

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 絮起云微
得分 79
总分 100

判断题总分： 20 得分： 20

1-1
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)

1-2
 $(\log N)^3$ is $O(N)$. (2分)

☐ T ☒ F

评测结果： 答案正确 (2 分)

1-3
If the inorder and the postorder traversal sequences of a binary tree have exactly the same order, then none of the nodes in the tree has a right subtree. (2分)

☐ T ☒ F

评测结果： 答案正确 (2 分)

1-4
To sort N records, heap sort requires at least $O(N)$ extra space. (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 4, 5, and 6, if 4 and 6 are on the same level, then 5 must be their parent. (2分)

☐ T ☐ F

评测结果： 答案正确 (2 分)

1-6
If N numbers are stored in a doubly linked list in increasing order, then the average time complexity for binary search is $O(\log N)$. (2分)

☐ T ☐ F

评测结果： 答案正确 (2 分)

1-7
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

评测结果： 答案正确 (2 分)

1-8
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)

1-9
Mergesort is stable. (2分)

☐ T ☐ F

评测结果： 答案正确 (2 分)

1-10
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

评测结果： 答案正确 (2 分)

单选题总分： 60 得分： 45

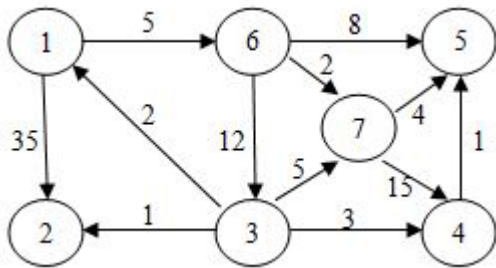
size of S must be: (3分)

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

评测结果：答案错误 (0 分)

2-4

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)

2-5

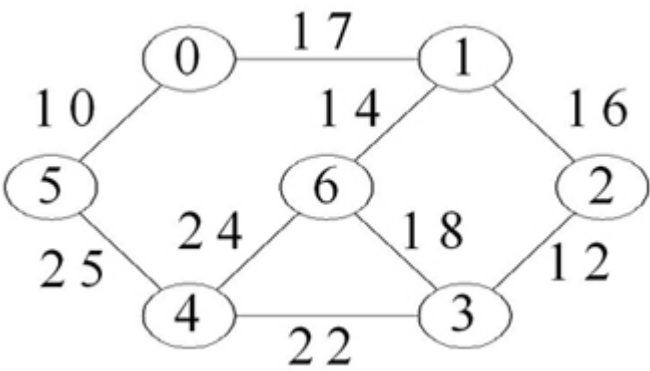
When inserting a new key **K** into a binary search tree **T** with 511 nodes, the worst-case number of comparisons between **K** and the keys already in **T** is in the range of: (3分)

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

评测结果：答案错误 (0 分)

2-6

To find the minimum spanning tree with Prim's algorithm for the following graph, a sequence of vertices 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)

2-7
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

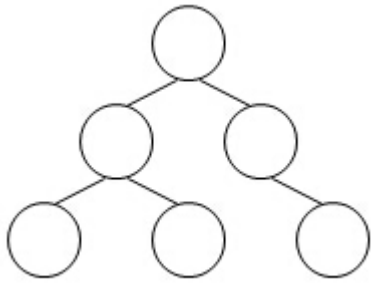
评测结果：答案正确 (3 分)

2-8
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, 28, 55, 60}. 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：答案错误 (0 分)

2-9
Given the structure of a binary search tree (as shown in the figure), which one of the following insertion sequences is impossible? (3分)

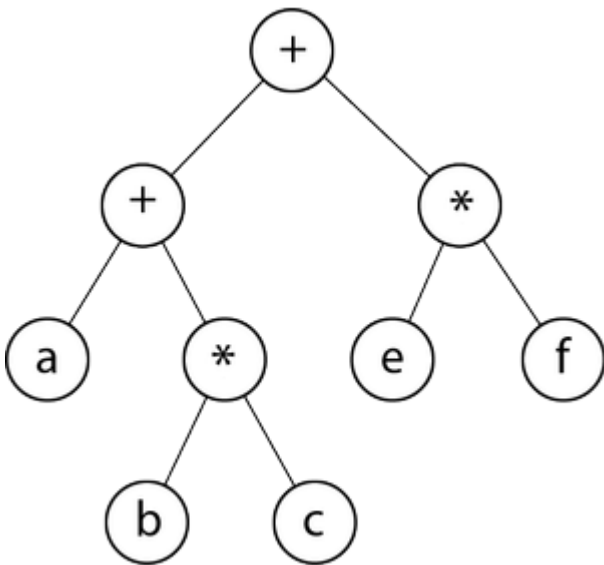


- A. 83 67 91 98 20 75
- B. 83 67 75 91 20 98
- C. 83 91 75 67 20 98
- D. 83 91 98 67 75 20

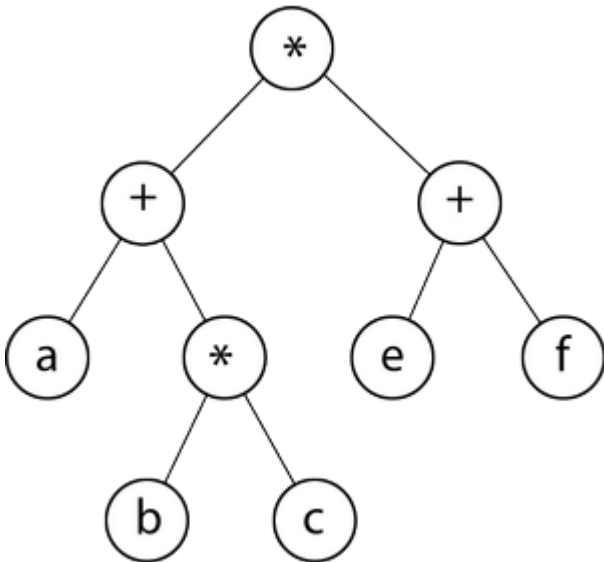
评测结果：答案正确 (3 分)

2-10

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



A.



B.

2-12

For the quicksort implementation with both the left and the right pointers stop when an element with the same key as the pivot is found during the partitioning, what is the running time when all keys are equal? (3分)



评测结果：答案正确 (3 分)

D. 2 1 4 5 3 10 7 9 8 6

评测结果：答案正确 (3 分)

- A. $O(\log N)$
- B. $O(N)$
- C. $O(N \log N)$
- D. $O(N^2)$

评测结果： 答案错误 (0 分)

2-13
Suppose that the size of a hash table is 11, and the hash function is $H(\text{key}) = \text{key} \% 11$. The following 4 elements have been inserted into the table as $\text{Addr}(14)=3$, $\text{Addr}(38)=5$, $\text{Addr}(61)=6$, $\text{Addr}(86)=9$. When open addressing with quadratic probing is used to solve collisions, the address of the element with $\text{key}=60$ will be . (3分)

- A. 5
- B. 7
- C. 8
- D. 10

评测结果： 答案错误 (0 分)

2-14
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)

2-15
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)

2-16
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)

2-17

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

- A. 35
- B. 31
- C. 33
- D. 37

评测结果： 答案正确 (3 分)

2-18

Let T be a tree created by union-by-size with N nodes, then the height of T can be . (3分)

- A. at most $\log_2(N) + 1$
- B. at least $\log_2(N) + 1$
- C. as large as N
- D. anything that is greater than 1

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** 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

评测结果：答案正确 (3 分)

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

评测结果：答案正确 (3 分)

程序填空题总分: 12 得分: 6

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

```
ElementType FindKthSmallest ( 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];
    BuildMaxHeap(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 && (3分) ) child++;

        if ( (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	答案正确	3
1	答案错误	0

5-2

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 ) ;
        (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) )
        (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

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 */
```

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

```
/*
#define MaxVertexNum 10
typedef int Vertex;

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;
*/
bool IsTopSeq( Vertex Seq[], LGraph G )
{
    int cnt[10000];
    int q[10000];
    int front = 0, rear = G->N_v - 1;
    PtrToAdjVNode p;
    for (int i = 0; i < G->N_v; i++)
    {
        cnt[i] = 0;
    }
    for (int i = 0; i < G->N_v; i++)
    {
        p = G->G[i].FirstEdge;
        while (p)
        {
            cnt[p->AdjV]++;
            p = p->Next;
        }
    }
    for (int i = 0; i < G->N_v; i++)
    {
        q[i] = Seq[i] - 1;
    }
    while (front != rear)
    {
        int temp = q[front++];
        if (cnt[temp] != 0)
            return false;
        else
        {
            p = G->G[temp].FirstEdge;
            while (p)
            {
                cnt[p->AdjV]--;
                p = p->Next;
            }
        }
    }
    return true;
}
```

评测结果：答案正确 (8 分)

```
a.c: In function 'ReadG':
a.c:68:2: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result [-Wunused-result]
    scanf("%d", &Nv);    /* 读入顶点个数 */
    ^~~~~~
a.c:71:2: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result [-Wunused-result]
    scanf("%d", &(Graph->N_e));    /* 读入边数 */
    ^~~~~~
a.c:76:7: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result [-Wunused-result]
    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_result [-Wunused-result]
    scanf("%d", &N);
    ^~~~~~
a.c:93:13: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result [-Wunused-result]
    scanf("%d", &Seq[j]);
    ^~~~~~
```