

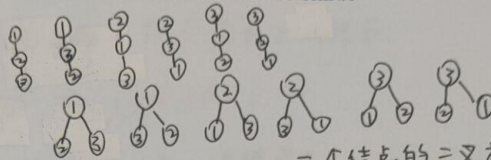
1902001038 岳宇轩

数据结构 第六章



中国海洋大学  
OCEAN UNIVERSITY OF CHINA

3. 三个结点的树:



三个结点的二叉树与上述相同

5.

$$\begin{cases} n = n_0 + n_1 + \dots + n_k \\ n = B + 1 \\ B = n_1 + 2n_2 + \dots + kn_k \end{cases}$$

$$\begin{aligned} n_0 &= n - (n_1 + n_2 + \dots + n_k) \\ &= (n_1 + 2n_2 + \dots + kn_k) + 1 - (n_1 + n_2 + \dots + n_k) \\ &= 1 + \sum_{i=1}^k (i-1)n_i \end{aligned}$$

$$6. \begin{cases} n = n_0 + n_k \dots ① \\ n = B + 1 \dots ② \\ B = kn_k \dots ③ \end{cases}$$

$$\begin{aligned} \text{由 } ②③ \quad n &= kn_k + 1 \\ n_k &= \frac{n-1}{k} \dots ④ \end{aligned}$$

$$\begin{aligned} \text{由 } ①④ \quad n_0 &= n - n_k \\ &= n - \frac{n-1}{k} \end{aligned}$$



中国海洋大学

OCEAN UNIVERSITY OF CHINA

10. (1) 设总共有  $N$  个结点,  $n_2$  个度为 2 的结点.

$$\begin{cases} N = n + n_2 \\ n = n_2 + 1 \end{cases} \Rightarrow N = 2n - 1$$

(2) ①  $n=1$  时, 只有一个根结点, 显然成立

② 假设  $n=k$  时成立, 即  $\sum_{i=1}^k 2^{-(i-1)} = 1$

那么,  $n=k+1$  时, 在第  $k$  层 少一个叶子结点, 在第  $k+1$  层多两个叶子结点,  $\sum_{i=1}^{k+1} 2^{-(i-1)} = \left( \sum_{i=1}^k 2^{-(i-1)} \right) - 2^{-(k-1)} + 2 \times 2^{-(k+1-1)}$

$$= \sum_{i=1}^k 2^{-(i-1)} = 1 \text{ 也成立}$$

$$\text{综上所述 } \sum_{i=1}^n 2^{-(i-1)} = 1$$

13.

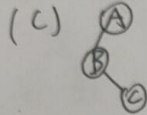
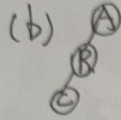
答	前序遍历时 $n$ 在 $m$ 前?	中序遍历时 $n$ 在 $m$ 前?	后序遍历时 $n$ 在 $m$ 前?
已知			
$n$ 在 $m$ 左方	1	1	1
$n$ 在 $m$ 右方	0	0	0
$n$ 是 $m$ 祖先	1	$\emptyset$	0
$n$ 是 $m$ 子孙	0	$\emptyset$	1

14. (a) 所有结点无左子树的树  
(b) 所有结点无右子树的树  
(c) 只有一个根结点的树

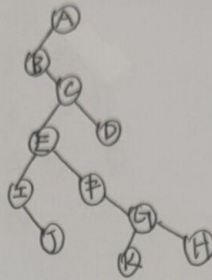


中國海洋大學  
OCEAN UNIVERSITY OF CHINA

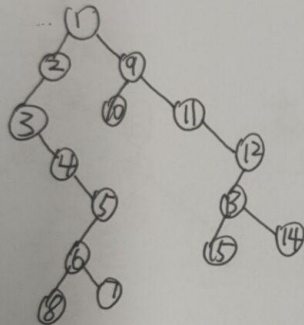
19. (a) A



(d)



20.



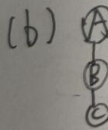
(1) 123456879101112131514

(2) 348675211091115131412

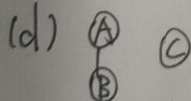
(3) 876543210151413121191

21.

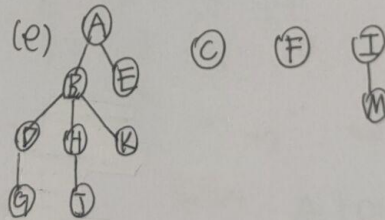
(a) A



(c) A B C



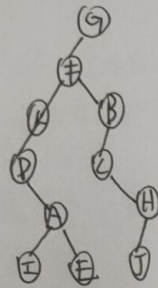
(e)



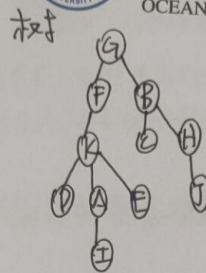


中国海洋大学  
OCEAN UNIVERSITY OF CHINA

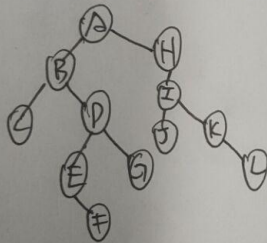
23. 二叉树



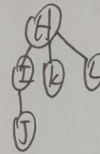
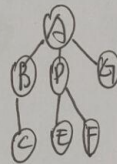
$\Rightarrow$



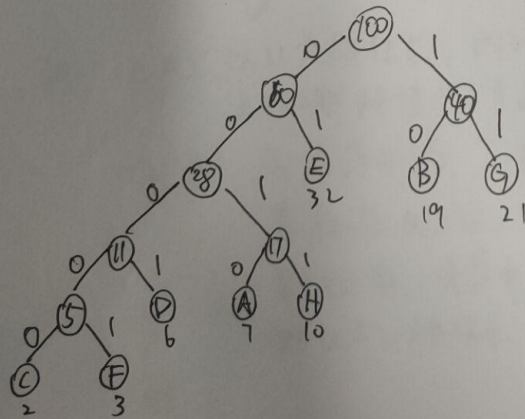
24. 森林  
二叉树



森林  
 $\Rightarrow$



26. A B C D E F G H  
7 19 2 6 32 3 21 10  
✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓  
5 11 17 28 40 19 21 28



Huffman

A	0010	4
B	10	2
C	0000	5
D	0001	4
E	01	2
F	0000	5
G	11	2
H	0011	4

ori

A	000	3
B	001	3
C	010	3
D	011	3
E	100	3
F	101	3
G	110	3
H	111	3





中国海洋大学

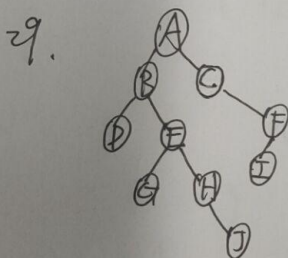
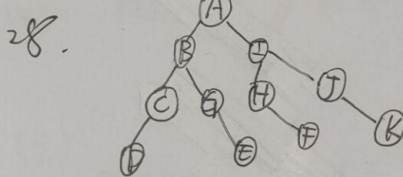
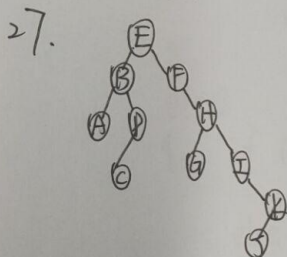
OCEAN UNIVERSITY OF CHINA

$$4 \times 0.07 + 2 \times 0.19 + 5 \times 0.02 + 4 \times 0.06 + 2 \times 0.32 + 5 \times 0.03 + 2 \times 0.24 + 4 \times 0.1$$

$$= 0.28 + 0.38 + 0.1 + 0.24 + 0.64 + 0.15 + 0.42 + 0.4$$

$$= 2.61 < 3$$

可以看到, Huffman 编码更高效, 可以节省空间, 但编码长度不一定规整



30. 证“ $\Rightarrow$ ”以  $u$  为根结点, 则  $v$  可能在  $u$  的左子树或右子树中。  
无论哪种情况, 先序序列  $u$  在  $v$  前, 后序序列  $u$  在  $v$  后。

证“ $\Leftarrow$ ”先序序列为“根  $\rightarrow$  左  $\rightarrow$  右”, 后序序列为“左  $\rightarrow$  右  $\rightarrow$  根”

若存在  $u, v$ , 使得先序中先  $u$  后  $v$ , 后序中先  $v$  后  $u$ ,

则必有: 先序中“根  $\rightarrow$  左  $\rightarrow$  右” 后序中“左  $\rightarrow$  右  $\rightarrow$  根”

即  $u$  为  $v$  根结点  $\Rightarrow u$  为  $v$  祖先

得证



中国海洋大学

OCEAN UNIVERSITY OF CHINA

36. Status Similar (BiTree & T1, BiTree & T2)

```
{ if (!T1)
  { if (!T2)
    return TRUE; // B1, B2 皆空
    else
    return FALSE; // B1 空, B2 不空
  }
  else
  { if (!T2)
    return FALSE; // B1 不空, B2 空
    else
    { if (Similar (T1->lchild, T2->lchild)
      && Similar (T1->rchild, T2->rchild))
      return TRUE; // 递归, 左右子树分别相似
      else
      return FALSE;
    }
  }
}
```



中国海洋大学

OCEAN UNIVERSITY OF CHINA

41. Status GetKNode (TElemType &e, int &i, int k, BiTree &T)

```
{ if (T)
    { i++;
      if (i == k) // 找到第k个位置.
        e = T->data;
      else
        { GetKNode (e, i, k, T->lchild); // 左子树中找
          GetKNode (e, i, k, T->rchild); // 右子树中找
        }
    }
  return OK;
}
```

42. Status CalculateLeafNodeNumber (int &i, BiTree &T)

```
{ if (T)
    { if (!T->lchild && !T->rchild) // 无左右子树是叶子结点
        i++;
      CalculateLeafNodeNumber (i, T->lchild); // 左子树叶子结点数
      CalculateLeafNodeNumber (i, T->rchild); // 右子树叶子结点数
    }
  return OK;
}
```





中国海洋大学  
OCEAN UNIVERSITY OF CHINA

43. Status Exchange (Bi Tree & T)

```
{ if (T)
```

```
{ BiTree temp; // 临时变量
```

```
temp = T->lchild; // 交换当前结点的左右子树
```

```
T->lchild = T->rchild;
```

```
T->rchild = temp;
```

```
Exchange (T->lchild); // 交换左子树的左右子树
```

```
Exchange (T->rchild); // 交换右子树的左右子树
```

```
}
```

```
return Ok;
```

```
}
```

47. Status LevelorderVisit (BiTree &T, Status (\*Visit)(ElemType e))

```
{ QElemType p;
```

```
Queue q;
```

```
InitQueue (q);
```

```
if (T)
```

```
EnQueue (q, T); // 树不为空, 进队列
```

```
while (!QueueEmpty (q)) // 当队列不为空时
```

```
{ DeQueue (q, p); // 出队列一个元素, 赋给 p
```

```
Visit (p->data);
```

```
if (p->lchild)
```

```
EnQueue (q, p->lchild); // 进队列 p 的左孩子 (若存在)
```

```
if (p->rchild)
```

```
EnQueue (q, p->rchild); // 进队列 p 的右孩子 (若存在)
```

```
}
```

```
return Ok;
```

```
}
```

地址: 青岛市松岭路238号

邮编: 266100

电话: 0532-66782730

传真: 0532-66782799

网址: <http://www.ouc.edu.cn>

edu.cn





中国海洋大学  
OCEAN UNIVERSITY OF CHINA

49. Status Complete BiTree (BiTree &T)

```
{ if (T)
{ int d = BiDepth(T->lchild) - BiDepth(T->rchild); //深度
  if (d < 0 || d > 1) //d < 0, 表示有结点无左孩子有右孩子
    return FALSE;
  else
  { if (Complete BiTree (T->lchild)
    && Complete BiTree (T->rchild)) //左右子树均为完全二叉树
    return TRUE;
    else
    return FALSE;
  }
}
```

56.

Status PreOrderThreading (BiThrTree &Thrt, BiThrTree &T)

{ // 构造先序线索二叉树

BiThrTree pre;

Thrt = new BiThrNode; // 线索二叉树的头结点

Thrt->LTag = Thread;

Thrt->RTag = Link;

Thrt->lchild = Thrt; // 左子树回指

if (!T)

Thrt->rchild = Thrt // 若二叉树空, 右子树回指



中国海洋大学  
OCEAN UNIVERSITY OF CHINA

```
else
{ Thrt → rchild = T;
  pre = Thrt;
  PreThreading(T, pre); // 次序遍历进行次序线索化
  pre → RTag = Thread;
  Thrt → lchild = pre;
}
return OK;
```

}

status PreThreading (BiThrTree &T, BiThrTree &pre)

```
{ if (T)
{ if (!T → lchild)
{ T → LTag = Thread;
  T → lchild = pre;
}
if (pre && !pre → rchild)
{ pre → RTag = Thread;
  pre → rchild = T;
}
pre = T;
if (T → LTag == Link) PreThreading(T → lchild, pre);
if (T → RTag == Link) PreThreading(T → rchild, pre);
}
return OK;
```



中国海洋大学  
OCEAN UNIVERSITY OF CHINA

```
60. int LeafNum (CSTree &T)
{ if (T)
    { if (!T->firstchild) //无孩子
        return 1+LeafNum(T->nextsibling); //本结点+兄弟
      else //孩子+兄弟
        return LeafNum(T->firstchild) + LeafNum(T->nextsibling);
    }
  else
    return 0; //空树
}
```

```
62. int Depth (CSTree &T)
{ int d1, d2;
  if (T)
    { d1 = 1 + Depth(T->firstchild);
      d2 = Depth(T->nextsibling);
      return d1 > d2 ? d1 : d2; //取最大
    }
  else
    return 0;
}
```





中国海洋大学  
OCEAN UNIVERSITY OF CHINA

```
Status FindNext (BiThrTree &q, TElemType *p)
{ //从二叉线索树上q结点开始查找结点p, 若找到则把后继存入p指针
  BiThrTree pt = q;
  if (!pt) return FALSE;
  if (pt->data == *p)
  { if (pt->LTag == Link)
    q = pt->lchild;
    else
    q = pt->rchild;
    return OK;
  }
  pt = q->rchild;
  while (pt != q && pt->data != *p)
  { if (pt->LTag == Link)
    pt = pt->lchild;
    else
    pt = pt->rchild;
  }
  if (pt == q) return FALSE;
  if (pt->data == *p)
  { if (pt->LTag == Link)
    q = pt->lchild;
    else
    q = pt->rchild;
  }
  return OK;
}
```

地址: 青岛市松岭路238号

邮编: 266100

电话: 0532-66782730

传真: 0532-66782799

网址: <http://www.ouc.edu.cn>





中国海洋大学

OCEAN UNIVERSITY OF CHINA

65. BiTree BuildTree (TElemType \*pre, TElemType \*mid, int prelen, int midlen)

```
{ BiTree tree;
  tree = (BiTree) malloc (sizeof (BiTNode));
  tree->data = pre[0] // 次序首元素作为根结点.
  int root = 0;
  if (pre[0] == mid[0]) // 根结点是中序的首个
    tree->lchild = NULL; // 无左孩子
  else
    { root = Search (tree->data, mid, midlen); // 查找根结点在中序中的位置
      tree->lchild = BuildTree (pre+1, mid, root, root); // root之前为左子树
    }
  if (pre[0] == mid[midlen-1]) // 根结点是中序的末位.
    tree->rchild = NULL; // 无右孩子.
  else
    tree->rchild = BuildTree (pre+root+1, mid+root+1,
                             prelen-root-1, midlen-root-1);
```

return tree;

}

```
int Search (int num, int *array, int length)
{ for (int i=0; i<length; i++)
  if (array[i] == num)
    return i;
```

return -1;

}

地址: 青岛市松岭路238号

邮编: 266100

电话: 0532-66782730

传真: 0532-66782799

网址: <http://www.ouc.edu>



中国海洋大学  
OCEAN UNIVERSITY OF CHINA

```
69. Status Print Tree (BiTree * T, int layer)
{ if (T)
  { Print Tree (T->rchild, layer + 1); // 访问右子树
    for (int j = 0; j < layer - 1; j++)
      printf(" ");
    printf("%c\n", T->data);
    Print Tree (T->lchild, layer + 1); // 访问左子树
  }
}
```