一．选择题

（3）选 C

首先，已知 next[1]=0,next[2]=1,对于 j>=3,令 k=next[j-1],若t_k=t_{j-1},则 next[j]=k+1;否则令 k=next[k],若 k 不等于 0，继续比较t_k与t_j，直至两者相等或 k=0,k=0 时 next[j]=1。

对于本题开始分析：模式串为 ababaaababaa

a	b	a	b	a	a	a
0	1					

分析 next[3]: k=next[2]=1; t1 不等于 t2; k=next[1]=0; next[3]=1

a	b	a	b	a	a	a
0	1	1				

分析 next[4]: k=next[3]=1;t1 等于 t3; next[4]=k+1=2

a	b	a	b	a	a	a
0	1	1	2			

分析 next[5]: k=next[4]=2;t2 等于 t4; next[5]=k+1=3

a	b	a	b	a	a	a
0	1	1	2	3	……	……

后面的分析以此类推

最后得到原模式串对应的 next 数组为：011234223456

(7)选 B

通过列序存储，实际在元素 A[5,8]前存储的元素个数为（8-1）*8+（5-1）=60 个元素

每个元素长度为 3 字节，故所求元素的存储首地址为 BA+74*3=BA+180

(9)选 B

由对称矩阵性质，要找到a_{ij}(i>j)只需要找到a_{ji}，k 等于前 j-1 行的所有元素数量加上第 j 行的 i 个元素，k=j(j-1)/2+i

(15)选 C

由定义知，L 中所含元素只有一个，为(a,b,c)，括号层数为 2，故长度和深度分别为 1 和 2

二．应用题

4.H(H(T(H(T(H(T(L)))))))

分析如下：

T(L)=((orange,(strawberry,(banana)),peach),pear)

H(T(L))= (orange,(strawberry,(banana)),peach)

T(H(T(L)))= ((strawberry,(banana)),peach)

H(T(H(T(L))))= (strawberry,(banana))

T(H(T(H(T(L)))))=((banana))

H(T(H(T(H(T(L))))))=(banana)

H(H(T(H(T(H(T(L)))))))=banana

三．算法题

先在每一行的内部比较，再和后面行数中的元素比较，保证每两个元素都只比较一次

int Equal(int a[m] [n], int m,int n)

```
{
    for(i=0;i<m;i++)
        for (j=0;j<n-l;j++)
        {
            for(p=j+1;p<n;p++;)
                if(a[i][j]==a[i][p])
                {
                    cout<<"no";
                    return0;
                }
            for (k=i+1;k<m;k++)
                for (p=0;p<n;p++)
                    if(a[i][j]==a[k][p])
                    {
                            cout<<"no";
                            return 0;
                    }
        }
    cout<<"yes";
    return 1;
}
```

分析:
情况 1: 若所有元素都不相等, 程序结束时, 所有元素都两两比较过一次, 总次数为 1~（m*n-1）差为一的等差数列求和，即为(m*n)(m*n-1)/2;
情况 2：若存在相等元素，则对于以其中任意一个元素为标准的比较中，比较次数都可能为 1~该元素的在情况 1 中总比较次数之间的任意值，与情况 1 量级相同
故算法的时间复杂度为 O(n^4)