（装订线内不要答题）

复旦大学信息科学与工程学院

2022 ～2023 学年第一学期

《数据结构与算法》测验卷

A卷　共 10 页

专业______________________学号______________姓名_______________

提示：请同学们秉持诚实守信宗旨，谨守考试纪律，摒弃考试作弊。学生如有违反学校考试纪律的行为，学校将按《复旦大学学生纪律处分条例》规定予以严肃处理。

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

以下为试卷正文

1. *Choose the best answer to complete each sentence (每题 1.5 分，总计 15 分)*

(1) In a doubly linked list, in order to insert the node pointed to by q before the node pointed to by p, a correct implementation is ________. Note that each node has three fields (*prev, data, next*).

A． p->prev = q; q->next = p; p->prev->next = q; q->prev = q;

B． p->prev = q; p->prev->next = q; q->next = p; q->prev = p->prev;

C． q->prev = p->prev; q->next = p; p->prev = q; p->prev->next = q;

D． q->next = p; q->prev = p->prev; p->prev->next = q; p->prev = q;

(2) Suppose we implement a circular queue with an array of size 6. The current positions of *rear* and *front* are 0 and 3, respectively. After two dequeue operations, two enqueue operations, and one dequeue operation, in the order given, the positions of *rear* and *front* will become ________.

A． 5, 1　　B．2, 5　　C．2, 0　　D．4, 0

(3) Consider the operations in a stack. If we want to run push() function to the stack but there is no space to insert any element, then we call an “overflow” happens. If we want to run pop() operation to the stack but no element can be found on the top of the stack, then we call an “underflow” happens. Then, the advantage of having two stacks share a single array is ________.

A． reduction of access time and the possibility of underflow

B． reduction of storage and the possibility of overflow

C． reduction of access time and the possibility of overflow

D．reduction of storage and the possibility of underflow

(4) Suppose that a tree of degree-4 contains twenty nodes with degree-4, ten nodes with degree-3, one node with degree-2 and ten nodes with degree-1. Then, what is the number of leaf nodes in this tree?

A. 82　　B. 41　　C. 113　　D. 122

(5) Which of the following statements is wrong? ________.

A．Dijkstra's algorithm for shortest paths does not allow negative edge weights.

B．Dijkstra's algorithm allows the existence of loops in a graph.

C．Floyd's algorithm does not allow negative edge weights.

D．The depth-first search algorithm can be used to decide if a loop exists in a directed graph.

(6) Suppose that for a given binary tree, its preorder enumeration is exactly the inverse of its postorder enumeration. Then ________.

A．the tree is empty or has only one node

B．the tree has at most one leaf node

C．none of the nodes has any left child

D．none of the nodes has any right child

(7) Suppose the elements of a stack are pushed in the order 1, 2, 3, … n. We have an output sequence p_1, p_2, p_3, … p_n with p_1 = n. Then p_i is________.

A．i　　B．n-i　　C．n-i+1　　D．uncertain

(8) Recall that one application of binary trees is to represent mathematical formulas. Typically, leaf nodes represent the operands (for example, numbers or algebraic symbols), and an internal node represents an operator (for example, +, -, *) that connects the two subtrees rooted at its two children. We call the sequence of in-order traversal of all elements as the "infix expression" of this formula, and the pre-order traversal of all elements as the "prefix expression" of this formula. For a given infix expression a*(b+c)-d, its prefix expression is________.

A. abcdd+-　　B．-*a+bcd　　C．abc*+d-　　D．-+*abcd

(9) If the PostOrder and InOrder traversals of a binary tree are dfebca and dbfeac, then its PreOrder traversal is _______.

A．abdfec　　B．abdefc　　C．acbdef　　D．acbefd

(10) A directed graph G=(V, E) has V={V1,V2,V3,V4,V5,V6,V7}, E={<V1,V2>,<V1,V3>,<V1,V4>, <V2,V5>, <V3,V5>, <V3,V6>,<V4,V6>,<V5,V7>,<V6,V7>}. Then, _______ is the topological sorting sequence of G.

A．V1,V3,V4,V6,V2,V5,V7　　B．V1,V3,V2,V6,V4,V5,V7

C．V1,V3,V4,V5,V2,V6,V7　　D．V1,V2,V5,V3,V4,V6,V7

2.　Simple sorting problems (12 points in total)　排序问题（总分 12，四舍五入）

Fill in this table with the **worst-case** (最坏) asymptotic running time of each operation when using

（装订线内不要答题）

the data structure listed. Assume the following:

- Items are comparable (given two items, one is less than, equal, or greater than the other) in O(1) time.
- For insertions, it is the client's responsibility not to insert an item if there is already an equal item in the data structure (so the operations do not need to check this).
- For insertions, assume the data structure has enough room (do not include any resizing costs).
- For deletions, assume we do not use lazy deletion. (懒惰删除指的是从一个散列表中删除元素 s 时仅仅是指标记一个元素被删除，而不去整个清除它)

Operations	insert	lookup	delete	getMax
	(take an item and add it to the structure)	(take an item and return if it is in the structure)	(take an item and remove it from the structure, if present in it)	(return largest item in the structure)
sorted array				
unsorted array array				
sorted array kept organized as a min-heap (堆)				
AVL tree (自平衡二叉树)				

3. *Filling blanks for the following programs (2 point for each blank)* *填空 (每空约 2 分，总计 12 分)*

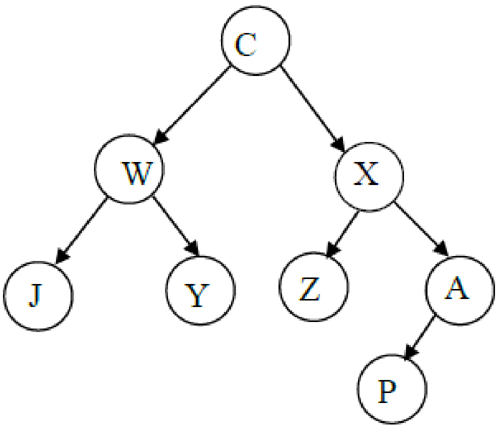
(1) *(2 points for each blank)* Please complete the following program that pushes an element x into a stack.

```
typedef struct
    {   int data[MaxSize];
        int top;                //stack top pointer
    } SqStack;                  //sequential stack

    void InitStack(SqStack *&s)
    {     s=(SqStack*) malloc(sizeof(SqStack));
          ______________________;
    }

    bool Push(SqStack *&s, int e)
```

```
{
if (s->top == ______________________)    //stack overflow
return false;
___________________;
s->data[s->top]=e;
return true;
}
```

(2) *(2 points)* When KMP algorithm is utilized for string pattern matching (串的模式匹配), what is the ***next[]*** array for a pattern substring 'abcac'? ______________ (assuming next[0] is initialized as -1)

(3) (4 *points)* Please give a Post-Order and Pre-Order traversals of the tree shown below.



Pre-order: ______________

Post-order: ______________

4. (8 points) Read the program below and answer questions

```
typedef struct LNode
{   ElemType data;
    struct LNode *next;
} LinkList;

LinkList mynote(LinkList L)
{   //L points to the first node in a singly linked list without a head node.
    if (L && L->next) {
        q = L; L = L->next; p = L;
S1:     while (p->next) p = p->next;
S2:     p->next = q; q->next = NULL;
    }
    return L;
```

```
}
```

Please answer the following questions:

1) Explain the task that the commands in line S1 complete;
2) Explain the task that the commands in line S2 complete;
3) Suppose that initially the singly linked list represents the following list of elements: (a_1, a_2, … a_n). Write down the list after an execution of the above algorithm.

（装订线内不要答题）

5. (10 points) 哈夫曼树和哈夫曼编码. One day, a mysterious ancient language was discovered by an archaeologist. It uses only the following ten symbols：! @ # $ % ^ & * ( ). By statistics, the archaeologist has found that the numbers of occurrences of these symbols in a given text are：13, 10, 45, 789, 8, 80, 33, 200, 28, 99, respectively (For example, the total number of times that the symbol "!" appeared in this given text is 13). Please build a Huffman tree so that those symbols can be represented by Haffman codes.

6. (8 points) For each of the six questions in parts (a)-(c), answer in terms of **big-O** and the number of vertices in the graph $|V|$.

(a) Suppose a graph has no edges.

i. What is the asymptotic space cost of storing the graph as an adjacency list?

ii. What is the asymptotic space cost of storing the graph as an adjacency matrix?

(b) Suppose a graph has every possible edge.

i. What is the asymptotic space cost of storing the graph as an adjacency list?

ii. What is the asymptotic space cost of storing the graph as an adjacency matrix?

(c) Suppose an undirected graph has one node A that is connected to every other node and the graph has no other edges.

i. What is the asymptotic space cost of storing the graph as an adjacency list?

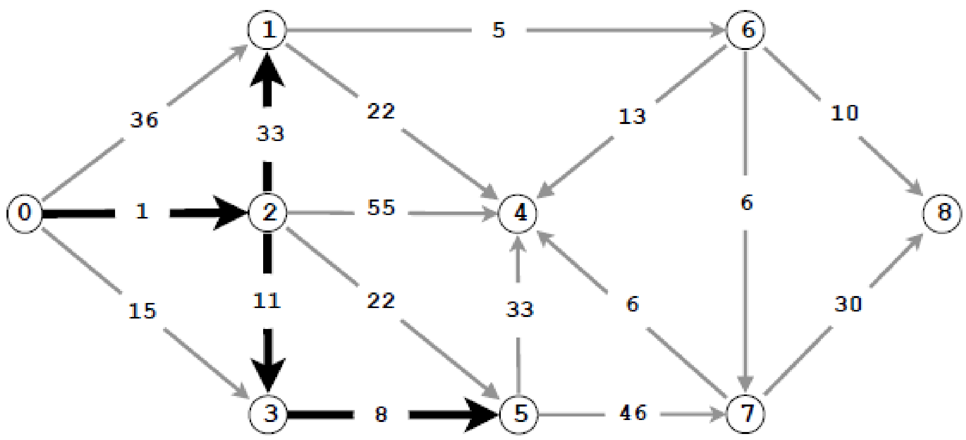
ii. What is the asymptotic space cost of storing the graph as an adjacency matrix?

(d) Is an adjacency list faster or slower than an adjacency matrix for answering queries of the form, “is edge (u, v) in the graph”?

(e) Is an adjacency list faster or slower than an adjacency matrix for answering queries of the form, “are there any directed edges with u as the source node”?

7. (10 points) Shortest-path problem. 最短路径问题 （(a)与(b)每个空格 1 分， (c)2 分）

Run Dijkstra’s algorithm on the following edge-weighted digraph, starting from vertex 0



(a) Complete the table of edgeTo[] and distTo[] values immediately after the first 5 vertices (0, 2, 3, 5, and 1) have been deleted from the priority queue and relaxed.

v	edgeTo[]	distTo[]
0	-	0.0
1	2->1 33.0	34.0
2	0->2 1.0	1.0
3	2->3 11.0	12.0
4		
5	3->5 8.0	20.0
6		
7		
8		

(b) Complete the table of edgeTo[] and distTo[] values immediately after the 6th vertex has been deleted from the priority queue and relaxed. Circle those values that changed from (a).

v	edgeTo[]	distTo[]
0	-	0.0
1	2->1 33.0	34.0
2	0->2 1.0	1.0
3	2->3 11.0	12.0
4		
5	3->5 8.0	20.0
6		
7		
8		

(c) Draw the edges in the (final) shortest-paths tree with thick lines in the figure above.

8. (8 points) Give the order-of-growth running time of each function f1–f4 as a function of N:

```
int f1(int N) {
        int sum = 0; for (int i = 0; i * i < N; i++)
        for (int j = 0; j * j < N; j++)
            for (int k = 0; k < N; k++)
                sum ++;
                return sum;
}
```

Running time: ______________

```
int f2(int N) {
        int sum = 0;
        for(int i = 0; i < N; i++)
```

```
        for(int j = 0; j < N * N; j++)
            for(int k = 0; k < j; k++)
                sum++;
                return sum;
}
```

Running time: ______________

```
int f3(int N) { return g(N, N); }

int g(int N, int K) {
    int sum = 0;
    if (N==0) return 1;
    for(int i = 1; i < K; i *= 2)
        sum += g(N-1, K)
    return sum;
}
```

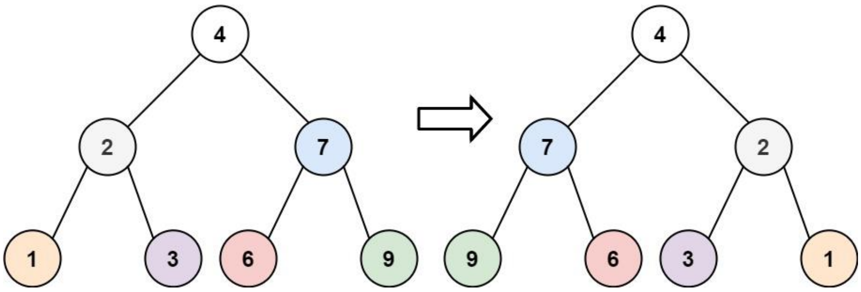
Running time: ______________

```
int f4(int N) {
    int sum = 0;
    for (int i = 0; i * i < N; i++)
        for (int j = 1; j < i; j *= 2)
            sum ++;
    return sum;
}
```

Running time: ______________

9. (10 points) Please write the function of inverting a binary tree.

Requirement: Given the root of a binary tree, invert the tree, and return *its root*.



Input: root = [4,2,7,1,3,6,9]   **Output:** [4,7,2,9,6,3,1]

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *      int val;
```

```
 *         struct TreeNode *left;
 *         struct TreeNode *right;
 * };
 */
struct TreeNode* invertTree(struct TreeNode* root){
   // Please fill in the codes. (NOT the pseudo codes)
}
```

10. (7 points) Suppose that a graph G is stored in an adjancy list. Please design an algorithm to judge whether a graph G is a tree or not. If it is a tree, return TRUE, and FALSE otherwise. (Please write down the pseudo codes).