班级：21级自动化 6 班 姓名： 吴俊达 学号： 210320621

**第六章 树与二叉树**

1.（P19）设深度为h的二叉树上只有叶子结点和同时具有左右子树的结点，则此类二叉树中所包含的结点数目至少为( 2h-1 )。[选项无效]

A.2^h B.2^(h-2) C.2^(h＋1) D.2^(h-l)

2. （P20）二叉树的深度为k，则二叉树最多有 ★ 个结点。( D )

  A.2^k             B.2^(k-1)              C.2^(k-1) D.2^k-1

3.（P25）若一个二叉树具有8个度为2的结点，7个度为1的结点，则度为0的结点（叶子）个数是 9 。【终端节点数 = 度为2的结点数+1】

4. （P25）具有n个结点的二叉树中，一共有 2n 个指针域，其中 n-1 个用来指向结点的左右孩子，有 n+1 个为NULL。

5. （P50）假设一棵二叉树的先序遍历序列为ABDEHCFGI，中序遍历序列为DBEHAFCIG

（1）请画出这棵二叉树；

（2）请写出后序遍历结果。

解：（1）二叉树为：【解题要点：ABDEHCFGI，DBEHAFCIG】

|  |  |
| --- | --- |
| ABDEHCFGI，DBEHAFCIG | → 先序遍历的第一个节点点A为根节点，进而其中序遍历左侧结点BDEH属于A的左子树，右侧结点FCIG属于A的右子树 |
| ABDEHCFGI，DBEHAFCIG | → 中序遍历序列第一个节点为子树根节点。则B为A的左子树根节点，C为A的右子树根节点 |
| ABDEHCFGI，DBEHAFCIG | → 同样，中序遍历结果中，每个子树的根节点的左侧为该子树的左子树，右侧为右子树。蓝色标出的为子树的左子树，肉色标出的为子树的右子树。 |



（2）后序遍历结果是：DHEBFIGCA

6.（P97）设叶子结点a、b、c、d、e的权值分别为5、4、10、12、9，构造Huffman树，并计算带权路径长度。

解：Huffman树为：



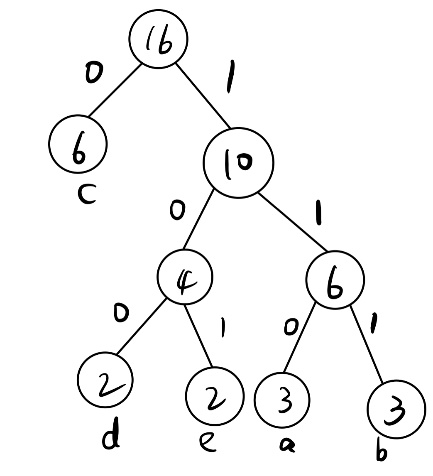
带权路径长度为3\*5+3\*4+9\*2+10\*2+12\*2=89

7.（P116）对电文'abbaccdeeccdabcc'进行Huffman编码。

画出生成的静态链表---Huffman树和各字符编码，最后形成的code数组。

解：在电文中a出现3次，b出现3次，c出现6次，d出现2次，e出现2次。

构建Huffman树如图：



静态链表如下表：**(空缺表示无双亲或无孩子)**

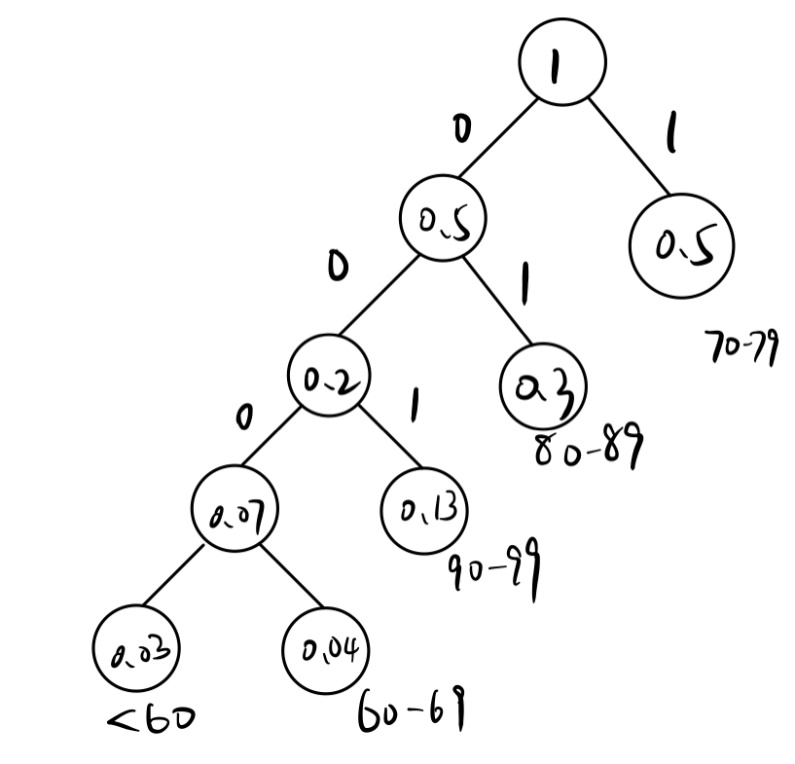
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Data | weight | Parent | Left Child | Right Child | code |
| 0 | a | 3 | 6 |  |  | 110 |
| 1 | b | 3 | 6 |  |  | 111 |
| 2 | c | 6 | 8 |  |  | 0 |
| 3 | d | 2 | 5 |  |  | 100 |
| 4 | e | 2 | 5 |  |  | 101 |
| 5 |  | 4 | 7 | 3 | 4 |  |
| 6 |  | 6 | 7 | 0 | 1 |  |
| 7 |  | 10 | 8 | 5 | 6 |  |
| 8 |  | 16 |  | 2 | 7 |  |

8.（P120）作业：假设某班成绩分布如下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 分数 | ＜60 | 60—69 | 70—79 | 80—89 | 90—99 |
| 比例数 | 0.03 | 0.04 | 0.5 | 0.3 | 0.13 |

为查找某一分数段同学成绩，需进行比较。试构造二叉树，使得总比较次数最少。

解：构造的二叉树为：

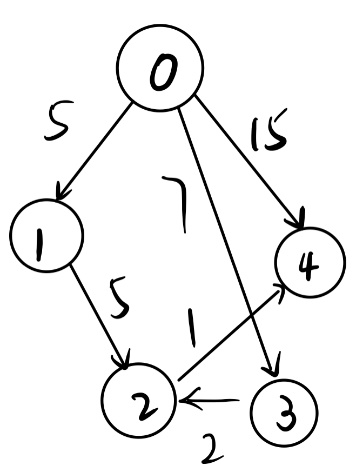


**第七章 图**

1.（P23）有向网N={V,E}，V={0,1,2,3,4}，E={<0,1,5>，<0,3,7>，<0,4,15>，<1,2,5>，<2,4,1>，<3,2,2> }，E中每个元组的第三个元素表示权。

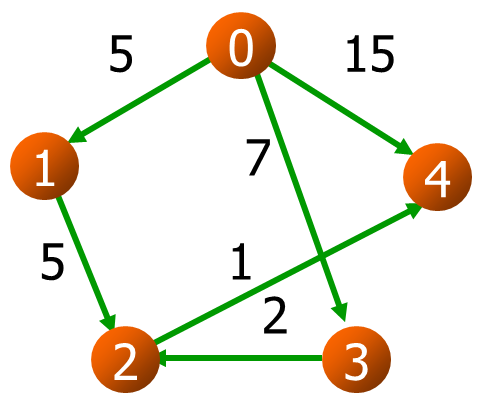
（1）画出该网； （2）写出该网的邻接矩阵。

解：（1）该网如下所示：

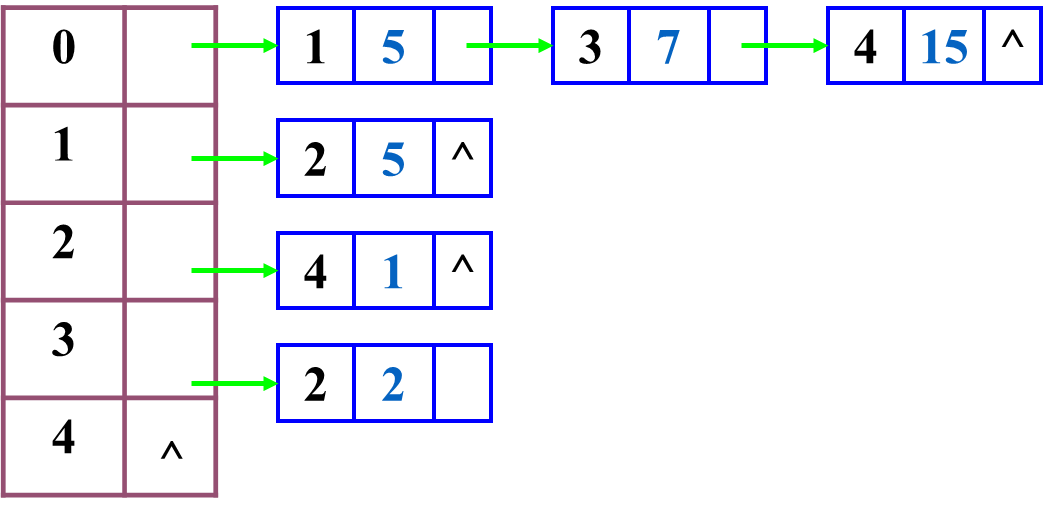


（2）邻接矩阵为：

2. （P31）写出该网的邻接表。



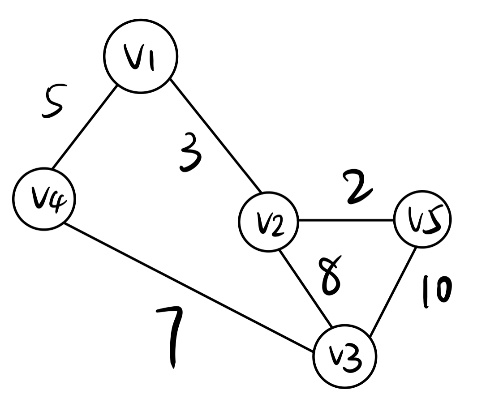
解：邻接表如下：（黑色为节点编号，蓝色为边的权值）



3. (P60) 假设无向网G的邻接表表示如下图，写出深度、广度优先遍历结果。



解：该图如下：



深度优先遍历结果：V1,V4,V3,V5,V2

（V1的孩子节点中第一个找到的是V4[因为邻接表中紧接着就存这个]，然后找V4的孩子节点，第一个找到的是V3，然后找V3的孩子节点，第一个找到的是V5，然后找V5的孩子节点，第一个找到V3但是已经访问过，所以再找一个得到V2，遍历结束。）

广度优先遍历结果：V1,V4,V2,V3,V5

（先找V1的孩子节点，找到V4、V2入队，V1出队，然后V4出队，找V4的孩子节点，找到V3并入队【V2已经访问过】，然后访问V2的孩子节点，找到V5[V1已经访问过]，此时所有节点都访问过，遍历结束。）

4. （P80）用Prim和Kruskal算法分别求下图的最小生成树，给出生成过程。



解：Prim算法的最小生成树生成过程：（从V1开始）

Kruskal算法的最小生成树生成过程：（依次选取权值最小边）

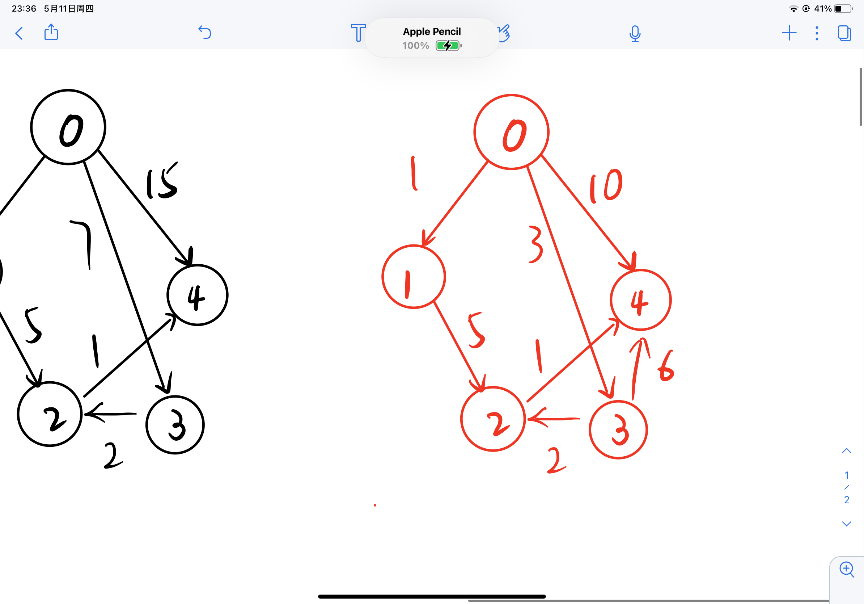
  

5. （P99）有向网N={V,E}，V={0,1,2,3,4}，E={<0,1,1>，<0,3,3>，<0,4,10>，<1,2,5>，<2,4,1>，<3,2,2>，<3,4,6>}，E中每个元组的第三个元素表示权。

1) 画出该网。

2) 写出该网的邻接矩阵。

3) 用Dijkstra算法求最短路径，写出顶点0到其它各顶点的最短路径长度、路径及产生过程。

解：（1）该网如右所示。

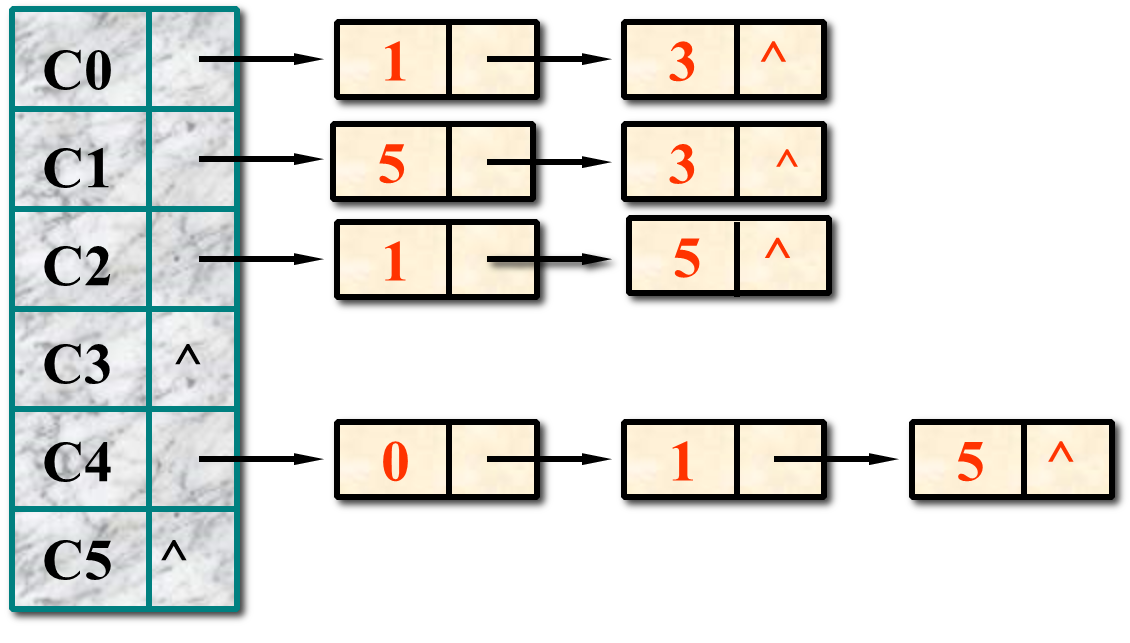
（2）邻接矩阵为：

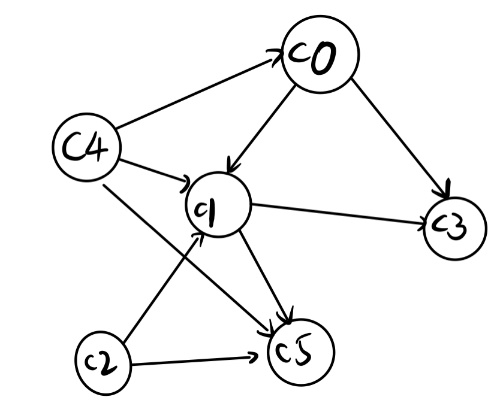
（3）Dijkstra算法求最短路径过程如下：

（每个终点的表格行的第一个数字是visited[访问标志]，0代表未访问过、1代表访问过；第二个数字是目前的最短距离；第三个值是路径所包含的节点集合）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 终点 | 从节点0到各终点的D值和最短路径的求解过程（Dijkstra） | | | | |
| i=1（初态） | i=2 | i=3 | i=4 | i=5（末态） |
| 1 | 0  1  {0, 1} | 1  1  {0, 1} | 1  1  {0, 1} | 1  1  {0, 1} | 1  1  {0, 1} |
| 2 | 0  ∞ | 0  6  {0, 1, 2} | 0  5  {0, 3, 2} | 1  5  {0, 3, 2} | 1  5  {0, 3, 2} |
| 3 | 0  3  {0, 3} | 0  3  {0, 3} | 1  3  {0, 3} | 1  3  {0, 3} | 1  3  {0, 3} |
| 4 | 0  10  {0, 4} | 0  10  {0, 4} | 0  9  {0, 3, 4} | 0  6  {0, 3, 2, 4} | 1  6  {0, 3, 2, 4} |
| 下一步加进的节点 | 1 | 3 | 2 | 4 |  |
| U | {0,1} | {0,1,3} | {0,1,3,2} | {0,1,3,2,4} | {0,1,3,2,4} |

6. （P127）某AOV网的邻接表存储结构如下，写出分别用队列和栈存储入度为零的顶点时的拓扑排序序列。



解：AOV图如右。

用队列实现时的拓扑排序序列：C2、C4、C0、C1、C3、C5

[过程：寻找入度为0的节点时，C2、C4依次入队，然后C2出队[先进先出]，修改入度，没有元素入队。然后C4出队，修改入度，C0入队。然后C0出队，修改入度，C1入队。然后C1出队，修改入度，C3、C5依次入队。然后C3、C5依次出队，结束。]

用栈实现时的拓扑排序序列：C4、C0、C2、C1、C5、C3

[过程：寻找入度为0的节点时，C2、C4依次入栈，然后弹出C4[后进先出]，修改入度，C0入栈。然后弹出C0，修改入度，没有元素入队。再弹出C2，修改入度，C1入栈。然后弹出C1，修改入度，C3、C5入栈。然后依次弹出C5、C3，结束。]