Q1.

T(n)=n5.

Let f(n)=2n, C=1, N=23.

Then for all n≥N, T(n)≤C\*2n → T(n)=O(2n)

Q2.

T(n)=100n+logn.

Let f(n)=n+(logn)2, C=100,N取对数函数的底数.

Then for all n≥N，T(n) ≤ 100\*(n+logn)<100\*(n+(logn)2) → T(n)=O(n+(logn)2).

Q3.

T(n) , k≥d,

Let f(n)=O(nk), x=max{a1…ai…ad}, C=x, N=1.

Then for all n≥N, T(n) ≤ =C\*d\*nk → T(n)=O(nk)

Q4.

T(n) =(n+a)b

Let f(n)=nb,

Q5.

1. 遍历所有的数

Numbers of 0’s：0

Numbers of 1’s：4

Numbers of 2’s：3

Numbers of 3’s：1

Numbers of 4’s：2

Numbers of 5’s：1

Numbers of 6’s：0

Numbers of 7’s：1

Numbers of 8’s：0

Numbers of 9’s：1

1. 输出

1，1，1，1，2，2，2，3，4，4，5，7，9

Q6.

这道题的本质是求两个字符串最长公共子序列S，之后比较B是否等于S即可

**Algorithm**:

dp[m+1][n+1]=0;

**for** i=1 to n do

**for** j=1 to m do

**if** A[i]==B[j] **then** dp[i][j]=dp[i-1][j-1]+1;

**else** dp[i][j]=MAX(dp[i][j-1], dp[i-1][j])

**end**

**end**

**return** length(B)==dp[m][n];

该算法需要遍历一遍m\*n大小的矩阵，所以其时间复杂度是O(m\*n)

Q7.

a=10, b=3, d=2.5, f(n)= n2.5

因为a\*f(n/b)=10\*(n/3)2.5=10/32.5\*n2.5,记作k\*f(n),易得k<1，所以T(n)= Θ(nlog310)