2024 年 复旦大学姬算妓科学与技术学院

数据结构期末考试（A 卷）

整理：Gary Agasa

一、填空题（20 分，每题 2 分）

1、 下列程序段的时间复杂度是_____。

```
count = 0;
for ( k = l ; k <= n; k *= 2)
    for ( j = l ; j <= n; j ++)
        count ++;
```

2、 已知二维数组 A 按行优先方式存储，每个元素占用 1 个存储单元。若元素 A[0][0]的存储地址是 100，A[3][3]的存储地址是 220，则元素 A[5][5]的存储地址是_____。

3、 设主串 T = "abaabaabcabaabc"，模式串 S = "abaabc"，采用 KMP 算法进行模式匹配，到匹配成功时为止，在匹配过程中进行的单个字符间的比较次数是_____。

4、 若栈 S1 中保存整数，栈 S2 中保存运算符，函数 F( )依次执行下述各步操作：

(1) 从 S1 中依次弹出两个操作数 a 和 b;
(2) 从 S2 中弹出一个运算符 op;
(3) 执行相应的运算 b op a;
(4) 将运算结果压入 S1 中。

假定 S1 中的操作数依次是 5, 8, 3, 2（2 在栈顶），S2 中的运算符依次是 *, -, +（+在栈顶）。调用 3 次 F( )后，S1 栈顶保存的值是_____。

5、 将关键字 6、9、1、5、8、4、7 依次插入到初始为空的最大堆 H，得到的 H 是_____。

6、 已知字符集{a,b,c,d,e,f},若各字符出现的次数分别为 6，3，8，2，10，4，则对应字符集中各字符的哈夫曼编码可能是_____。

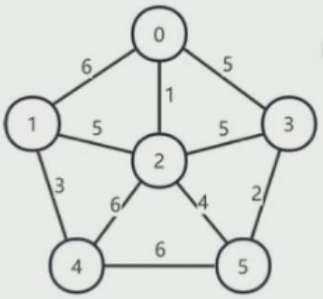
7、 设图 G = (V, E)，其中：

$V = \{V_0, V_1, V_2, V_3\}$

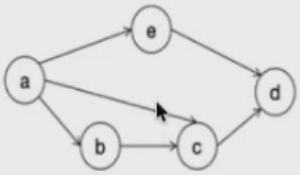
$E = \{(V_0, V_1), (V_0, V_2), (V_0, V_3), (V_1, V_3)\}$

则从顶点 V_0 开始对图 G 的深度优先遍历序列共有_____种。

8、 使用 Prim 算法从结点 0 出发求下图的最小生成树，依次写出每次被加入到最小生成树中边的编号（使用顶点序号表示）：_____。



9、 对下图进行拓扑排序，可得到的不同拓扑序列个数是_____。



10、设数组 S[ ] = {93, 946, 372, 9, 146,151, 301, 485, 236, 327, 43, 892}，采用最低位优先(LSD)基数排序将 S 排列成升序序列。第 1 趟分配、收集后，元素 372 之前、之后紧邻的元素分别是_____、_____。

二、选择题（10 分，每题 2 分）

1、 已知头指针 h 指向一个带头结点的非空单循环链表，结点结构为：

data	next

其中，next 是指向直接后继结点的指针，p 是尾指针，q 是临时指针。现要删除该链表的第一个元素，正确的语句序列是_____。

A. h->next = h->next->next; q = h->next; free(q);

B. q = h->next; h->next = h->next->next; free(q);

C. q = h->next; h->next = q->next; if (p != q) p = h; free(q);

D. q = h->next; h->next = q->next; if (p == q) p = h; free(q);

2、 已知初始为空的队列 Q 的一端仅能进行入队操作，另外一端既能进行入队操作又能进行出队操作。若 Q 的入队序列是 1, 2, 3, 4, 5, 则不能得到的出队序列是_____。

A. 5, 4, 3, 1, 2　　B. 5, 3, 1, 2, 4　　C. 4, 2, 1, 3, 5　　D. 4, 1, 3, 2, 5

3、 某森林 F 根据对应的二叉树为 T，若 T 的先序遍历序列是 a, b, d, c, e, g, f，中序遍历序列是 b, d, a, e, g, c, f，则 F 中树的棵数是_____。

A. 1　　B. 2　　C. 3　　D. 4

4、 对于无向图 G=(V, E)，下列选项中正确的是_____。

A. 当|V| > |E|时，G 一定是连通的

B. 当|V| < |E|时，G 一定是连通的

C. 当|V| = |E| - 1 时，G 一定是不连通的

D. 当|V| > |E| + 1 时，G 一定是不连通的

5、 对一组数据{2, 12, 16, 88, 5, 10}进行排序，若前 3 趟排序结果如下：

第一趟：2, 12, 16, 5, 10, 88
第二趟：2, 12, 5, 10, 16, 88
第三趟：2, 5, 10, 12, 16, 88

则该排序算法可能是_____。

A. 冒泡排序　　B. 插入排序　　C. 选择排序　　D. 二路归并排序

三、问答题（30 分，每题 5 分）

1、 阅读以下程序，回答以下问题：

(1) 运行这个程序，通过 **cout** 在屏幕上输出的内容是什么？

(2) 如果把函数声明“**int** f1(stack <**int**> &s)”改为“**int** f1(stack <**int**> s)”，程序的运行效果会怎样？

```
int f1(stack <int> &s) {
    int result = s.top( );
    s.pop( );
    if (s.empty( ))    return result;
    int t = f1(s);
    s.push(result);
    return t;
}
void f2(stack <int> &s) {
    if (s.empty( ))    return;
    int i = f1(s);
    f2(s);
    s.push(i);
}
```

2、 将关键字序列{7, 8, 30, 11, 18, 9, 14}散列存储到散列表中，散列表的存储空间是一个下标从 0 开始的一维数组，散列函数为 H(key) = (key*3) mod 7。处理冲突采用线性探查法，要求装填因子为 0.7。

(1) 请画出所构造的散列表。

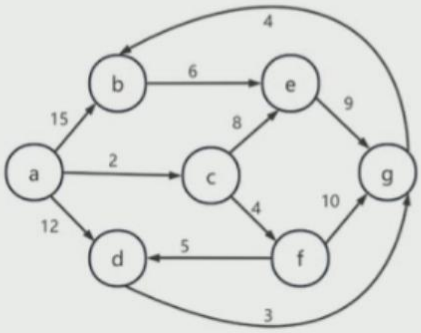
(2) 分别计算等概率情况下，查找成功和查找不成功的平均查找长度。

3、 给定一个关键码的输入序列{34, 23, 15, 98, 115, 28, 107}：

(1) 从空树开始构造 AVL 树，画出每加入一个新结点时二叉树的形态。若发生不平衡，指明需要进行的平衡旋转的类型及平衡旋转的结果。

(2) 计算该 AVL 树在等概率下搜索成功的平均搜索长度和搜索不成功的平均搜索长度。

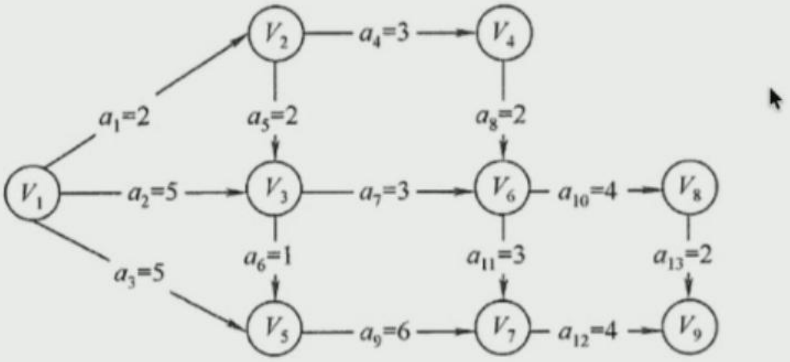
4、 写出用 Dijkstra 算法求的的下图 G 中顶点 a 到其他各顶点的最短路径，并给出执行过程表：



5、 下图所示为一个用 AOE 网表示的工程，试回答：

(1) 完成此工程至少需要多少时间？

(2) 那些活动加速可以缩短完成工程所需时间？



6、 下面的算法利用一个栈将一给定的序列从小到大排序。长度为 n 的序列由 n 个正整数组成。请补全下面的代码段，使其可以判断给定的序列是否可以利用一个栈进行排序，如果可以，输出相应的操作。

例如：序列 4 3 1 2，此序列可以排序，输出：push push push pop push pop pop pop。

```
template <class T>   //栈的元素类型为 T
class Stack {
  public:
        void clear( );                  //变为空栈
        void push(const T item);        //item 入栈
        T pop( );                       //返回栈顶内容并弹出
        T top( );                       //返回栈顶内容但不弹出
        bool isEmpty( );                //若栈已空返回真
        bool isFull( );                 //若栈已满返回真
};
Stack <int> s;   //s 为栈

int sequence[maxLen], len;   //sequence 为待排序序列，len 为序列长度
```

```
// 判断序列是否可以排序
bool islegal(){
    int i, j, k;

    // Judge if the sequence can be sorted
    for(i = 0; i < len; i++){
        for(j = i + 1; j < len; j++){
            for(k = j + 1; k < len; k++){
                /*############# TODO #################*/
                if(...)
                /*############# TODO #################*/
                    return false;
            }
        }
    }

    // print the "push" and "pop" operation
    printStackOps(s);
    return true;
}

// 输出push和pop顺序
void printStackOps(stack<int>& s) {
    /*############# TODO #################*/
    /*############# TODO #################*/
}
```

四、算法题（40 分）

（以下题目，首先用简明的文字写出算法基本思路，然后给出数据结构和算法的 C 或 C++描述。另，有关堆栈和队列的相关操作如 Push、Pop、GetTop、EnQueue、DelQueue、GetFront 等可直接调用，不需要给出具体函数实现。）

1、 如果有向图中的一个结点 a 没有出边，但从其它所有结点出发都有指向 a 的边，则称 a 为这个图的汇点(sink)。请写一个函数“**int** FindSink(**int** M[n][n], **int** n)”能够以时间复杂度 O(n)返回邻接矩阵表示的有向图的汇点在 M 中的下标，或者返回-1 表示该图没有汇点。规定图中没有自环，即形如<a, a>的边。表示有向图的邻接矩阵为 M[n][n]，M[i][j]==1 表示存在从 i 到 j 的边，M[i][j]==0 表示不存在从 i 到 j 的边。（10 分）

例 1：M = {{0, 1, 0},{0, 0, 1},{1, 0, 0}}，则 FindSink(M, 3)的返回结果为-1。

例 2：M = {{0, 1, 1},{0, 0, 1},{0, 0, 0}}，则 FindSink(M, 3)的返回结果为 2。

2、 写一个函数“ListNode * mergeKLists(ListNode **lists, **int** k)”合并 k 个有序链表，lists 是存放 k 个有序链表表头指针的数组，返回合并后的有序链表表头。要求时间复杂度为：$O(N\log_2 k)$，这里 N 是这 k 个链表的结点总数。要求空间复杂度 O(k)。（15 分）

示例:
输入:
[
  1->4->5,
  1->3->4,
  2->6
]
输出: 1->1->2->3->4->4->5->6
链表结点结构定义如下：

```
struct TreeNode {
    int val;
    TreeNode *left;
    TreeNode *right;
    TreeNode(int x) : val(x), left(nullptr), right(nullptr) { }
};
```

3、 写一个函数“**int** countNodes(TreeNode *root)”根据给定的完全二叉树的根结点指针 root 统计树中的结点数，作为函数的返回值。要求时间复杂度为 O(logN)，N 为二叉树的结点数。（15 分）

提示：二叉树结点结构定义如下：

```
struct TreeNode {
    int val;
    TreeNode *left;
    TreeNode *right;
    TreeNode(int x) : val(x), left(nullptr), right(nullptr) { }
};
```