北京大学信息科学技术学院考试试卷

考试科目：数据结构与算法 A **姓名：**_____ **学号：**_____

考试时间： 2017 年 1 月 4 日 **任课教师：**_____

题号	一 11 分	二 15 分	三 5 分	四 9 分	五 25 分	六 35 分	**总分**
分数							
阅卷人							

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机等）不得带入座位，已经带入考场的必须放在监考人员指定的位置，并关闭手机等一切电子设备。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束监考人员宣布收卷时，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准旁窥、交头接耳、打暗号，不准携带与考试内容相关的材料参加考试，不准抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有严重违纪或作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学习纪律管理规定》及其他相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共　　页。

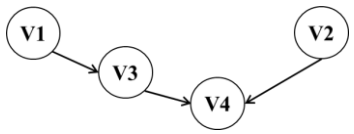
装订线内

不要答题

得分

一、选择填空题（每空 1 分，共 11 分）（答案写在答题纸上）

1. G 是一个非连通无向图，共有 21 条边，则图 G 至少有__8__个顶点。
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，那么下列哪些情况是**错误**的____AB____。
   A. 如果是有向图，则其任何一个极大强连通子图都无法进行拓扑排序。
   B. 如果是无向连通图，则其最小生成树一定不包括权重最大的边。
   C. 如果是无向连通图，假设所有边的权重均为正值，Dijkstra 算法给出的生成树不一定是最小生成树，但是与该图的任何一个最小生成树都至少有一条相同边。
3. 有向图 G 如下图所示：



（1）写出所有可能的拓扑序列：__1234、1324、2134__。
（2）若要使该图只有惟一的拓扑序列，则可以添加一条弧__v2v1 或 v3v2__。

4. 在快速排序中，定义一次平分的划分为 “幸运的划分”，而一次划分如果有一边为空则是“不幸的划分”。假设划分的过程总是“幸运”和“不幸”交替的，则该快速排序的时间复杂性为_____nlogn_____。
5. 具有 10000 个关键码的 16 阶 B 树的查找路径长度（从根到叶节点访问 B 树索引块的次数）不会小于___3___。
6. 设有 8 个初始归并段，其长度分别为 32，46，56，64，20，87，70，40；进行 3 路归并排序，所构造的最佳归并树对应的总读写次数为__1590___。
7. A[N][N]是对称矩阵，现将下三角矩阵按行存储到一维数组 T[N(N+1)/2]中（包括对角线），则对任一上三角元素 A[i][j]其对应值（0 <= i <= j < N）在 T[k]中的下标 k 是_____j(j+1)/2+i_____。
8. 在一棵空 AVL 树中，顺序插入如下关键码：{5, 9, 4, 2, 1, 3, 8}，请问全部插入后，在等概率下查找成功的平均检索长度为__17/7_____。
9. 已知广义表 C=(c, (d, A), B, e)，则广义表 C 的深度为___2_____，tail(head(tail(C)))的运算结果为__(A)_____。

得分

二、简答辨析题（每题 3 分，共 15 分）

1. 如果要找出一个具有 n 个元素集合中的第 k (1≤k≤n)个最小元素，所学过的排序方法中哪种最适合？给出实现的基本思想。

   答案：在具有 n 个元素的集合中找第 k 个最小元素，应使用快速排序算法。其基本思想如下：设 n 个元素的集合用一维数组表示，其第一个元素的下标为 1，最后一个元素下标为 n。以第一个元素为“枢轴”，经过快速排序的一次划分，找到“枢轴”的位置 i，若 i = k，则该位置的元素下标为 n；若 i > k 则在 1 至 i-1 间继续进行快速排序的划分；若 i < k 则在 i+1 至 n 间继续进行快速排序的划分。这种划分一直进行到 i = k 为止，第 i 位置上的元素就是第 k 个最小元素。

2. 已知一组关键码为（26，36，41，38，44，15，68，12，06，51，25），散列表长度为 15，用线性探查法解决冲突构造这组关键码的散列表。散列函数为：h(k)= k %13。请回答：

   1) 构造顺序插入上述关键码集合后的散列表；

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
26	25	41	15	68	44	6				36		38	12	51

   2) 下一记录放到第 11 个槽和第 7 个槽中的概率分别是多少？

   下一条记录放在第 11 个槽中的概率是 2/13

   放到第 7 个槽中的概率是 9/13

   3) 查找成功和失败情形下的平均查找长度 ASL 分别是多少？

$$ASL_{succ} = \frac{1}{11}\sum_{i=1}^{11} C_i = \frac{1}{11}(1*6+2+2+2+3+5) = \frac{20}{11}$$

$$ASL_{unsucc} = \frac{8+7+6+5+4+3+2+1+1+1+2+1+11}{13}$$

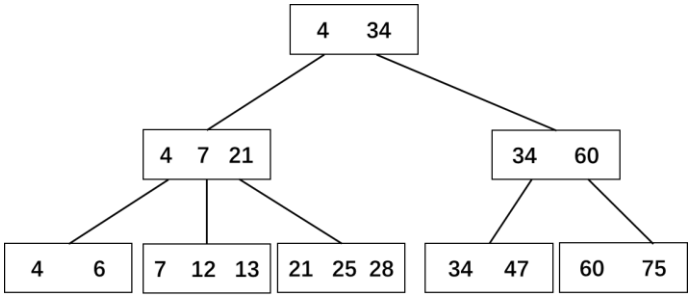
$$= \frac{52}{13} = 4$$

3. 有如下图所示的一个 3 阶 B+树，请完成如下题目：

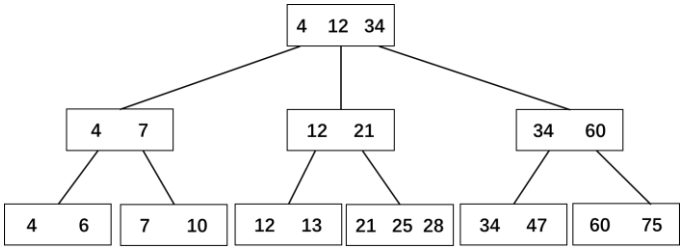
   1) 分别画出依次插入关键码 10 和删除关键码 4 的 B+树；

   2) 分析上述操作过程中的访外读写次数。

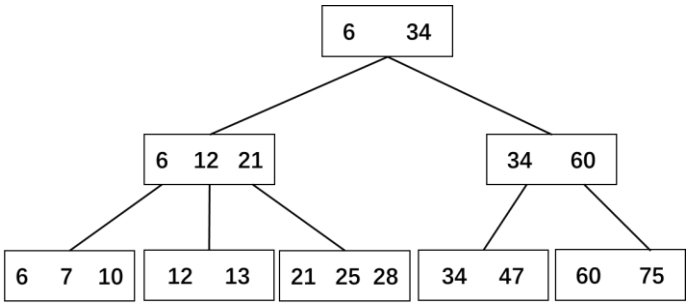
   （注：插入和删除操作过程中读入内存的节点都在内存）



参考答案：

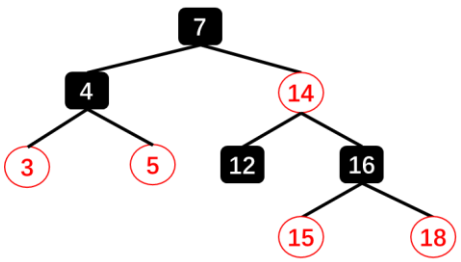


插入 10 的结果（读 3 次，写 5 次）

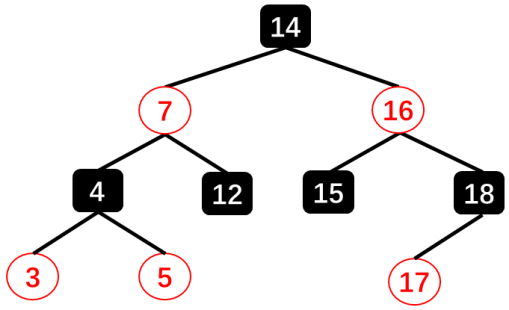


删除 4 的结果（写 3 次）

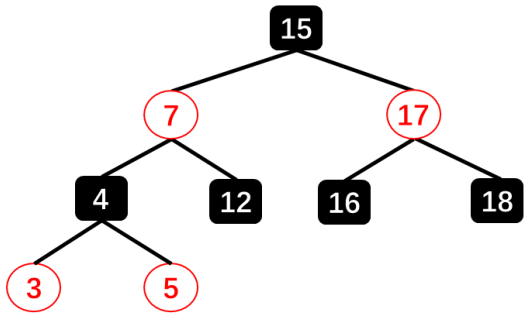
4. 一棵红黑树如下图所示（空树叶未画出），请首先画出插入节点 17 的红黑树，在此基础上，然后画出删除节点 14 的红黑树。（画图说明：黑节点用方框，红节点用圆圈，不画空树叶）



参考答案:

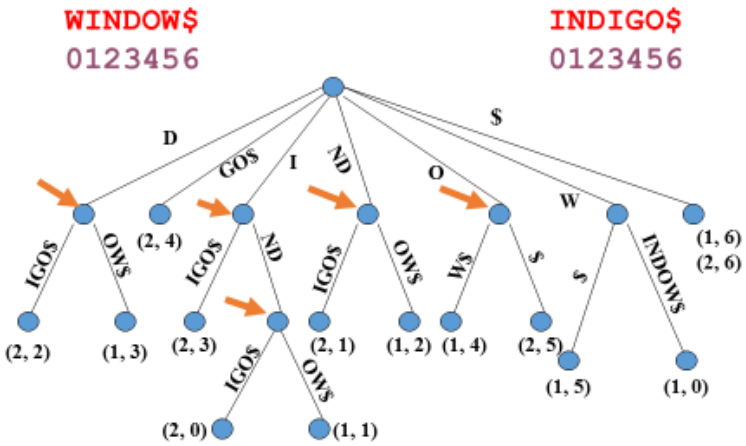


插入 17 号节点



删除 14 号节点

5. 字符串的后缀是指字符串的任意尾部串。比如 “abbc”的后缀有“abbc”, “bbc”,“bc”, “c”和“$”(表示为空串)。现有两个字符串“window”和“indigo”，请画出他们所有后缀串所组成的 Trie 树（注意：对路径进行压缩，并把局部子串标注在相应的边上）。



得分	

三、算法填空题（每空 1 分，共 5 分）

1. 下述算法实现在图 G 中计算从顶点 i 到顶点 j 之间长度为 len 的简单路径条数。图的 ADT 如下：

```
Class Graph {
    public:
      int VerticesNum();
      int EdgesNum();
      Edge FirstEdge(int oneVertex);
      Edge NextEdge(Edge preEdge);
      bool IsEdge(Edge onEdge);
      int FromVertex(Edge oneEdge);
      int ToVertex(Edge oneEdge);
};

int visited[MAXSIZE];        //初始化为 0
int GetPathNum_Len(Graph& G, int i, int j, int len)   {
      if (_____i == j && len == 0_____)   return 1;
      sum = 0;                    //sum 表示通过本结点的路径数
      visited[i] = 1;
      for (Edge e = G.FirstEdge(i); G.IsEdge(e); e = G.NextEdge(e)) {
            int v = G.ToVertex(e);
            if (!visited[v])
                  sum += GetPathNum_Len(G, v, j, len - 1) ;
      } // for
      visited[i] = 0; //本题允许曾经被访问过的结点出现在另一条路径中
      return sum;
} // GetPathNum_Len
```

2. 下面的代码实现了一种计数排序：对每个待排序记录，扫描整个序列统计比它小的记录个数 count，count 即是该记录在序列中的争取位置。请将下面是计数排序的程序代码补充完整。

```
Template <class Record> void Sort(Record Array[], int n){
      int curIndex = 0;            //待排下标
      while (curIndex < n) {
            int count=0;
            for (_int  j=0; j<n; j++_)   //统计比 Array[curIndex]小的值的个数
                  if (Array[j] < Array[curIndex])
                        _____count++_____
            swap(Array, curIndex, count);
            if (curIndex == count)
                  _____curIndex++_____
      }
}
```

得分	

四、设计分析题（共 9 分）

1. （3 分）某小区有 N 座别墅需要供水。在第 i 座别墅里挖井需要 w[i]的费用，在第 i 座和第 j 座别墅之间铺水管需要 c[i][j]的费用。给每座别墅供水，要么挖井、要么跟其他有井的别墅铺设连通的水管路径。请设计算法，求解使每座别墅都得到供水的最小费用方案。

参考答案：

构造由 N 座别墅和井组成的带权无向图，别墅之间边的权重是 c[i][j]，井跟别墅之间边的权重是 w[i]。最小生成树就是最小费用的方案。

2. （6 分）现有一个工资系统，请实现满足如下操作需求的 Splay 树，并给出算法的伪代码：（Splay 树根为 root，其他变量可以自己定义）

   1) void insert(int w);　新加一位工资为 w 的员工（假设 w 没有重复值）

   2) void fire(int t);　　解雇工资大于 t 的员工

相关定义和函数说明如下：

伸展树是一种自平衡的BST，数据结构如下：

```
struct TreeNode {
    int     wage;                  //表示员工工资
    TreeNode * father,* left,* right;
};
```

可以直接使用的函数：

```
void Splay(TreeNode* x, TreeNode* f);//将x旋为f的子结点(f在x的祖先路)
                                     //把x旋到根结点即Splay(x, NULL)
TreeNode* find(int x, TreeNode* S);   //表示在以 S 为根的树中查找元素 x 的位置，若找不到，返回 x 应该插入位置的父亲结点
void Delete(TreeNode* x); //  删除以 x 结点为根的子树
```

```
void insert(int w){
    if (!root) {
        root = new TreeNode(w);
        return;
    }
    int salary = w;
    TreeNode * tmp = find(wage, root);
    if (tmp->wage > salary) {
        tmp->left = new TreeNode(salary);
        Splay(tmp->left, NULL);
    }
    else {
        tmp->right = new TreeNode(salary);
        Splay(tmp->right, NULL);
```

```
        }
    }

    }
    void fire(int t){
        if (!root) return;
        salary = t;
        TreeNode * tmp = find(salary, root);
        if (tmp->wage == salary){
            Splay(tmp, NULL);
            Delete(tmp->right);
        }
        else {
            if (tmp->wage > salary) {
                tmp->left = new TreeNode(salary);
                Splay(tmp->left, NULL);
            }
            else {
                tmp->right = new TreeNode(salary);
                Splay(tmp->right, NULL);
            }
            Delete(root->right);
            if (root->left) {
                Splay(root->left, NULL);
                Delete(root->right);
            } else Delete(root);
        }
    }
```

得分

五、期中考试题（共 25 分，成绩由助教登记）

得分

六、上机考试题（共 35 分，成绩由助教登记）