256 KB
1	答案正确	1	4 ms	256 KB
2	答案正确	1	4 ms	256 KB
3	答案正确	1	4 ms	256 KB
4	答案正确	1	30 ms	512 KB

```
a.c: In function ‘ReadG’:
a.c:68:2: warning: ignoring return value of ‘scanf’, declared with attribute warn_unused_re
  scanf("%d", &Nv);   /* 读入顶点个数 */
  ^~~~~~~~~~~~~~~~
a.c:71:2: warning: ignoring return value of ‘scanf’, declared with attribute warn_unused_re
  scanf("%d", &(Graph->N_e));   /* 读入边数 */
  ^~~~~~~~~~~~~~~~~~~~~~~~~~
a.c:76:7: warning: ignoring return value of ‘scanf’, declared with attribute warn_unused_re
```

```
        scanf("%d %d", &E->V1, &E->V2);
        ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
a.c: In function ‘main’:
a.c:90:5: warning: ignoring return value of ‘scanf’, declared with attribute warn_unused_re
     scanf("%d", &N);
     ^~~~~~~~~~~~~~~
a.c:93:13: warning: ignoring return value of ‘scanf’, declared with attribute warn_unused_r
             scanf("%d", &Seq[j]);
             ^~~~~~~~~~~~~~~~~~~~
```