一、单项选择题（本大题共 15 小题，每小题 2 分，共 30 分）

提示：在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在下表中。错选、多选或未选均无分。

1. Given the input order of a stack is 1 2 3 … n，if the first element of output is n，then the ith element (1<=i<=n）of the output is ( ).

   A. uncertain B. n-i+1 C. i D. n-i

2. If the MaxSize of a Circular Queue is n and there is always a space not use, front points to the previous of the front element in the queue, and rear points to the rear element in the queue. The number of items in the Queue can be expressed by ( ).

   A. (rear - front + n) % n B. rear-front+1 C. rear-front-1 D. rear-front

3. A sorting algorithm is stable if it ( ).

   A. Works for all inputs.

   B. Doesn’t change the relative ordering of records with identical key values.

   C. Always sorts in the same amount of time for a given input size.

   D. Always sorts in the same amount of space for a given input size.

4. In the following sorting methods, the time **complexity of ( ) is irrelative with the** initial order of sequence.

   A. Insertion sort B. Bubble sort C. Quick sort D. Selection sort

5. According to the average case, ( ) is the quickest sorting algorithm if the size of sequence is very big.

   A. Bubble sort B. Shell sort C. merge sort D. Quick sort

6. In the following sequence, ( ) is not a heap?

   A. 100,85,98,77,80,60,82,40,20,10,66 B. 100,98,85,82,80,77,66,60,40,20,10

   C. 10,20,40,60,66,77,80,82,85,98,100 D. 100,85,40,77,80,60,66,98,82,10,20

7. The data Structures can be divided into ( ) according to their **Logical form**

   A. Dynamic structures, Static structures B. **Array-based** structures, Linked structures

   C. Simple structures, Complex structures D. Liner structures and Non-liner structures

8. In the following data structures，( ) is liner structure.

   A. DAG B. BST C. Array-based list D. General tree

9. The node number of a non-Empty Full Binary Tree must be ( ).

   A. an odd number B. an even number C. any positive number D. any integer

10. When sorting the sequence {8，9，7，15，20，-1，4}, the middle result after one pass is: {4，-1，7，8，20，15，9}; Then the sort method used is ( ).

    A. Insertion Sort B. Heap sort C. Quick sort D. Bubble Sort

11. The correct traversal to use on a BST to visit the nodes in sorted order is ( ).

    A. Preorder traversal B. Inorder traversal C. Postorder traversal D. Broadwise traversal

12. Huffman coding provides the optimal coding when ( ).

    A. The messages are in English

B. The messages are binary numbers

C. The frequency of occurrence for a letter is independent of its context within the message

D. Never

13.The primary access function used to navigate the general tree when performing UNION/FIND is (　　).

A. left child　　B. leftmost child　　C. parent　　D. right sibling

14.The sorting algorithm used as a model for most external sorting algorithms is (　　).

A. Insertion sort　　B. Quicksort　　C. Mergesort　　D. Radix Sort

15.A good hash function will (　　).

A. Use the high-order bits of the key value.　　B. Use the middle bits of the key value.

C. Use the low-order bits of the key value.　　D. Make use of all bits in the key value.

二、名词解释题（本大题共 4 小题，1–3 题每小题 4 分，4 题 3 分，共 15 分）。

提示：解释每小题所给名词的含义，若解释正确则给分，若解释错误则无分，若解释不准确或不全面，则酌情扣分。

1. stack

2. priority queue

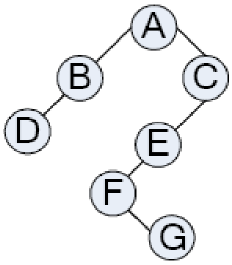
3. BFS

4. collision (in hashing)

三、应用题（本大题共 5 小题，每小题 7 分，共 35 分）

提示：有求解过程的要尽量给出解题步骤，只有最终答案会酌情扣分。

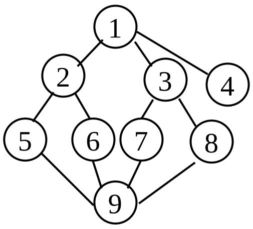
1. The binary tree is like the following. Please write the traversalling result in pre-order, in-order, and post-order respectively.



2. Starting from an empty binary tree, sequentially insert the following elements one by one according to the insertion algorithm of binary search tree: 100, 50, 302, 450, 66, 200, 30, 260.
   (a) Draw the binary search tree after inserting all the above elements.
   (b) Draw the binary search tree after deleting the element with value 302.

3. Given an array containing the elements {27, 84, 21, 47, 15, 25, 68, 35, 24}. Show how the order of the elements changes during the first pass of quicksort (choosing the first element of the array to be the pivot). Show the array after each swap.

4. Draw the adjacency list representation for the following graph, and show the order of vertices by performing DFS and BFS on the graph respectively, starting at Vertex 1.



5．Build a hash table of 2, 8, 31, 20, 19, 18, 53, 27, using the hash function H1(key)=k mod 13, using double hashing solve collisions, and the second hash function is H2(key)=(Rev(k+1) mod 11). Rev(k) reverses the decimal digits of k, for example, Rev(37) = 73; Rev(7) = 7. The size of hush table n = 13. **Be sure to indicate how** you are using H1 and H2 to do the hashing.

四、编程、设计及分析题（本大题共 2 小题，每小题 10 分，共 20 分）。

提示：每小题给出了一个程序设计要求，请按照要求填空（每空只填一条语句或表达式）或写出源程序代码，如果源程序代码中出现语法错误或逻辑错误，则酌情扣分。

1．The following is to sort data in array A[] of size n. The main idea is: at first pass, the smallest element is put in A[0], the biggest one is put in A[n-1]; at second pass, the second smallest element is put in A[1], and the second biggest one is put in A[n-2]; ……, and so on. The statement x ↔ y means exchanging the value of x and y.

```
void sort ( ElemType   A[ ],   int   n)   {
    int   min,   max,   j,   i=0;
    while (____(1)_____)     {
         min = max = i;
         for ( j = i + 1;  ____(2)______; ++j ) {
             if ( A[j] < A[min] )    min = j;
             else if ( A[j] > A[max] )    max = j;
             }
         if (___(3)______)    A[min] ↔ A[i];    //exchange the value of A[min] and A[i]
         if ( max != n-i-1 ) {
             if (____(4)___)    A[min] ↔ A[n-i-1];
             else    ____(5)_____;
             }
         i++;
         }
  }
```

(1)____________________ (2)____________________ (3)____________________

(4)____________________ (5)____________________

2．You are given a linked list **L**, and another linked list **P**, containing integers sorted in ascending order. The operation **PrintLots(L, P)** will print the elements in **L** that are in positions specified by

P. For instance, if **P** = 1, 3, 4, 6, the first, third, fourth, and sixth elements in **L** are printed. Write the function **PrintLots(L, P)**. What is the running time of your algorithm?