一. 1. A 2. B 3. D 4. A 5. B

6. D 7. B 8. C 9. B 10. A

二. 1. × 2. × 3. × 4. × 5. × 6. ×

7. × 8. √ 9. × 10. √ 11. ⊗ 12. × 13. √

三. 1. 2^{i-1} 2^k-1

2. 2^h-1 $2^{h-1}+1$

3. 5. 2. 4.

4. CEDBGFA

5. 线性结构. 非线性结构

6. $\lfloor \log_2 n \rfloor + 1$

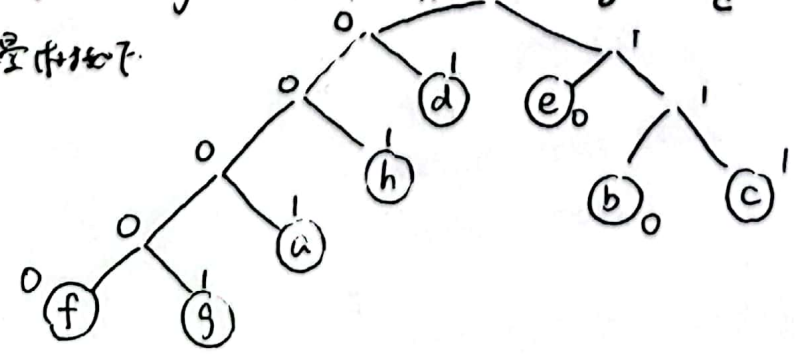
7. 集 10.

8. O(n).

#9. 500.

四. 1. 以小到大排得

0.03	0.04	0.05	0.10	0.14	0.16	0.20	0.28
f	g	a	h	b	c	d	e.



得编码

a	0001
b	110
c	111
d	01
e	10
f	00000
g	00001
h	001

则每个字符二进制位期望为

$0.05\times4+0.1\times3+0.16\times3+0.20\times2+0.28\times2+0.3\times5+0.09\times5+0.1\times3=2.71$

100个字符为 $2.71\times100=271$ 位

2.

```
int judge(Link ptr){                    // 头指针.
    Link p=ptr, q=ptr;
    int L=1, l=0, ret=1;
    initstack(G);
    while(q->next!=p){
        q=q->next;
        L++;
    }
    q=p;                                // 头指针.
    while(l<floor(L/2)){
        pushstack(q, G);
        q=q->next;
    }
    if(L%2)
        q=q->next;                      // 奇数长
    while(!stackempty(G) && ret){
        if(q->data!=top(G)->data)
            ret=0;                      // 不对称
        q=q->next;
        pop(G);
    }
    destroystack(G);
    return ret;
}
```

3. (1)
```
void traverse ( BiTree ptr ) {
static *BiTree pointer = ptr -> next; //静态变量
if (ptr -> lchild)
    traverse (ptr -> lchild);
if (ptr -> rchild)
    traverse (ptr -> rchild);
*pointer = ptr;        // 在前一结点的 next 指针域
pointer = &(ptr -> next);
ptr -> next = NULL;    //当前结点的后继置空
}
```

(2) 只需依次调用 ~~Find (ptrx, ptry);~~ ~~Find (ptry, ptrx);~~ 即可.

```
int Find ( BiTree ans, BiTree des ) {
    if (ans == des)
        return 1;
    if (ans -> lchild && Find (ans -> lchild, des) ||
        ans -> rchild && Find (ans -> rchild, des))
        return 1;
    return 0;
}
```

主函调用

```
if ( Find ( ptrx, ptry ) || Find (ptry, ptrx))
    return 1;   // 有关系
else
    return 0;   // 无关系
```

3) int Flor (BiTree root, BiTree ptr, int floor) {

```
    if (root == ptr)
        return floor;
    int lf = 0, rf = 0;
    if (root -> lchild)
        lf = Flor (~~BiTree~~ root -> lchild, ptr, floor+1);
    if (root -> rchild)
        rf = Flor ( root -> rchild, ptr, floor+1);
    if (lf)
        return lf;
    if (rf)
        return rf;
    return 0;
}
```

调用时为 Flor (root, ptr, 1);