

浙江大学2015-16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 NQ-0002
得分 90
总分 100

判断题总分: 10 得分: 9

1-1
In a directed graph, the sum of the in-degrees must be equal to the sum of the out-degrees of all the vertices. (1分)

☐ T ☐ F

评测结果：答案正确 (1 分)

1-2
For a sequentially stored linear list of length N , the time complexities for deleting the first element and inserting the last element are $O(1)$ and $O(N)$, respectively. (1分)

☐ T ☐ F

评测结果：答案正确 (1 分)

1-3 The Fibonacci number sequence $\{F_N\}$ is defined as: $F_0 = 0, F_1 = 1, F_N = F_{N-1} + F_{N-2}, N=2, 3, \dots$. The time complexity of the function which calculates F_N iteratively is $\Theta(F_N)$. (1分)

T F

评测结果：答案正确 (1 分)

1-4
Shell sort is stable. (1分)

☐ T ☐ F

评测结果：答案正确 (1 分)

1-5
To sort N records by merge sort, the number of merge runs is $O(N\log N)$. (1分)

T

F

评测结果： 答案正确 (1 分)

1-6
In hashing, functions "insert" and "find" have the same time complexity. (1分)

T

F

评测结果： 答案正确 (1 分)

1-7
If a graph is represented by adjacency lists, then the space taken depends only on the number of vertices, not the number of edges. (1分)

T

F

评测结果： 答案正确 (1 分)

1-8
In a graph G, if we have to do BFS twice to visit every one of its vertices, then there must be a cycle in G. (1分)

T

F

评测结果： 答案正确 (1 分)

1-9
There exists a binary tree with 2016 nodes in total, and with 16 nodes having only one child. (1分)

T

F

评测结果： 答案错误 (0 分)

1-10
Store a complete binary tree in an array (root at position 1). Then the nodes at positions 23 and 24 are siblings. (1分)

T

F

评测结果： 答案正确 (1 分)

单选题总分： 55 得分： 55

2-1

What is a collision in hashing? (2分)

- A. Everything else are the same for the two elements except their keys
- B. Two elements have different data but share the same key
- C. Two elements with different keys share the same hash value
- D. Two elements with different hash values share the same key

评测结果： 答案正确 (2 分)

2-2

If besides finding the shortest path from `S` to every other vertices, we also need to count the number of different shortest paths, we can modify the Dijkstra algorithm in the following way: add an array `count[]` so that `count[V]` records the number of different shortest paths from `S` to `V`. Then `count[V]` shall be initialized as: (3分)

- A. `count[V]=0` for all vertices
- B. `count[V]=1` for all vertices
- C. `count[S]=0;` and `count[V]=1` for other `V`
- D. `count[S]=1;` and `count[V]=0` for other `V`

评测结果： 答案正确 (3 分)

2-3

If graph G is NOT connected and has 21 edges, then it must have at least _ vertices. (3分)

- A. 7
- B. 8
- C. 9
- D. 10

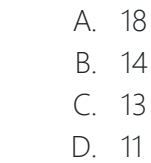
评测结果： 答案正确 (3 分)

2-4

Given the adjacency matrix of a weighted graph as shown by the figure. The total weight of its minimum spanning tree is: (2分)

∞	3	1	15	1	9
3	∞	3	13	∞	8
1	3	∞	1	1	2
15	13	1	∞	18	∞
1	∞	1	18	∞	2
9	8	2	∞	2	∞

2-5
The maximum flow in the network of the given Figure is: (3分)



2-6
If the input is a set of 10^5 1-digit numbers, which one of the following methods can sort them in $O(N)$ time? (2分)

- A. quick sort
- B. insertion sort
- C. bucket sort
- D. Shell sort

2-7 Use the linear algorithm to adjust the given sequence of numbers { 28, 12, 7, 8, 19, 20, 15, 22 } into a max-heap, and then insert 29. The resulting sequence is _____. (2分)

- A. { 29, 28, 20, 22, 12, 7, 15, 8, 19 }
B. { 29, 28, 20, 22, 19, 7, 15, 8, 12 }
C. { 29, 28, 22, 20, 19, 15, 12, 8, 7 }
D. { 7, 8, 28, 12, 19, 20, 15, 22, 29 }

评测结果：答案正确 (2 分)

2-8

Given a quadtree(四叉树) with 3 nodes of degree 2, 2 nodes of degree 3, 4 nodes of degree 4. The number of leaf nodes in this tree is . (2分)

- A. 10
B. 12
C. 20
D. 21

评测结果：答案正确 (2 分)

2-9

To check if there is a cycle in a directed graph, which one of the following algorithms can be used, besides the topological sort? (2分)

- A. Prim
- B. BFS
- C. DFS
- D. Dijkstra

评测结果：答案正确 (2 分)

2-10

BFS is similar to the traversal of a binary tree. (2分)

- A. preorder
- B. inorder
- C. postorder
- D. level-order

评测结果：答案正确 (2 分)

2-11

For a binary tree, if its pre-order travel sequence is $\{4, 2, 1, 3, 6, 5, 7\}$, and its in-order travel sequence is $\{1, 2, 3, 4, 5, 6, 7\}$, then which of the following statement is FALSE? (3分)

- A. This is a complete binary tree.
- B. All the odd numbers are at leaf nodes.
- C. 6 is the parent of 3.
- D. This is a binary search tree.

评测结果： 答案正确 (3 分)

2-12

To sort the sequence { 2, 12, 16, 88, 5, 10, 34 }, if the results of the first 2 runs are given as the following:

- after the 1st run: 2, 12, 16, 10, 5, 34, 88
- after the 2nd run: 2, 5, 10, 12, 16, 34, 88

Then the possible sorting method is: (2分)

- A. bubble sort
- B. merge sort
- C. insertion sort
- D. quick sort

评测结果： 答案正确 (2 分)

2-13

In-order traversal of a binary tree can be done iteratively. Given the stack operation sequence as the following:

```
push(1), push(2), push(3), pop(), push(4), pop(), pop(), push(5), pop(), pop(), push(6), pop()
```

Which one of the following statements is TRUE? (3分)

- A. 3 and 5 are siblings
- B. 1 is the parent of 5
- C. 6 is the root
- D. None of the above

评测结果： 答案正确 (3 分)

2-14

In a max-heap with n (> 1) elements, the array index of the minimum key may be __. (2分)

- A. 1
- B. $\lfloor n/2 \rfloor - 1$
- C. $\lfloor n/2 \rfloor$
- D. $\lfloor n/2 \rfloor + 2$

评测结果： 答案正确 (2 分)

2-15

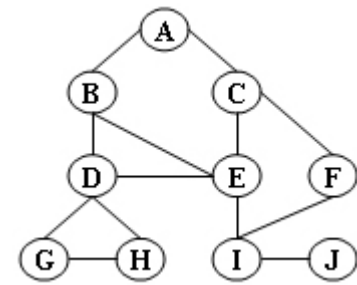
Given input { 4321, 56, 57, 46, 28, 7, 331, 33, 234, 63 }. Which one of the following is the result after the 1st run of the Least Signification Digit (LSD) radix sort? (2分)

- A. $\rightarrow 331 \rightarrow 4321 \rightarrow 33 \rightarrow 63 \rightarrow 234 \rightarrow 56 \rightarrow 46 \rightarrow 57 \rightarrow 7 \rightarrow 28$
 B. $\rightarrow 4321 \rightarrow 331 \rightarrow 33 \rightarrow 63 \rightarrow 234 \rightarrow 56 \rightarrow 46 \rightarrow 57 \rightarrow 7 \rightarrow 28$
 C. $\rightarrow 56 \rightarrow 28 \rightarrow 4321 \rightarrow 331 \rightarrow 33 \rightarrow 234 \rightarrow 46 \rightarrow 57 \rightarrow 63 \rightarrow 7$
 D. $\rightarrow 57 \rightarrow 46 \rightarrow 28 \rightarrow 7 \rightarrow 33 \rightarrow 234 \rightarrow 63 \rightarrow 56 \rightarrow 4321 \rightarrow 331$

评测结果：答案正确 (2 分)

2-16

The articulation points of the given graph are: (2分)



- A. I, D, E
B. F, E, D
C. D, I
D. E, C

评测结果：答案正确 (2 分)

2-17

Given a hash table of size 17 with the hash function $H(Key) = Key \% 17$. Quadratic probing ($h_i(k) = (H(k) \pm i^2) \% 17$) is used to resolve collisions. Then after inserting { 6, 22, 7, 26, 9, 40 } one by one into the hash table, the address of 40 is __.

(2分)

- A. 2
B. 6
C. 8
D. 15

评测结果：答案正确 (2 分)

2-18

Insert { 3, 8, 9, 1, 2, 6 } one by one into an initially empty binary search tree. The post-order traversal sequence of the resulting tree is: (3分)

- A. 2, 1, 3, 6, 9, 8
B. 1, 2, 8, 6, 9, 3
C. 2, 1, 6, 9, 8, 3
D. 1, 2, 3, 6, 9, 8

评测结果： 答案正确 (3 分)

2-19
Given the result of the 1st run of a sorting method as { 19, 21, 7, 14, 5, 27, 1, 10 }. Then among the following, this method has to be: (2分)

- A. quicksort
- B. heap sort
- C. merge sort
- D. selection sort

评测结果： 答案正确 (2 分)

2-20
Suppose that a polynomial is represented by a linked list storing its non-zero terms. Given two polynomials with N_1 and N_2 non-zero terms, and the highest exponents being M_1 and M_2 , respectively. Then the time complexity for multiplying them is: (2分)

- A. $O(M_1 \times M_2)$
- B. $O(N_1 \times N_2)$
- C. $O(M_1 + M_2)$
- D. $O(N_1 + N_2)$

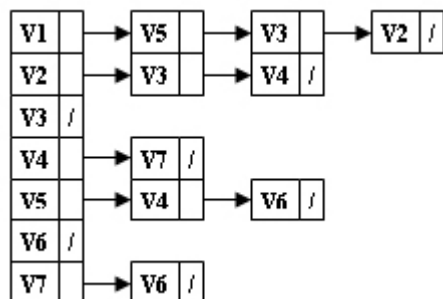
评测结果： 答案正确 (2 分)

2-21
If the pushing sequence of a stack is {1, 2, 3, ..., N } and the first number being popped out is i , then the j -th number being popped must be: (2分)

- A. $i - j - 1$
- B. $i - j$
- C. $j - i - 1$
- D. cannot be determined

评测结果： 答案正确 (2 分)

2-22
Given the adjacency lists of a directed graph as shown by the figure. Then starting from V1, a possible DFS sequence is: (2分)



- A. V1,V5,V4,V7,V6,V2,V3
B. V1,V2,V3,V4,V7,V6,V5
C. V1,V5,V4,V7,V6,V3,V2
D. V1,V5,V6,V4,V7,V2,V3

评测结果：答案正确 (2 分)

2-23

To judge an integer N (> 10) is prime or not, we need to check if it is divisible by any odd number from 3 to \sqrt{N} . The time complexity of this algorithm is . (2分)

- A. $O(N/2)$
 B. $O(\sqrt{N})$
 C. $O(\sqrt{N} \log N)$
 D. $O(0.5 \log N)$

评测结果：答案正确 (2 分)

2-24

How many of the following statements is/are TRUE? (3分)

1. If **e** is the only shortest edge in the weighted graph G , then **e** must be in the minimum spanning tree of G .
2. If the BFS sequence of a graph is **1 2 3 4 ...**, and if there is an edge between vertices 1 and 4, then there must be an edge between the vertices 1 and 3.
3. In a directed graph G with at least two vertices, if DFS from any vertex can visit every other vertices, then the topological order must NOT exist.
4. Suppose that a graph is represented by an adjacency matrix. If there exist non-zero entries in the matrix, yet all the entries below the diagonal are zeros, then this graph must be a directed graph.

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

评测结果：答案正确 (3 分)

程序填空题总分：25 得分：16

5-1

The function is to sort the list { `r[1] ... r[n]` } in non-decreasing order. Unlike selection sort which places only the minimum unsorted element in its correct position, this algorithm finds both the minimum and the maximum unsorted elements and places them into their final positions.

```
void sort( list r[], int n )
{
    int i, j, mini, maxi;

    for (i=1; i<n-i+1; i++) {
        mini = maxi = i;
        for( j=i+1; (3分); ++j ){
            if( (3分) ) mini = j;
            else if(r[j]->key > r[maxi]->key) maxi = j;
        }
        if( mini != i ) swap(&r[mini], &r[i]);
        if( maxi != n-i+1 ){
            if( (3分) ) swap(&r[mini], &r[n-i+1]);
            else swap(&r[maxi], &r[n-i+1]);
        }
    }
}
```

评测结果： 部分正确 (6 分)

序号	结果	得分
0	答案正确	3
1	答案正确	3
2	答案错误	0

5-2

Please fill the array with the results after the following union/find operations.

```
union( find(1), find(5) )
union( find(3), find(6) )
union( find(0), find(1) )
union( find(6), find(5) )
union( find(6), find(4) )
```

Note: Assume `union-by-size` (if two sets are equal-sized, the first root will be the root of the result) and `find-with-path-compression`. `S[i]` is initialized to be `-1` for all $0 \leq i \leq 7$.

A binary tree is said to be "height balanced" if both its left and right subtrees are height balanced, and the heights of its left and right subtrees can differ by at most 1. That is, $|H_L - H_R| \leq 1$ where H_L and H_R are the heights of the left and right subtrees, respectively. An empty binary tree is defined to be height balanced.

The function `IsBalanced` is to judge if a given binary tree `T` is height balanced. If the answer is yes then return `true` and store the tree height in the parameter `pHeight`, else simply return `false`. The height of an empty tree is defined to be 0.

```
typedef struct TNode *BinTree;
struct TNode{
    int Key;
    BinTree Left;
    BinTree Right;
};

bool IsBalanced ( BinTree T, int *pHeight )
{
    int LHeight, RHeight, diff;

    if( T == NULL) {
        *pHeight = 0;
        return true;
    }
    else if ( IsBalanced(T->Left, &LHeight) && IsBalanced(T->Right, &RHeight) ) {
        diff = LHeight - RHeight;
        if ( (6分) ) {
            *pHeight = 1 + ( diff>0 ? (6分) );
            return true;
        }
        else return false;
    }
    return false;
}
```

序号	结果	得分
0	答案错误	0
1	答案正确	6

函数题总分：10 得分：10

6-1

CheckBST[2]

Given a binary tree, you are supposed to tell if it is a binary search tree. If the answer is yes, try to find the K -th smallest key, else try to find the height of the tree.

Format of function:

```
int CheckBST ( BinTree T, int K );
```

where **BinTree** is defined as the following:

```
typedef struct TNode *BinTree;
struct TNode{
    int Key;
    BinTree Left;
    BinTree Right;
};
```

The function `CheckBST` is supposed to return the `K`-th smallest key if `T` is a binary search tree; or if not, return the negative height of `T` (for example, if the height is `5`, you must return `-5`).

Here the height of a leaf node is defined to be 1. **T** is not empty and all its keys are positive integers. **K** is positive and is never more than the total number of nodes in the tree.

Sample program of judge:

```
#include <stdio.h>

#include <stdlib.h>

typedef struct TNode *BinTree;

struct TNode{
    int Key;
    BinTree Left;
    BinTree Right;
};

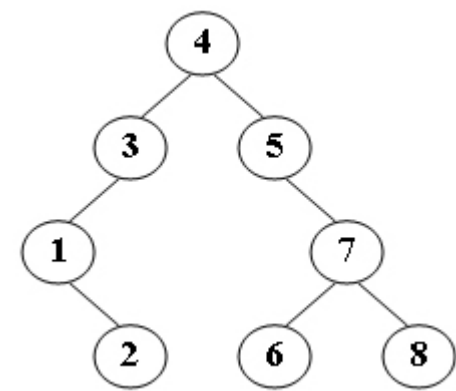
BinTree BuildTree(); /* details omitted */
int CheckBST ( BinTree T, int K );
```

```
int main()
{
    BinTree T;
    int K, out;

    T = BuildTree();
    scanf("%d", &K);
    out = CheckBST(T, K);
    if ( out < 0 )
        printf("No. Height = %d\n", -out);
    else
        printf("Yes. Key = %d\n", out);

    return 0;
}
/* 你的代码将被嵌在这里 */
```

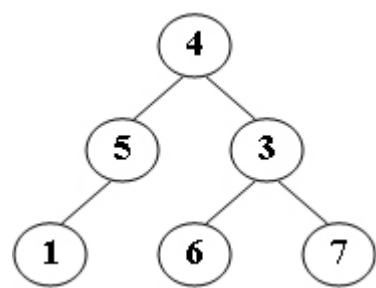
Sample Input 1: (for the following tree)



Sample Output 1:

Yes. Key = 3

Sample Input 2: (for the following tree)



Sample Output 2:

No. Height = 3

代码

```
int max(int a, int b){return a > b ? a : b;}
int GetHeight( BinTree T)
{
    if(T == NULL) return 0;
    return max(GetHeight(T->Left), GetHeight(T->Right)) + 1;
}
int num[100000], cnt;
void GetInOrder( BinTree T)
{
    if(T == NULL) return;
    GetInOrder(T->Left);
    num[cnt++] = T->Key;
    GetInOrder(T->Right);
}
int CheckBST ( BinTree T, int K )
{
    cnt = 0;GetInOrder(T);
    for(int i = 0;i < cnt-1;i++)
        if(num[i] > num[i+1])
            return -GetHeight(T);
    return num[K-1];
}
```

评测结果： 答案正确 (10 分)

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

```
a.c: In function ‘BuildTree’:
a.c:35:5: warning: ignoring return value of ‘scanf’, declared with attribute warn_unused_result [-Wunused-result]
    scanf("%d", &N);
    ^~~~~~
a.c:36:25: warning: ignoring return value of ‘scanf’, declared with attribute warn_unused_result [-Wunused-result]
    for (i=0; i<N; i++) scanf("%d", &postorder[i]);
                        ^~~~~~
```

```
a.c:37:25: warning: ignoring return value of 'scanf', declared with attribute
warn_unused_result [-Wunused-result]
```

```
for (i=0; i<N; i++) scanf("%d", &inorder[i]);
```

[illegible]

```
a.c: In function 'main':
```

```
a.c:50:5: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result [-Wunused-result]
```

```
scanf("%d", &K);
```

 $\wedge \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim \sim$