

北京大学信息科学技术学院本科期末考试

试卷规范及阅卷要求

按照学校教务部的有关规定，为了加强对试卷与成绩档案的科学管理，特作如下要求：

1. 请按学院统一的试题格式排版。
2. 在每大题的题号后面括号内标明该题的分数值，见后面的模版。
3. 试题要认真检查、核对，并经过解答验证，确保无误。
4. 试题文字一律用计算机录入，图表可以粘贴，但是要保证图表清晰。
5. 应在考试前一周把试题送本科教学副院长审批，然后把打印好的试题交学院教务办公室开印刷单，到教材中心印制。单面印刷，试题按份装订，**答题时不允许学生擅自拆掉装订。**
6. 做好试题保密工作，不要把试题存在联网的计算机内。
7. 如需答题纸和草稿纸，请到信息学院本科教务办领取。
8. 如果考试中出现违纪或作弊问题，应在考场记录中如实填写，并在考试后立即把考场记录和作弊的证据等材料交到开课院系教务办公室。
9. 主讲教师要亲自参加阅卷，一律用红笔给分，按小题或答题要点分别给出得分（不能得分和扣分混用），保证所有试题得分相加与总分对应。卷面上成绩改动处要签名。如果多个教师分别负责不同题目的阅卷，请在分数栏的相应题号下签名，总分下由课程主讲教师签名；如果一个教师独立阅卷，可以只在总分下签名。
10. 答题纸上只能给出卷面成绩，不要加权或加平时成绩，平时成绩写在成绩单上，其他加分或奖励分都包含在平时成绩中。
11. 成绩单中平时成绩、期中考试和期末考试成绩都用百分制，要标明各部分成绩所占百分比。如果没有期中考试，那一栏不填。各门课的总成绩中，一般优秀率（85 分以上）不超过 40%，不及格率（60 分以下）不超过 10%。
12. 考试后一周内完成阅卷工作，教师如有特殊原因不能按时完成阅卷，必须写出书面报告说明原因以及交成绩单的时间，由学院主管教学的副院长审批签字后送教务办公室备案。
13. 完成阅卷工作后任课教师把空白试卷、考场纪录、成绩单（从成绩系统打印，请在每页签字）、试卷参考答案、试卷分析以及全部考生答题纸（按成绩单排序）交到学院本科教务办公室。教务员应认真检查教师上交的材料，如果材料不全或不符合要求不予接受，验收合格后进行登记，由任课教师和教务员共同签字。
14. 试卷格式模版如下：

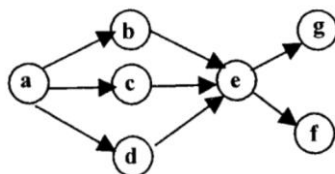
北京大学信息科学技术学院试题

考试科目: 数据结构与算法 A 姓名: _____ 学号: _____

考试时间: 2021 年 12 月 29 日 任课教师: _____

一. 单选、多选或填空题 (10 小题, 每空 1 分, 共 10 分)

1. 如下图所示的有向无环图, 可以排出 _____ 种不同的拓扑序列。



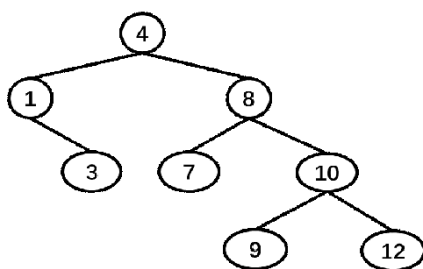
2. 对于一个包含 N ($N > 1$) 个顶点的图, 假定任意两点间最多只有一条边, 那么下列哪些是错误的 (_____)。
- A. 在深度优先遍历稠密图时, 用邻接表存储图比用相邻矩阵存储图不一定具有更高的效率。
 - B. 单源最短路径问题中, 当路径中存在负权边时 (不存在负权回路) 不可以使用 Dijkstra 算法, 可以先在每条边上加上一个常数, 从而消除负权边, 然后再利用 Dijkstra 算法得到单源最短路径。
 - C. 考虑单源最短路径问题, 在 Dijkstra 算法中可以利用最小堆来寻找权值最小结点, 也可以不利用堆 (即直接扫描距离数组) 来寻找权值最小结点。对于稠密图而言, 前者总是比后者在单源最短路径搜索中的复杂性要低。
 - D. 最小生成树问题中, Kruskal 算法适用于稀疏图而 Prim 算法适用于稠密图。
3. 设用希尔排序对数组 {98, 36, -9, 0, 47, 23, 1, 8, 10, 7} 进行从小到大排序。初始步长为 4, 其后每次步长折半, 则需调用 _____ 次插入排序; 第一趟希尔排序结束后, 数据的排列次序为 _____ 。
4. 下面的排序算法中, 不稳定的是 (_____)
- | | | |
|--------|--------|--------|
| A 冒泡排序 | B 插入排序 | C 快速排序 |
| D 希尔排序 | E 基数排序 | F 堆排序 |
5. 设依次输入降序关键码序列 $k_1 > k_2 > \dots > k_n$, 缓冲区大小为 m 。在 $n=250, m=25$ 的情况下, 用最小值堆进行置换-选择排序, 可产生初始归并段的个数为 _____; 如果 $k_1 < k_2 < \dots < k_n$, 用最小值堆进行置换-选择排序, 可产生初始归并段的个数为 _____。
6. 设磁盘块大小是 1024 字节, 每个索引项需要 8 个字节。一个包含 25000 条记录的数据文件, 若采用二级索引, 则一级索引文件和二级索引文件共需占用 _____ 磁盘块。
7. $A[64][64]$ 是一个对称矩阵 (即 $A[i][j] = A[j][i], 0 \leq i, j < 64$), 现将其非零元素按行存储到一维数组 T 中, 并尽量节约空间, 那么所需要的数组长度是 _____; 假设存储下三角的元素, 那么 $A[21][20]$ 对应的下标是 _____。

二. 简答辨析题（6 小题，共 16 分）

1. 【3 分】由点集 $\{0, 1, 2, 3, 4\}$ 组成的有向图，其边权值由以下矩阵给出：

$$W = \begin{matrix} & \begin{matrix} 0 & 3 & 0 & 9 & 0 \end{matrix} \\ \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 21 \end{matrix} & \begin{matrix} 0 & 0 & 18 & 0 & 4 \\ 0 & 0 & 0 & 6 & 0 \\ 0 & 7 & 0 & 0 & 11 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \end{matrix}$$

- (1) 请画出从结点 0 开始进行广度优先搜索得到的树。（注意：在同等条件下先走标号较小的结点；标注结点的标号和边的权值）
 - (2) 设图中的每个结点 v 与一个整数区间 $I_v = [i, j]$ 关联，其中 j 为结点 v 在(1)中得到的树的后序遍历中的序号， i 为在(1)中得到的树中以结点 v 为根的子树中最小的后序遍历序号。请写出每个结点对应的区间。（注意：后序遍历的序号从 1 开始）
 - (3) 给定一个有向图 and 一棵包含其所有结点的广度优先搜索树，问 $I_u \supseteq I_v$ ($u \neq v$) 是否为该有向图中结点 u 与结点 v 之间有路径的充分条件？若是，请简要说明理由；若否，请在本题的图中给出一个反例。
2. 【2 分】 $n=10000$ 个整数，要求使用基于比较的方法，找到最大的前 $k=10$ 个数，并且对这 k 个数进行排序，其他数的顺序不作要求。请给出在以下场景中所需的最大比较次数，并给出计算依据。
- (1) 对 n 个数进行排序，排序过程中若发现已经得到了排好序的最大的前 k 个数，则输出并终止算法。这里使用某种能使该算法终止时所需比较次数最小的基于比较的非递归排序算法。
 - (2) 对 n 个数建立最大堆，调用 k 次 `pop_max()` 函数（`pop_max()` 函数返回堆的最大元素，并将该元素从最大堆中删除。）注：本题写出表达式即可。
3. 【2 分】将关键字序列（7、8、30、11、18、9、14）散列存储到散列表中。散列表是一个下标从 0 开始的一维数组，散列函数为： $H(\text{key}) = (\text{key} * 5) \text{ MOD } 7$ ，处理冲突采用二次探测法，要求装填（载）因子为 0.7。
- (1) 请画出所构造的散列表。
 - (2) 计算等概率情况下查找成功和查找失败的平均查找长度。
4. 【2 分】对于一棵包括 n 个关键码的 m 阶 B-树，请问在一次检索中最多涉及多少个结点？最少涉及多少个结点？（要求写出推导过程）
5. 【2 分】请将关键码 13、11、3、7、5、2，10，9 依次插入一棵初始为空的红黑树，请画出每一步操作后的结果。（黑色结点用圆圈○表示，红色结点用方框□表示，不用画出外部空叶）
6. 【2 分】对下图所示的 AVL 树，依次插入关键码为 2，11 的两个结点，画出每次插入后得到的 AVL 树并指出如何旋转。



7. 【3 分】假设广义表 $P = (L_1: (a, b, L_2), L_2: (L_1, b, c), L_3: (c, d, L_2))$

- (1) 画出图形表示;
- (2) 该表的深度是多少?
- (3) 使用头尾(Head(), Tail())运算找出 d 元素。

三. 算法填空题 (每空 1 分, 共 5 分)

1. 给定长度为 N 的待排序整数数组 `array`, 采用冒泡排序将数组从小到大排列时, 必然需要进行数组元素交换。请完善下面的代码, 使 `count_min_swap` 函数能够返回冒泡排序过程中, 所需的最少交换次数。

```
int array[N];
int temp[N];
int count_swap_left_right (int* array, int left, int right) {
    int mid = (left + right) / 2;
    int result = 0;
    for (int i = left; i <= mid; i++) {
        temp[i] = array[i];
    }
    for (int j = 1; j <= right - mid; j++) {
        temp[right - j + 1] = array[j + mid];
    }

    int index1 = left, index2 = right, k = left;
    for (; k <= right; k++) {
        if (_____) { (1)
            array[k] = temp[index1++];
        }
        else { (2)
            _____ (3)
        }
    }
    return result;
}

int count_min_swap(int* array, int left, int right) {
    if (left == right || array == NULL) return 0;
    int mid = (left + right) / 2;
    int l_result = count_min_swap(array, left, mid);
    int r_result = count_min_swap(array, mid + 1, right);
    int lr_result = _____ (4)
    return _____ (5)
}
```

四. 设计分析题（2 小题，共 9 分）

注：本部分答题应申明算法思想，进行必要的算法复杂性分析，代码加以恰当注释。

1. 【5 分】已知带权无向连通图 $G=(V, E)$ 为稀疏图， $|V|=n$ ， $|E|=m$ ， $m \ll n^2$ ，每条边的权重 $w \in \{1, 2\}$ ，使用邻接表存储。给定 G 中不同的两点 s 和 t ，它们之间存在一条权重和最短的路径 p ，请设计一个算法，用尽可能低的时间复杂度（不考虑空间复杂度），找到 p 上的权重之和（不需要找到 p ），并给出时间复杂度分析。

注：所给出算法的复杂度尽可能低；只需要给出伪代码或算法流程即可，不需要给出完整的代码。

2. 【4 分】假设你正在做一个炫酷的文本编辑器，它不仅具有插入和删除功能，还具有翻转功能，描述如下：

(1) *delete*(m, n): 删除 $[m, n]$ 区间的字符， m 与 n 为字符串下标；

(2) *rotate*(m, n): 翻转 $[m, n]$ 区间的字符， m 与 n 为字符串下标；

请使用伸展树完成上述操作。（注：只需要描述算法的思想以及实现思路，也可以用伪代码来展示）

伸展树是一种自平衡的 BST，数据结构如下，也可以根据需要自行添加 `TreeNode` 的成员变量。

```
struct TreeNode {
    int key;           //表示当前节点在数组中的下标;
    int value;         //表示当前节点的字符值，即字符的 ASCII 码
    TreeNode * father,* left,* right;
};
```

可以直接使用的函数包括：

```
void Splay(TreeNode* x, TreeNode* f); //将  $x$  旋为  $f$  的子结点， $f$  为  $x$  的祖先
                                     //把  $x$  旋到根结点即 Splay(x, NULL)
void Delete(TreeNode* x);           //删除以  $x$  结点为根的子树；
```

五. 期中成绩（30 分）

六. 上机成绩（30 分）