

信息科学技术学院 2006-2007 学年第一学期

本科生期末考试试卷

考试科目：数据结构与算法 A 考试时间：2007 年 1 月 8 日

_____专业_____级_____班 考场_____室

姓名_____ 学号_____ 教师：张铭、赵海燕、王腾蛟

题号	一	二	三	四	五	总分
得分						

注意事项：

1. 第一题填空答案请写在试卷的留白上，其他题目都写在空白答题纸上。若超出所留空间，请使用后面的续页。
2. 本试卷对于算法改进、算法设计都有质量要求，请尽量按照试题中的要求来写算法。否则将酌情扣分。
3. 您所写的算法应该申明算法思想，并在算法段加以恰当的注释。

诚实考试宣言：我真诚地保证严格遵循考场纪律，诚实地用自己所掌握的知识 and 能力独立地回答试卷。不携带试卷出考场，也不会把考题以泄露出去。

保证人：_____

不在“诚实考试宣言”之后签名，则计零分或根据作弊与否的情况上报学校处理！

一、填空（28 分）

1. （3 分） n ($n > 0$) 个顶点的无向图最多有_____条边， n ($n > 0$) 个顶点的连通无向图至少有_____条边。
2. （5 分）设图 G 的邻接矩阵 A 如右图所示（顶点依次是 V_0, V_1, V_2, V_3, V_4 ），则 G 中出度最大的顶点是_____，入度最大的顶点是_____，并请给出 G 的所有拓扑序列_____。

$$A = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{vmatrix}$$

3. (3分) 对字符串数组进行快速排序, 第一趟选 ff 为轴值, 分割后得到的序列是()。
- A. [da, ax, eb, de, bb] ff [ha, gc]
 B. [cd, eb, ax, da] ff [ha, gc, bb]
 C. [gc, ax, eb, cd, bb] ff [da, ha]
 D. [ax, bb, cd, da] ff [eb, gc, ha]
4. (7分) 散列表长度 $m=13$, 散列函数为 $h_1(\text{key}) = \text{key} \% m$ 。处理冲突的方法为双散列法, 探查函数为:

$$p(\text{key}, i) = (\text{key} \% (m-2) + i * h_1(\text{key})) \% m, i=1, \dots, m-1。$$

依次将关键码{34, 96, 12, 8, 37, 35, 22} 填入散列表。

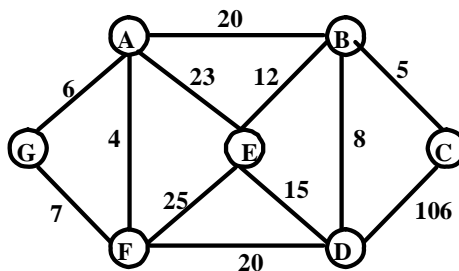
- (1) (3分) 请把关键码填写到下面散列表中相应的位置(散列槽):

0	1	2	3	4	5	6	7	8	9	10	11	12

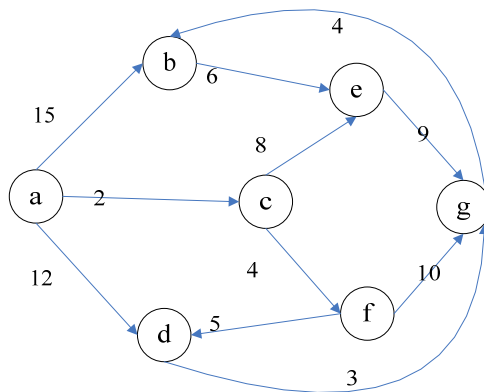
- (2) (4分) 对散列表中现有元素等概率查找时查找成功的平均查找长度是_____。
5. (3分) 外排序是针对_____所进行的排序操作, 其算法的主要目标是_____。
6. (3分) 在一个二级索引中, 每一条二级线性索引记录对应于一个一级线性索引文件的磁盘块。磁盘块的大小是 1024 字节, 每个(关键码, 指针)对的记录共需要 8 个字节。假设线性文件索引中包含 10000 条记录, 那么相应的二级线性索引文件中有_____项记录。
7. (4分) 设某文件经内排序后得到 100 个初始顺串, 若使用多路归并排序算法, 并要求三趟归并完成排序, 问归并路数最少为_____。

二、辨析题 (26 分)

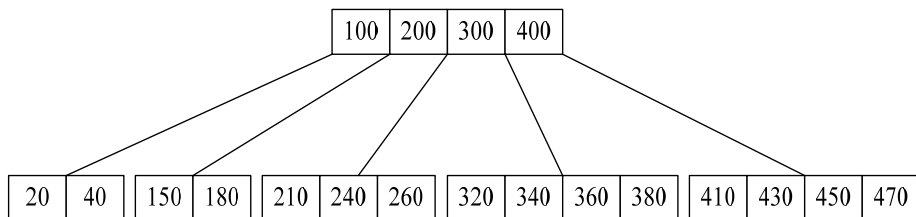
1. (6分) 写出 Prim 算法构造右图所示图 G 的一棵最小生成树的过程(从顶点 A 开始), 并且画出最后的最小生成树。



2. (8分) 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, 并给出执行过程中 dist 数组的变化情况:



3. (4 分) 请简述置换选择排序的基本思想。
4. (8 分) 请画出往下图的 5 阶 B 树中插入一个关键码 390 后得到的 B 树，以及再删除关键码 150 后得到的 B 树。并请分别分析插入和删除时的读写页块次数。（假设删除关键码时，如果需要从兄弟结点借关键码的话，先查看左边的、再查看右边的兄弟。在操作之前 B 树所有的结点页块均在外存，在插入和删除过程中，新近被访问过的外存磁盘块都被缓冲在内存缓冲区中，不必重新访外）



三、算法填空（16 分）

在下面算法段的空格中填写表达式、C++程序段（语句）或函数参数，完成下面增量为 3 的 shell 排序算法。

```
// Shell 排序，Array[]为待排序数组，n 为数组长度
template <class Record,class int_intCompare>
void ShellSorter<Record,int_intCompare>::delta3Sort(Record Array[], int n)
{
    for (int delta=n/3; delta>=3; ①_____ ) // 增量 delta 递减
        for (int j=0; j<delta; j++) // 分别对 delta 个子序列进行插入排序
            ModifiedInsertSort(②_____);
    ③_____ ;
}
```

// 针对变化的增量而修改的插入排序算法，参数 delta 表示当前的增量

```
template <class Record,class Compare>
void ShellSorter<Record,Compare>::ModifiedInsertSort(Record Array[],int n, int delta)
{
    for (int i=delta; i<n; i+=delta)           // 对子序列中第 i 个记录排序
        for (int j=i; j>=delta; j-=delta)
        {
            if (Compare::lt(Array[j], Array[j-delta]))
                ④ _____;
            else break;
        }
}
```

四、算法设计（30 分）

1. （15 分）有向图 $G = (V,E)$ 中，假设 v 是图 G 的一个顶点，把 v 的偏心度定义为：

$$\max\{dist(w,v), \forall w \in V(G)\}$$

其中 $dist(w,v)$ 是从顶点 w 到顶点 v 的最短距离。如果 v 是有向图 G 中具有最小偏心度的顶点，则称顶点 v 是 G 的中心点。对于任意的 $e \in E$ 都有各自的权。
请设计一个求图 G 的中心点的算法。

2. （15 分）有一种简单的排序算法，叫做计数排序（rank sorting），该算法用一个新数组存储原数组各元素排序后的正确下标位置（结果下标）。假设所有待排序的关键码互不相同，算法对每个记录都扫描一趟待排序数组，统计表中有多少个记录的关键码比该记录的关键码小。假设针对某一个记录，统计出的计数值为 c ，那么，这个记录在新有序表中的合适的存放位置即为 c 。例如，对于排序码{34, 96, 12, 8, 37}，计数排序后的结果如下表所示。

下标	0	1	2	3	4
排序码	34	96	12	8	37
结果下标	2	4	1	0	3

得到这个结果下标后，还需要一个整理的过程，在原来传入待排数据的数组 `array` 中返回排序后的正确结果，而且把计数结果在 `rank` 数组中返回。对于上例，按照排序码的顺序依次为 { 8, 12, 34, 37, 96}。

- (1) 编写实现计数排序的算法；函数原型为：
void Rank(Record* array, int *rank, int n);
(2) 与简单选择排序相比较，这种方法是否更好？为什么？

注释：待排序的数据类型 `Record` 是 “（key, info）” 对。不能简单认为待排数据就是整型。