1. 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=,则next[j]=k+1;否则令k=next[k],若k不等于0，继续比较与，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(i>j)只需要找到，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

1. 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

1. 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(