

1. 一棵有 124 个叶结点的完全二叉树最多有_____个结点。

答案: 248

解答: 根据二叉树的性质,叶子结点数目为双分支节点数加 1, 当 $n_0=124$ 时, $n_2=123$, 因此该完全二叉树最多可有 $124+123+1=248$ 个结点。

2. 设完全二叉树的第 6 层有 24 个叶子结点, 则此树最多有_____个结点。

答案: 79

解答: 根据完全二叉树每层结点总数最大值的计算公式可以算出第 6 层最多有 $2^{6-1}=2^5=32$ 个结点。其中有 24 个叶子结点, 假设他们全部在该层靠右排列, 那么在这一层还有 $32-24=8$ 个非叶子结点, 由此可以推断第 7 层还有 $2 \times 8=16$ 个叶子结点, 因为上一层靠左的 8 个结点都有两个孩子。综上所述, 该完全二叉树最多有 $1+2+4+8+16+32+16=79$ 个结点。

3. 设一棵二叉树的中序遍历序列为 badce, 后序遍历序列为 bdeca, 则该二叉树前序遍历的序列为_____。

答案: abcde

解答: 由二叉树的后序遍历序列和中序遍历序列即可唯一确定这棵二叉树。做法如下: 首先用后序遍历序列确定根结点, 利用得到的根结点对中序遍历序列分割根以及其左、右子树得到左右子树的中序遍历子序列。按照同样的方法递归处理每一个子序列即可还原出这棵二叉树。前序遍历序列的结果为 abcde。

4. 在哈夫曼编码中, 若编码长度只允许小于或等于 4, 则除了已知两个字符编码为 0 和 10 外, 还可以最多对_____个字符编码。

答案: 4

解答：根据哈夫曼树生产哈夫曼编码的规则可知，若哈夫曼编码的长度只允许小于或等于 4，则哈夫曼树的高度为 5。已知一个字符编码为 0，另一个为 10。这说明第 2 层和第 3 层各有一个叶子结点，为使得该树从第 3 层起能够对尽可能多的字符编码，余下的二叉树应该都为满二叉树。因此最底层可以有 4 个叶子结点，则最多可以再对 4 个字符编码。

5. 在具有 n ($n \geq 1$) 个结点的 k 叉树中，用 k 叉链表来存储，则有_____个空指针。

答案： $(k-1) \times n + 1$

解答：有 n 个结点的 k 叉树中总共有 $n-1$ 个分支， k 叉链表中总共有 $k \times n$ 个子指针，其中有 $n-1$ 个非空指针对应到 $n-1$ 个分支，因此有 $k \times n - (n-1) = (k-1) \times n + 1$ 个空指针。

6. 设森林 F 对应的二叉树为 B ，它有 m 个结点。 B 的根为 p ， p 的右子树中结点个数为 n ，则森林 F 中第一棵树的结点个数是_____。

答案： $m-n$

解答：将森林 F 转化为二叉树表示 B ，则 B 的根是第一棵树的根，根的左子树是第一棵树的根的子树森林，根的右子树是森林中除去第一棵树外其他树构成的森林。根据题意， B 的根是 p ， p 的右子树中的结点个数为 n ，则森林 F 的第一棵树中结点个数为 $m-n$ 。可以参照课本 130 页的图 5-33 来理解。

7. 设 G 是一个非连通无向图，有 15 条边，则该图至少有_____个顶点。

答案：7

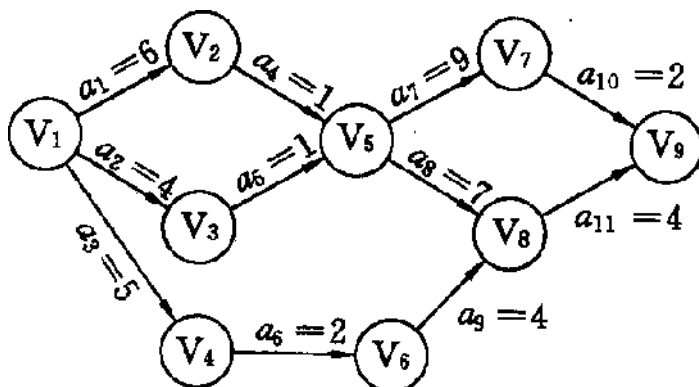
解答：根据连通图的性质以及顶点与边数的关系即可求解：设无向图有 n 个顶点，它的边数 $e \leq n(n-1)/2$ 。若 $e=15$ ，则可求解得 $n \geq 6$ 。在连通图情形下至少需有 6 个顶点，在非连通图情形下则至少需有 7 个顶点。

8. 设有向图具有 n 个顶点和 e 条边, 如果用邻接表作为它的存储结构, 则拓扑排序的时间复杂度为_____。

答案: $O(n+e)$

解答: 对于有 n 个顶点和 e 条边的有向图, 建立各顶点的入度的时间复杂度为 $O(e)$, 建立入度为 0 的栈的时间复杂度为 $O(n)$ 。在拓扑排序过程中, 最多每个顶点进一次栈, 入度减 1 的操作最多总共执行 e 次, 可知总的时间复杂度为 $O(n+e)$ 。

9. 画出下图 AOE 网的关键路径

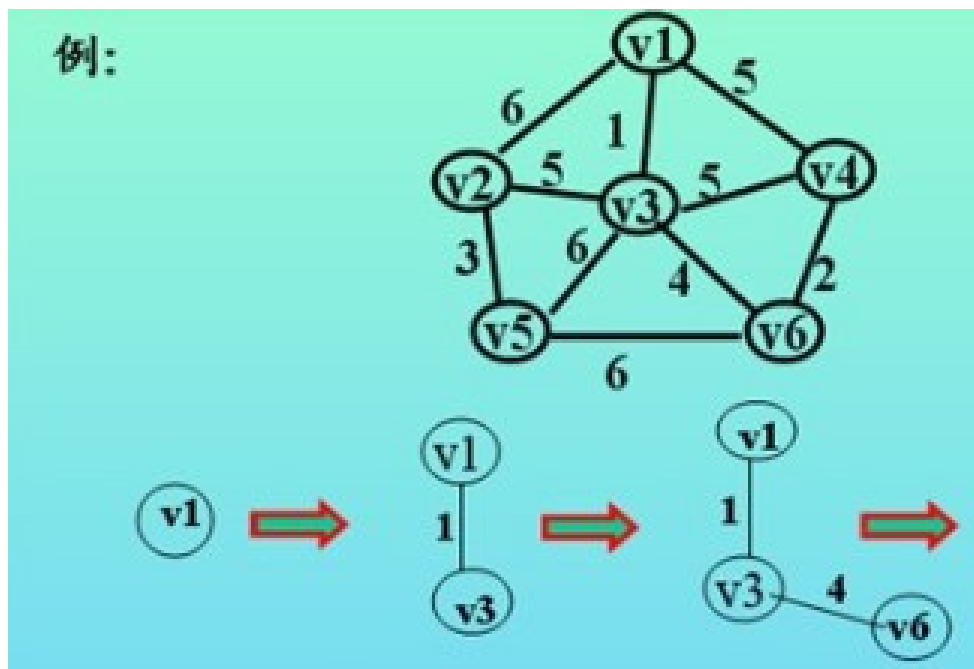


解答: 第一条关键路径 $a_1a_4a_7a_{10}$ 、第二条关键路径 $a_1a_4a_8a_{11}$

事件 j	$e_v[j]$	$L_v[j]$	活动 i	$e[i]$
1	0	0	1	0
2	6	6	2	0
3	4	6	3	0
4	5	8	4	6
5	7	7	5	4
6	7	10	6	5
7	16	16	7	7

10. 使用 Prim 算法构造如下图所示的一棵最小生成树。

解答:



11. 编程题: 计算以 t 为根的二叉树各结点中的最大元素的值, 假设数据类型为 `float` 型。

解答: 设 `max` 是事先设定的最大值, 判断二叉树中各结点是否有比 `max` 还大的值, 有则修改 `max` 使之保持最大值。注意, 算法在递归语句中不是通过函数返回最大值, 而是通过函数参数表返回最大值, 所以 `max` 是引用型参数。

答案:

```
void MaxValue(BiNode *t, float &max)
{
    if(t!=NULL)
    {
        if(t->data>max)
            max=t->data;
        MaxValue(t->lchild,max);
        MaxValue(t->rchild,max);
    }
}
```

12. 编程题：将一个无向图的邻接矩阵 A 转换为邻接表 B。

解答：设置一个空的邻接表，填入对应的顶点信息，然后在邻接矩阵上查找非零元素，找到后在邻接表对应边表中插入相应的边表结点。

答案：

```
void MatToList(MGraph A, ALGraph &B)
{
    B.vertexNum=A.vertexNum;
    B.arcNum=A.arcNum;
    for(i=0; i<A.vertexNum; i++)
    {
        B.adjlist[i].vertex=A.vertex[i];
        B.adjlist[i].firstedge=NULL;
    }
    for(i=0; i<A.vertexNum; i++)
        for(j=0; j<i; j++)
            if(A.arc[i][j]!=0) {
                ArcNode *p=new ArcNode;
                p->adjvex=j;
                p->next=B.adjlist[i].firstedge;
                B.adjlist[i].firstedge=p;
            }
}
```