http://sourcekade.com/source/?p=1022

```
Int fact(n) p80
{
If (n==0)
    Return 1
Else
    Return n*fact(n-1)
}
```

****

T(n)=c*n

t(n)=O(n)

****

T(n)=T(n-1)+c

=T(n-2) +c+c

=T(n-3) +c+c+c

=T(0) +c+c+…+c

```
Void tower(int n,pg a,pg b,pg c)
p86
{
     If (n==1)
       Move top A to C
     Else{
     Tower(n-1,A,C,B)
       Move top A to C
     Tower(n-1, B,A,C)}
}
```

T(n)$\begin{cases} 1 & n = 1 \\ 2T(n-1)+1 & n > 1 \end{cases}$

T(n)=2^1T(n-1)+1

=2^2T(n-2)+2+1

=2^3T(n-3)+3+2+1

.

.

=2^n-1

T(n)=O(2^n)

```
Int rgcd(int a , int b) p78
{
    If (b==0)
        Return a;
    Else
        Return rgcd(a,a%b)
}
```

$O(\log_2 a)$

```
void quick_sort (int p, int q) p116
{
int j;
if (p>=q) return
else {
  j= partition(p ,q+1);
  quick_sort(p,j-1);
  quick_sort(j+1,q);}
}
```

T(n)$\begin{cases} 0 & n < 1 \\ T(n-1)+C(n+1) & n > 1 \end{cases}$

T(n)=T(n-1)+C(n+1)

=T(n-2)+Cn+C(n+1)

=T(n-3)+C(n-1)+cn+C(n+1)

.

.

=c(3+4+…+(n+1)

<=(c(n+1)(n+2))/2

T(n)=O(n^2)

```
int binsearch(int a[],int low,int hight,int x) p98
{
int m;
    if (low>high)
          return -1;
    else
    {
          m=[(low+high)/2]
          if(x==a[m]
                    return m
          if(x>a[m]
                    return Binsearch(a,m+1, hight,x)
          else
                    return Binsearch(a,low,m-1,x)
    }
}
```

T(n)=T(n/2)+c

= T(n/4)+c+c

=T(n/8)+c+c+c

.

.

=T(0)+c+c+…+c

T(n)=$c\log_2 n$=$O(\log_2 n)$

```
Float xton(float x ,int n)  p124
{
          if(n==1)
                    return x
          y=xton(x,n/2)
          if(n==n/2*2)
                    return y*y
          else
                    return y*y*x
}
```

T(n)$\begin{cases} c & n = 1 \\ T(n/2)+3 & n > 1 \end{cases}$

binsearch همانند

T(n)=$O(\log\ \ n)$

```
void mergsort(int low,int high)
p109
{
    int mid
    if(low<high){
        mid=(low+high)/2
        mergsort(low,mid)
        mergsort(mid+1, high)
        merg(low,mid,high)}

}
```

T(n)$\begin{cases} a & n = 1 \\ 2T(n/2)+cn & n > 1 \end{cases}$

T(n)=2T(n/2)+cn

T(n)=2(2T(n/4)+cn/2)+cn

=4T(n/4)+2cn

.

=2^kT(1)+cnk

=2^k*a+cnk

T(n)=O (n log n)

```
long int lnmult(long int u,long int v,int n){
          long int w,x,y,z
          int s
          if(n<=L)
                    return u*v
          else{
          s=n/2;   w=u/2^s; x=u%2^s; y=v%2^s;
          z=v%2^2;
return lnmult(w,y, [n/2]* 2^2s+(lnmult( w,z ,[n/2])
+lnmult ( x,y, [n/2]))*2^s+lnmul(x ,y,n/2)}
```

T(n)=O(n^2)