北京大学信息科学技术学院 2005-2006 学年

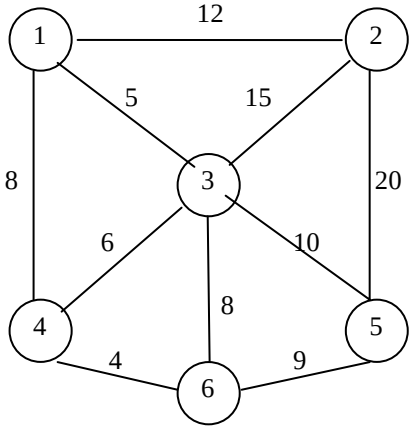
第一学期本科生期末考试参考答案

考试科目：数据结构与算法 A　考试时间：2005 年 1 月 6 日

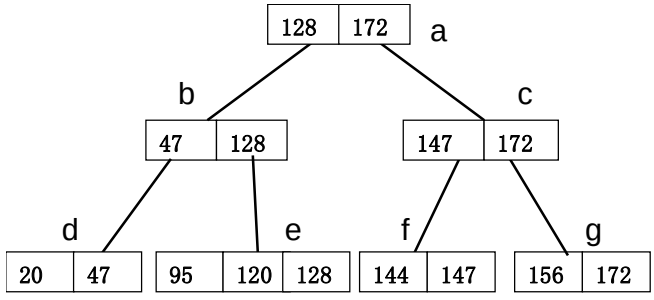
——张铭、赵海燕、王腾蛟

一、（30 分）填空

1. （4 分）设 G=（V，E），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，对图的不同的深度优先序列有_5__个，它们是_ $V_0V_1V_3V_2$； $V_0V_2V_1V_3$； $V_0V_2V_3V_1$； $V_0V_3V_1V_2$； $V_0V_3V_2V_1$。
2. （2 分）n（n>0）个顶点的连通无向图各顶点度之和最少为_2×(n-1)_。
3. （4 分）对于下图，用 Prim 算法从顶点 1 开始求最小生成树，按次序产生的边为_(1,3),(3,4),(4,6),(6,5),(5,2)_；用 Kruskal 算法产生的边次序为_(4,6),(1,3),(4,3),(6,5),(2,5)_。



一、3 图



一、9 B+树

4. （4 分）排序算法的时间开销主要可用算法执行中的__比较__次数和__移动/交换__次数来衡量。
5. （2 分）对 10 个元素的序列（49,38,65,97,76,12,27,50,49*,80）按从小到大的顺序进行排序，选择排序的第 3 趟的结果为__12，27，38，97，76，49，65，50，49*, 80__。
6. （2 分）40　27　32　15　14　20　25　11 是一个堆，其中 32 的子女为 __20（左），25（右）__。

7. （2 分）用快速排序算法对线性表排序，若选择表中第 1 个元素作为分界元素，表中元素按下列____C___情况分布时排序效率最高。

A．已经有序　　B．部分有序　C．完全无序　　D．逆序

8. （5）设输入文件包含记录的关键码是：14, 22, 7, 24, 15, 16, 11, 100, 10, 9, 20, 12, 90, 17, 13, 18, 26, 38, 30, 25, 50, 28, 110, 21, 40，采用外部结点数为 8 的败者树对该文件生成初始顺串，请给出生成的第一个顺串：_____

解答：7, 10, 11, 14, 15, 16, 17, 20, 22, 24, 26, 30, 38, 50, 90, 100, 110

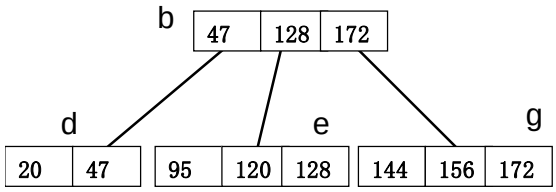
以下解答也有败者树思想，给 3 分：7, 10, 11,12, 14, 15, 16, 17, 18, 21, 22, 24, 25, 26, 28, 30, 38, 40, 50, 90, 100, 110

（5 分）如上图所示为一棵 3 阶 B+树。假定读入一个结点、新建一个结点、修改一个结点、归还一个空结点都是一次访问外存的操作，那么删除关键码 147 的修改过程访问外存操作的顺序为 读 a、c、f；读 g（；合并 f、g；），写 g；读 b（；合并 b、c），写 b；归还 f、c、a。

注意：(1) 回答说归还 g、b、a 也可以；

(2) 没有回答“合并”也不扣分；

(3) 回答读 5 次，写 3 次，归还 3 个也可以。



二、（20 分）简答和辨析题

1. （8 分）假设一个数据文件每个记录对象需要占用 128 字节（其中关键码占用 4 字节），且所有记录均已按关键码有序地存储在主磁盘文件中。设磁盘页块大小为 2048（= 2K）字节，若主存中有 12M 空间可以用来存储索引结构，索引项中每一个地址指针占 8 字节。请简要回答以下问题（请写明你的计算过程）。

（1） 使用 B 树索引，B 树的阶 m 最多可以为多少？

注：在 B 树中找到关键码的同时，应该可以得到其在主文件中的地址。

（2） 4 层 m 阶 B 树，最多可以索引多少字节的数据文件？

注：独根 B 树算 1 层，空 B 树算 0 层；要求根据题目给出的数据，给出计算结果和具体的计算过程。

（3） 假设尽量把 B 树的头几层放入内存（本题规定不能超过 12M），那么给定关键码，通过 B 树查找到(2)小题中主数据文件的一个记录，最少几次访外？最多几次访外？

答案：

（1） （2 分）(m + m-1) * 8 + (m-1) * 4 <= 2048, m = 103 阶

本小题如果算成 m * 8 + (m-1) * 4 <= 2048, m =171 阶，给 0 分。

（2） （3 分）一个根结点 103-1 = 102 个关键码；

第 2 层，是根的 103 个子结点，103 * 102 个关键码；

第 3 层，有 103* 103 个结点，103 * 103 * 102 个关键码；

第 4 层，有 103* 103 * 103 个结点，103 * 103 * 103 * 102 个关键码；

共有 102 * [1 + 103 + 103*103 + 103 * 103 * 103] =

= 102 * 1103440 = 112550880 个关键码

一个关键码对应于一个数据记录，因此有 128 * 112550880 = 14406512640 字节约为 14.4G 字节的数据文件。

如果按照 2^10 = 1024 算，2^30 = 1073741824，14406512640 /（2^30）= 13.42G。也算正确，不扣分。

注意：本题如果计算公式正确，m 是其他值（例如 171），也可以得满分。

（3） （3 分）假设尽量把 B 树的头几层放入内存（本题规定不能超过 12M），那么给定关键码，通过 B 树查找到(2)小题中主数据文件的一个记录，最少几次访外？最多几次访外？

第 1 层，2048 字节，

第 2 层，103 * 2048 = 210944 字节，

第 3 层，103 * 103 * 2048 = 21727232 字节 = 21 M > 12 M

因此，只有第 1、2 层可以存放在内存。

最少访外一次，即在根中找到（关键码，主文件索引）对，然后通过主文件索引找到数据记录。

最少访外三次，即叶层（第 4 层）找到（关键码，主文件索引）对，因为头两层在内存，所以找索引要两次访外，然后通过主文件索引找到数据记录。

注意：本题如果按照 171 计算，也应该得出第 1、2 层可以存放在内存。本题如果回答，“0 次，2 次”共给 2 分。

2. （6 分）什么叫基本聚集？什么叫二级聚集（或称二次聚集）？二次探查法可以解决二级聚集吗？为什么？

解答：大致意思正确就给分。

（2 分）基本聚集：**散列地址不同的关键码**争夺同一后继散列地址的现象。或者这样说：基地址不同的关键码，其探查序列的某些段重叠的现象。

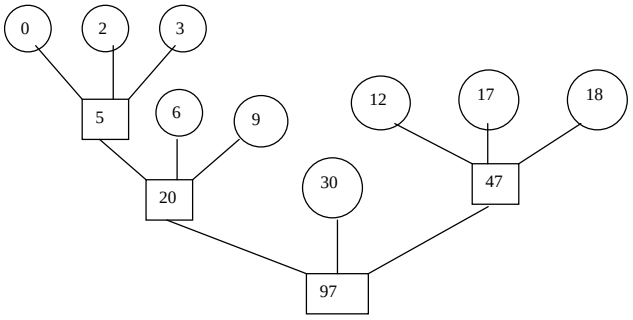
（2 分）二级聚集：**散列到同一基地址的两个不同的关键码**，探查序列重合的现象。

（2 分）二次探查法不能解决二级聚集，因为探查函数是基地址的函数，这样散列到同一基地址的关键码，其探查序列是相同的。

3. （6 分）假设置换选择得到 8 个初始归并段，其长度（即记录数）依次为：9，30，12，18，3，17，2，6，如果假定每个记录占一个物理块，现作 3 路归并，请设计出一个读写外存次数最小的归并方案，并求出归并这些顺串读写外存的次数。

解答：

(1) 归并方案，3 分

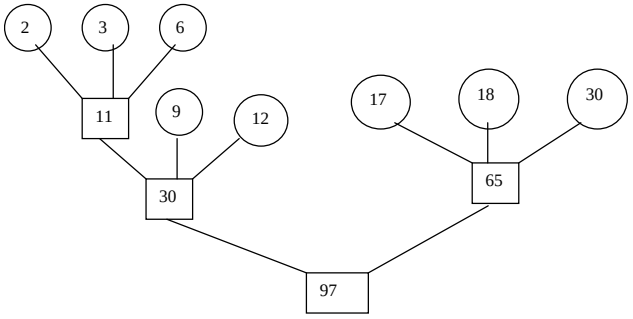


（2）3 分

上述归并方案，读、写外存次数分别为（2＋3）+(5+6+9)+(12+17+18)+(30+20+47)=169 次。

或者 (2+3)*3 + (6+9+12+17+18)*2 + 30 = 169

注意：本题如果给出以下形式，只给 3 分



方案给 1.5 分

(2+3+6)*3 + (9+12+17+18+30)*2 = 205，结果给 1.5 分

三、（20 分, 每空分值不等）算法填空

阅读下面的算法，填充空格(可能是一个表达式、一条语句，也可能是多条语句)，使其成为完整的算法。

1. 完成找出有向图所有的根的算法：（12 分，每空 3 分）

```
static int length=1;
void root (graph &G) {
Using std queue;
Queue <int> aqueue;
for(int i=0;i<G.verticesNum();i++) {
        aqueue.push(i);
        G.mark[i]=VISITED;

        while (①  (!aqueue.isempty())  ) {        //循环条件
            int x=aqueue.pop();
```

```
            for(Edge e=G.FirstEdge(x);G.Isedge(e);e=G.NextEdge(e))
                if (G.Mark[G.ToVertices(e)]==UNVISTED)        {

                        ②_____length++___;      //记录访问的结点数目
                    G.Mark[G.ToVerties(e)]=VISITED;
                    ③ aqueue.push(G.ToVertices(e))  ;          //保存当前结点
                }
            }
            if  (④___length==G.numberVert  )
                        //如果访问的结点数与图的顶点数相等，则该顶点就是图的根
                cout<<i;
        length=1;
        aqueue.MakeEmpty();
        G.MarkAllUNVISITED();
    }
    return;
}
```

2. 下面的算法对一个由非零实数组成的数列进行重排列，使得负数排在前面，正数排在后面，其中有些空缺，请根据题意填写之。（8 分，每空 4 分）

```
template <class Record,class Compare>
void Sort (Record Array[], int n)  {
    int i, j;
    i = 0;
    j = n-1;
    while (i < j)  {
            while (①_____i<n && Array[i] <0_____)
                i++;
            while (②__j>=0 && array[j] >0_____)
                j--;
            if (i>j) break;
            swap(Array, i,  j);
            i++;        j--;
    }
}
```

注释：习题集上 7.14 题同，参考快速排序的 partition 算法，以 0 为轴值进行一次分割即可。每一空如果只填对一个条件的话，可给 2 分。

四、（30 分）算法设计分析题

1. （15 分）假设有向图采用邻接表存储，设计一个算法，判定该图中是否存在回路。

（1） 请概要说明你的算法思想。
（2） 编写你的算法，请在算法关键的地方给出必要的注释。
（3） 分析该算法的时间复杂度。

解答：

（1）思路－用拓扑排序的方法来检查是否存在回路，具体步骤如下：

（1）选择一个入度为 0 的未被访问的顶点

若找到，继续第（2）

若没找到，则图中有环，算法结束

（2）将该顶点标记为已访问，并将其所有出边所指向的顶点的入度减一

（3）回到第一步继续执行

（2）

```
bool HasCirclePath( Graph &G) {
    for (int i=0; i<G.VerticesNum(); i++)       //初始化 Mark 数组
        G.Mark[i]=0;
    using std::stack;
    Stack<int> S;
    for(i=0;i<G.VerticesNum();i++){
        if(!G.Indegree[i])                      //度数为零的顶点入栈
            S.push(i);
    }
    int count = 0;                              //已访问过的顶点数，排序开始前为 0
    while( !S.empty() ) {                       //如果栈中还有图的顶点
        int V = S.top();
        S.pop();
        count++;                                //已访问过的顶点数加一
        G.Mark[V] = 1;
        for( Edge e = G.FirstEdge(V); G.IsEdge(e); e= G.NextEdge(e) ){
            //所有与之相邻的顶点入度减一
            G.Indegree[G.ToVertex(e)]--;//边 e 的终点的入度值减一
            if(!G.Indegree[G.toVertex(e)])
                S.push(G.toVertex(e));//入度为零的点入栈
        }
    }
    //若已访问的顶点个数小于图的顶点个数则为有环图，否则无环
    return ( count<G.VerticesNum() );
}
```

(3) O(2|V|+|E|) 其中 V 为图的顶点数，E 为图的边数

2. （15 分）已知“奇偶转换排序”如下所述：第 1 趟对所有奇数的 i，将 a[i]和 a[i+1]进行比较，第 2 趟对所有偶数的 i, 将 a[i] 和 a[i+1]进行比较，每次比较时若 a[i]>a[i+1]，则将二者交换，以后重复上述两趟过程交换进行，直至整个数组有序。

（1） 试问排序结束的条件是什么？

（2） 编写一个实现上述排序过程的算法，请给出必要的注释。
（3） 分析该算法的时间复杂度。

解答：

请参考习题集的 7.30。

（1）（3 分）当扫描至第 n（n>1）趟时，无论该趟是奇数趟还是偶数趟，如果交换次数为 0 的话，则说明此序列已经有序，结束扫描。 故排序结束的条件为连续两趟（奇数趟和偶数趟）都没有交换，说明每相邻的三个元素是单调递增的，所以整个序列是单调递增的，已经有序。

（2）（8 分）算法的主体部分为一个两重循环，内层循环为奇数/偶数趟的比较交换过程，外层循环为重复这样的比较交换。可用一个变量来记录一次外层循环有无交换，若没有，则已排好序，结束。 算法思想正确者可给 5 分，注释给为 1 分， 代码为 2 分。

常见算法有 2：

方法一：习题集 p205 的代码：（1、内层的两个循环合为 1 个来表示,用一个记录趟数的变量来控制奇数 趟还是偶数趟； 2、用交换计数变量 num 来记录有无交换发生）

```
/奇偶排序，Array[]为待排序数组，n 为数组长度
template <class Record,class Compare>
void OddAndEvenSorter<Record,Compare>::Sort(Record Array[], int n) {
      int t=0;            // 记录趟数
      int num;            // 记录每趟交换的次数
      do {
            num=0;        // 每趟交换次数初值为 0
           for (int i=t%2;i<n-1;i=i+2)  //若 t 为奇数则奇排序，否则偶排序
               if Compare::gt(Array[i],Array[i+1])  {
                   swap(Array,i,i+1);
                    num++;              // 交换次数加 1
               }
               t++;          // 趟数加 1
   } while (t>=2 && num==0); // 循环中止条件：某趟交换次数为 0 且该趟非第一趟
}
```

方法二：更一般的方法，大致如下：

```
…….
Sort(Record Array[], int n) {
      int I, noswap;
      Record temp;
      noswa = TRUE;
      do {
            noswap = TRUE;            // 每趟交换开始 noswap 为真
            for (int i=1; i< n-1; i=i+2)   // 奇排序
               if (Array[i].key > Array[i+1].kry)  {
                   swap(Array,i,i+1);
```

```
                noswap = FALSE；              // 置 noswap 为假
            }
        for (int i=0; i< n-1; i=i+2)   // 偶排序
            if (Array[i].key > Array[i+1].kry)  {
                swap(Array,i,i+1);
                noswap = FALSE；              // 置 noswap 为假
            }
    } while (noswap == FALSE ); // 循环中止条件：不再有交换发生
}
```

(3) （4 分） 算法的时间复杂度分析，可以参考冒泡排序的分析或习题集 p205-206 对最坏情况的分析。时间复杂度平均和最坏情况为 O（n^2），最好情况为 O(n)。 答对平均情况者给 3 分，其他也对为 4 分。

当初始序列为从小到大排列即最好的情况时，只需两趟就可判断此序列为有序序列，两趟中对于任意的 i（0<=i<n-1）都比较了 A[i]和 A[i+1]的大小，所以排序码比较次数为 n-1 次。

当初始序列为从大到小排列即最坏的情况。分情况分析。

平均情况与冒泡排序类似。