

一、判断题

R1-1 分数 1

作者 DS课程组 单位 浙江大学

If a graph is represented by adjacency lists, then the space taken depends only on the number of vertices, not the number of edges.

☐ T ☐ F

R1-2 分数 1

作者 冯雁 单位 浙江大学

In hashing, functions "insert" and "find" have the same time complexity.

☐ T ☐ F

R1-3 分数 1

作者 DS课程组 单位 浙江大学

Shell sort is stable.

☐ T ☐ F

R1-4 分数 1

作者 DS课程组 单位 浙江大学

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

☐ T ☐ F

R1-5 分数 1

作者 何钦铭 单位 浙江大学

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

☐ T ☐ F

R1-6 分数 1

作者 徐镜春 单位 浙江大学

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.

☐ T ☐ F

R1-7 分数 1

作者 DS课程组 单位 浙江大学

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

☐ T ☐ F

R1-8 分数 1

作者 徐镜春 单位 浙江大学

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 $\Theta(F_N)$.

☐ T ☐ F

R1-9 分数 1

作者 何钦铭 单位 浙江大学

There exists a binary tree with 2016 nodes in total, and with 16 nodes having only one child.

☐ T ☐ F

R1-10 分数 1

作者 DS课程组 单位 浙江大学

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.

☐ T ☐ F

二、单选题

▶ R2-1 分数 3

作者 何钦铭 单位 浙江大学

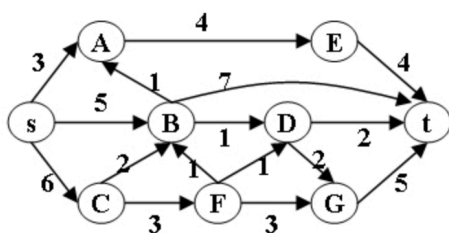
In-order traversal of a binary tree can be done iteratively. Given the stack operation sequence as the following:

```
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?

- ☐ A. 2 is the parent of 4
- ☐ B. None of the above
- ☐ C. 6 is the root
- ☐ D. 3 and 5 are siblings

The maximum flow in the network of the given Figure is:



- ☐ A. 18
- ☐ B. 14
- ☐ C. 11
- ☐ D. 13

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:

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

How many of the following statements is/are TRUE?

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 $1\ 2\ 3\ 4\ \dots$, 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. 1
- ☐ B. 2
- ☐ C. 4
- ☐ D. 3

R2-5 分数 2

作者 冯雁 单位 浙江大学

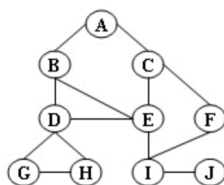
Given a hash table of size 17 with the hash function $H(Key) = Key \% 17$. Quadratic probing ($h_i(k) = (H(k) \pm i^2) \% 17$) is used to resolve collisions. Then after inserting { 23, 22, 7, 26, 9, 6 } one by one into the hash table, the address of 6 is__.

- ☐ A. 6
- ☐ B. 15
- ☐ C. 2
- ☐ D. 10

R2-6 分数 2

作者 陈越 单位 浙江大学

The articulation points of the given graph are:



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

R2-7 分数 2

作者 陈越 单位 浙江大学

Given input { 4321, 56, 57, 46, 289, 17, 331, 33, 234, 63 }. Which one of the following is the result after the 1st run of the Least Signification Digit (LSD) radix sort?

- ☐ A. $\rightarrow 4321 \rightarrow 331 \rightarrow 33 \rightarrow 63 \rightarrow 234 \rightarrow 56 \rightarrow 46 \rightarrow 57 \rightarrow 17 \rightarrow 289$
- ☐ B. $\rightarrow 57 \rightarrow 46 \rightarrow 289 \rightarrow 17 \rightarrow 33 \rightarrow 234 \rightarrow 63 \rightarrow 56 \rightarrow 4321 \rightarrow 331$
- ☐ C. $\rightarrow 331 \rightarrow 4321 \rightarrow 33 \rightarrow 63 \rightarrow 234 \rightarrow 56 \rightarrow 46 \rightarrow 57 \rightarrow 17 \rightarrow 289$
- ☐ D. $\rightarrow 56 \rightarrow 289 \rightarrow 4321 \rightarrow 331 \rightarrow 33 \rightarrow 234 \rightarrow 46 \rightarrow 57 \rightarrow 63 \rightarrow 17$

R2-8 分数 2

作者 何钦铭 单位 浙江大学

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 __.

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

R2-9 分数 2

作者 冯雁 单位 浙江大学

Use the linear algorithm to adjust the given sequence of numbers { 28, 12, 5, 8, 19, 20, 15, 22 } into a max-heap, and then insert 30. The resulting sequence is __.

- ☐ A. { 30, 28, 20, 22, 19, 5, 15, 8, 12 }
- ☐ B. { 5, 8, 28, 12, 19, 20, 15, 22, 30 }
- ☐ C. { 30, 28, 22, 20, 19, 15, 12, 8, 5 }
- ☐ D. { 30, 28, 20, 22, 12, 5, 15, 8, 19 }

R2-10 分数 2

作者 陈越 单位 浙江大学

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:

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

R2-11 分数 3

作者 何钦铭 单位 浙江大学

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?

- ☐ A. This is a binary search tree.
- ☐ B. 4 is the parent of 3.
- ☐ C. This is a complete binary tree.
- ☐ D. All the odd numbers are at leaf nodes.

R2-12 分数 2

作者 干红华 单位 浙江大学

Given an undirected graph, and the edge set of a DFS from V_0 as: $\{(V_0, V_1), (V_0, V_4), (V_1, V_2), (V_1, V_3), (V_4, V_5), (V_5, V_6)\}$. Which one the following **cannot** be the sequence of another DFS?

- ☐ A. $V_0, V_1, V_4, V_5, V_6, V_2, V_3$
- ☐ B. $V_0, V_1, V_2, V_3, V_4, V_5, V_6$
- ☐ C. $V_0, V_3, V_1, V_2, V_4, V_5, V_6$
- ☐ D. $V_0, V_5, V_4, V_6, V_1, V_2, V_3$

R2-13 分数 2

作者 陈越 单位 浙江大学

BFS is similar to the ____ traversal of a binary tree.

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

R2-14 分数 2

作者 徐镜春 单位 浙江大学

Given that the pushing sequence of a stack is $\{1, 2, \dots, n\}$ and popping sequence is $\{p_1, p_2, \dots, p_n\}$. If $p_2 = n$, how many different possible popping sequences can we obtain?

- ☐ A. $n - 1$
- ☐ B. 2
- ☐ C. 1
- ☐ D. n

R2-15 分数 2

作者 陈越 单位 浙江大学

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:

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

R2-16 分数 2

作者 冯雁 单位 浙江大学

If N keys are hashed into the same slot with separate chaining used to solve collisions, then the number of comparisons for finding these N keys is ____.

- ☐ A. $N(N - 1)/2$
 - ☐ B. $N + 1$
 - ☐ C. $N(N + 1)/2$
 - ☐ D. N
-

R2-17 分数 2

作者 DS课程组 单位 浙江大学

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

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

R2-18 分数 3

作者 陈越 单位 浙江大学

If graph G is NOT connected and has 15 edges, then it must have at least ____ vertices.

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

R2-19 分数 2

作者 徐镜春 单位 浙江大学

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:

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

R2-20 分数 2

作者 陈越 单位 浙江大学

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?

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

R2-21 分数 2

作者 冯雁 单位 浙江大学

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

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

R2-22 分数 3

作者 DS课程组 单位 浙江大学

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:

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

R2-23 分数 2

作者 徐镜春 单位 浙江大学

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 __.

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

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

0	4	10	3	2
4	0	9	5	6
10	9	0	8	7
3	5	8	0	1
2	6	7	1	0

- ☐ A. 10
- ☐ B. 14
- ☐ C. 12
- ☐ D. 11

三、程序填空题

1. Union - Find

分数 4
作者 陈越
单位 浙江大学

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

```
union( find(4), find(6) )
union( find(2), find(7) )
union( find(0), find(4) )
union( find(7), find(6) )
union( find(7), find(1) )
```

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 $0 \leq i \leq 7$.

i	0	1	2	3	4	5	6	7
s[i]	4	1 1 分	2 1 分	-1	3 1 分	-1	4	4 1 分

2. Balanced Tree [2]

分数 12
作者 何钦铭
单位 浙江大学

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;
}
```

3. Modified Selection Sort [2]

分数 9

作者 陈越

单位 浙江大学

The function is to sort the list `{ r[1] ... r[n] }` in non-increasing order. Unlike selection sort which places only the maximum 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( maxi != i ) swap(&r[maxi], &r[i]);
        if( mini != n-i+1 ){
            if( 3 分 ) swap(&r[maxi], &r[n-i+1]);
            else swap(&r[mini], &r[n-i+1]);
        }
    }
}

```

四、函数题

R6-1 CheckBST[3] 分数 10

全屏浏览题目 切换布局

作者 陈越 单位 浙江大学

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 largest key, else try to find the height of the tree.

Format of function:

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

where `BinTree` is defined as the following:

```

1 typedef struct TNode *BinTree;
2 struct TNode{
3     int Key;
4     BinTree Left;
5     BinTree Right;
6 };

```

The function `CheckBST` is supposed to return the K -th largest 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 0. `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.

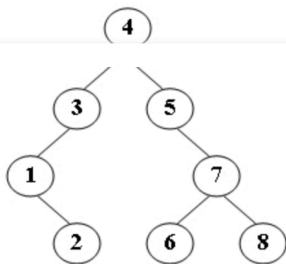
Sample program of judge:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct TNode *BinTree;
5  struct TNode{
6      int Key;
7      BinTree Left;
8      BinTree Right;
9  };
10
11 BinTree BuildTree(); /* details omitted */
12 int CheckBST ( BinTree T, int K );
13
14 int main()
15 {
16     BinTree T;
17     int K, out;
18
19     T = BuildTree();
20     scanf("%d", &K);
21     out = CheckBST(T, K);
22     if ( out < 0 )
23         printf("No. Height = %d\n", -out);
24     else
25         printf("Yes. Key = %d\n", out);
26
27     return 0;
28 }
29 /* 你的代码将被嵌在这里 */

```

Sample Input 1: (for the following tree)

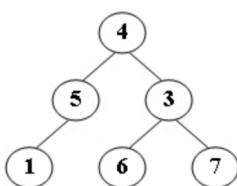


4

Sample Output 1:

Yes. Key = 5

Sample Input 2: (for the following tree)



3

Sample Output 2:

No. Height = 2