华东师范大学期末试卷（A）

2015—2016 学年第二学期

课程名称：数据结构与算法

学生姓名：__________________　　　　学号：__________________

专业：__________________　　　　年级/班级：__________________

课程性质：专业必修

一	二	三	四	五	六	七	八	总分	阅卷人签名

一、 单项选择题(15 分，每题 3 分)

1. ***p*** and ***s*** are pointer variables.***p***pointers to a node in a linear linked list. ***s*** pointers to a new node which is needed to insert after ***p***. Which is correct?

a) s->link=p; p->link=s;

b) s->link=p->link; p->link=s;

c) s->link=p->link; p=s;

d) p->link=s; s->link=p;

2. One difference between a Queue and a Stack is:

a) Queue implementations require linked lists, but stack implementations do not.

b) Stack implementations require linked lists, but queue implementations do not.

c) Queues use two ends of a linear structure, stacks use only one.

d) Stacks use two ends of a linear structure, queues use only one.

3. The average search length of binary search algorithm is:

a) O(n)

b) $O(\log_2 n)$

c) $O(n\log_2 n)$

d) O(2n)

4. The original order of the sequence is in the correct order, which one of the following sort algorithm is the best choice?

a) Insertion sort

b) Shell sort

c) Selection sort

d) Quick sort

5. A full binary tree whose height is ***h*** has ______ nodes. Provide that theheight of binary tree with one node is 0.

a) 2^h
b) 2^h-1
c) $2^{(h+1)}$
d) $2^{(h+1)}$-1

二、 填空题（15 分，每题 3 分）

1. Givethe infix expression (5+6)-2/9, please write the postfix expression.

2. For a link list, where *n* is the number of items being sorted and *k* is the number of characters in a key. The time complexity for radix sort is____________ and for merge sort is _______________.

3. Please write the output of the following program.

```
template<class List_entry>
void print(List_entry&x){
    cout<<x<<" ";
}
void main( ){
    List<int>mylist;
    for(int i=0;i<5;i++)mylist.insert(i,i);
    cout<<"Your list have "<<mylist.size()<<" elements."<<endl;
    mylist.remove(0,i);
    mylist.remove(2,i);
    mylist.insert(i,i);
    mylist.traverse(print);
    mylist.clear( );
    for(i=1;i<3;i++)mylist.insert(i, i);
    cout<<"Your list have "<<mylist.size()<<" elements."<<endl;
}
```

4. If the following function is called with a value of 2 for ***n***, what is the resulting output?

```
void Quiz( int n )
    {
        if (n > 0)
        {
            cout << 0;
            Quiz(n - 1);
            cout << 1;
            Quiz(n - 1);
        }
    }
```

21 15 31 25 29 8 24

25 21 15 31 24 29 31 29

5. For List 21, 15, 31, 25, 29, 8, and 24, use quick sort and the middle entry is the pivot. We will get the sub-lists______________ and ___________________at first iteration.

三、 简答题（48 分，每题 6 分）

1. Suppose that *q* is a queue that holds **int** type and *ss*, *se* are stacks that also hold **int**. Please write the running result of *q*, *ss* and *se* according to following code segment.

```
/*beginningof the code*/
Stack ss,se; // ss and se are stacks.
Queue q; //q is a queue

ss.push(6); ss.push(9); ss.push(8); ss.push(7);ss.push(5);

while (!ss.empty( ))
{    q.append(ss.top( ));
     ss.top( );
}

while (!q.empty( ))
{
    int x;
q.retrieve(x);
se.push(x);
     q.serve();
}
/* end of the code */
```

2. Give a binary tree **T**, the ***postorder*** traverse of T is 21,12,23,17,9,16,18,15， and ***inorder*** traverse of**T** is 21,17,23,12,15,16,9,18，What is ***preorder*** traverse? Please draw this binary tree.

3. Suppose that a hash table contains hash_size=11 entries indexed from 0 through 10. The following keys are to be mapped into the table.
   (1) Please determine the hash addresses. Hash function is %hash_size. Collision resolution method is linear probe.　K={17, 5, 28, 16, 13, 25, 61}.
   (2) Please compute the load factor of this hash table.
   (3) Please give the result of removing key 17.

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

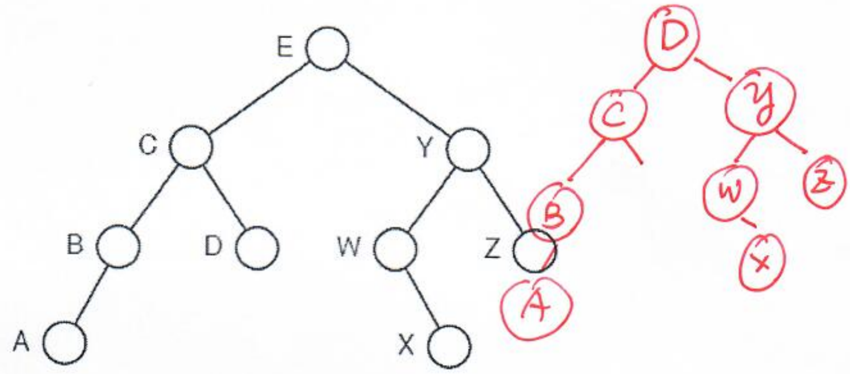
4. The initial list is (23 7 92 6 12 14 40 44 20 21),please trace the action of Build_heap algorithm in heap sort to build the initial ***heap***(大根堆).

5. ***Ackermann's*** function is defined as follows,

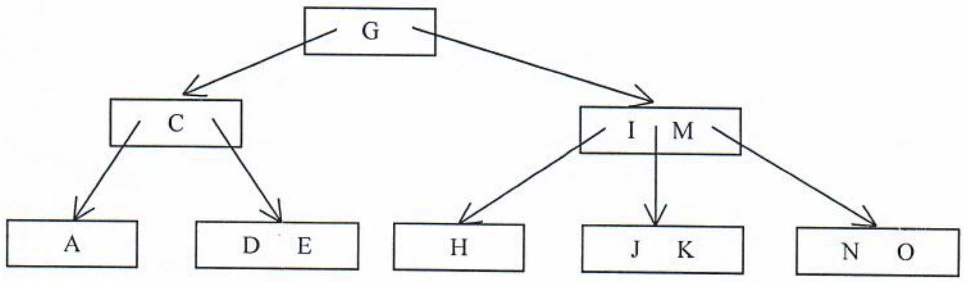
$$
\begin{array}{ll}
A(0, n) = n+1 & \text{for } n \geqslant 0 \\
A(m, 0) = A(m-1, 1) & \text{for } m>0 \\
A(m, n) = A(m-1, A(m, n-1)) & \text{for } m>0 \text{ and } n>0
\end{array}
$$

Please draw the recursion tree of ***A(2, 1)***.

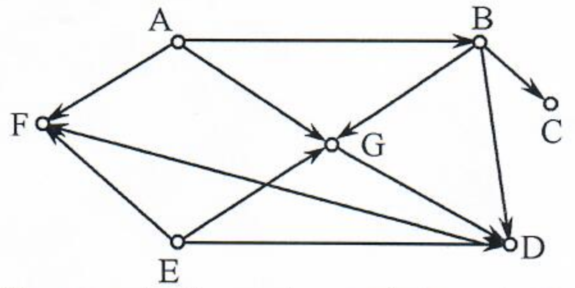
6. Starting with the following AVL tree and remove following keys (E, W) step by step.



7. Please insert the letters(B, L, P) in the order into following initial B-tree of order 3.



8. Give the graph G which is a directed graph without cycles,please write the topologicalorder of G. The depth-first order and breadth-firth order should be written separately.



The edges in the graph are <A, B>, <A, F>, <A, G>, <B, C>, <B, D>, <B, G>, <D, F>, <E, D>, <E, F>, <E, G>, <G, D>;

四、 算法与程序题（22 分，第一题 10 分，第二题 12 分）

1. （10 分）Write a C++ function to count the leaves of a linked binary tree.

```
structBinary_node {
// data members:
    int data;
    Binary_node *left;
    Binary_node *right;
// constructors:
    Binary_node( );
    Binary_node(constint&x);
};

Template <class Entry>
intBinary_tree<Entry> :: recursive_leaf_count(
                                        Binary_node<Entry> *sub_root) const
/* Post: The number of leaves in the subtree rooted at sub_root is returned. */
{
…
}
```

2. （12 分） Write a function to count the similar edge of directed graph.The edge e_i<v_i, u_i> is similar to edge e_j<v_j, u_j>, if and only if the in-degree of v_i is equal to v_j's and the in-degree of u_i is equal to u_j's.

```
typedefint Vertex;
template<intmax_size>
class Digraph {
private:
    int count; // number of vertices, at most max_size
    List<Vertex>neighbors[max_size];
public:
   //similarEdgeCount will count the number of edges in graph which are similar to
  // edge<vx, ux>.
voidsimilarEdgeCount(int& numbers/*out, The number of similar edges*/,
                      Vertex vx/*in, the start node of search edge*/,
                      Vertex ux/*in, the end node of search edge*/) const;
};
```

Following graph is an example. Suppose we call similarEdgeCount (numbers, V0,V1), and we will get numbers=3. They are <V0, V1>, <V0, V3>and <V4, V3>.