习题2

1.填空

（1）解决很多问题的复杂性往往是因为问题的规模，当问题规模足够小时，问题实际上很容易解决。

（2）递归程序调用应该遵循后调用先返回的原则，具体通过系统所维护的一个栈实现。

（3）递归程序主要的缺点是执行效率低，时间耗费比非递归算法多，主要优点是算法结构清晰简介，可读性强。

（4）平衡子问题指的是子问题的规模大致相同。

（5）大整数乘法和Strassen矩阵乘法的改进思路都是通过适当增加加减法次数来减少乘法次数。

（6）利用递归实现分治法，其计算时间一般包括分解问题，递归解决自问题和合并问题的时间。

（7）问题的复杂度下界是指最优情况下解决问题所耗费的代价。

（8）归并排序算法是分治法比较完美的应用，其完美体现在对初始序列不敏感。

。

（9）归并排序的改进办法主要有二种，分别是消除递归，自上而下，控制合并和自然合并排序。

（10）快速排序最好和最坏情况下的时间复杂度分别是O(nlogn)和O(n^2)。

2.给出梵塔问题的递归方程，并使用递归跟踪方法求时间复杂度

时间复杂度2^n-1

3.分治法的适用条件是什么

问题规模缩小到一定程度就可以容易地解决；

问题可以分解成若干个规模较小的相同问题；

利用自问题的解，可以合并成问题的解，即该问题具有最优子结构性质；

4. 给出分治法的一般算法框架

divide-and-conquer(P)

{

if(|P| <= n0) adhoc(P);

else

{

divide P into smaller substances P1,P2,...,Pk;

for(i=1;i<=k;i++)

{

yi=divide-and-conquer(Pi);

}

Return merge(y1...yk);

}

}

5. Stranssen算法的缺点有哪些？

算法中蕴含常数因子较大；

不太适用与稀疏矩阵，为稀疏矩阵设计的算法更有效；

在某些运用中，数值稳定性不好；

递归中子矩阵需要空间支持；

6.改写二分搜索算法

设a[0:n-1]是已排好序的数组.请改写二分搜索算法,使得当X不在数组中时,返回小于x的最大元素位置i和大于x的最小元素位置j.当搜索元素在数组中时，i和j相同，均为x在数组中的位置。

Int i,j;

void search()

{

if(l > r) return;

int f=(l+r)/k;

int k=a[f];

if(x==k) {i=j=f; return;}

if(x<k) {j=f; search(l,f-1);}

else if(x>k) {i=f;search(f+1,r);}

}

7. 整数因子分解问题

问题描述： 大于1 的正整数n 可以分解为：n=x1 \*x 2\*…\*xm 。                              
例如，当n= 12 时，共有8 种不同的分解式：   
12= 12；   
12=6\*2；   
12=4\*3；   
12=3\*4；   
12=3\*2\*2；   
12=2\*6；   
12=2\*3\*2；   
12=2\*2\*3。

对于给定的正整数n，编程计算n 共有多少种不同的分解式

int Div(int n)

{

int s=0;

if(n==1) {s++; return s;}

for(int i=2;i <= n;i++)

{

if(n%i==0) s+=Div(n/i);

}

return s;

}