

信奥算法

三元表达式

三元表达式

```
if(x%2==1)
    cout<<"Odd";
else
    cout<<"Even";
```



```
cout<<(x%2==1?"Odd":"Even");
```

三元表达式

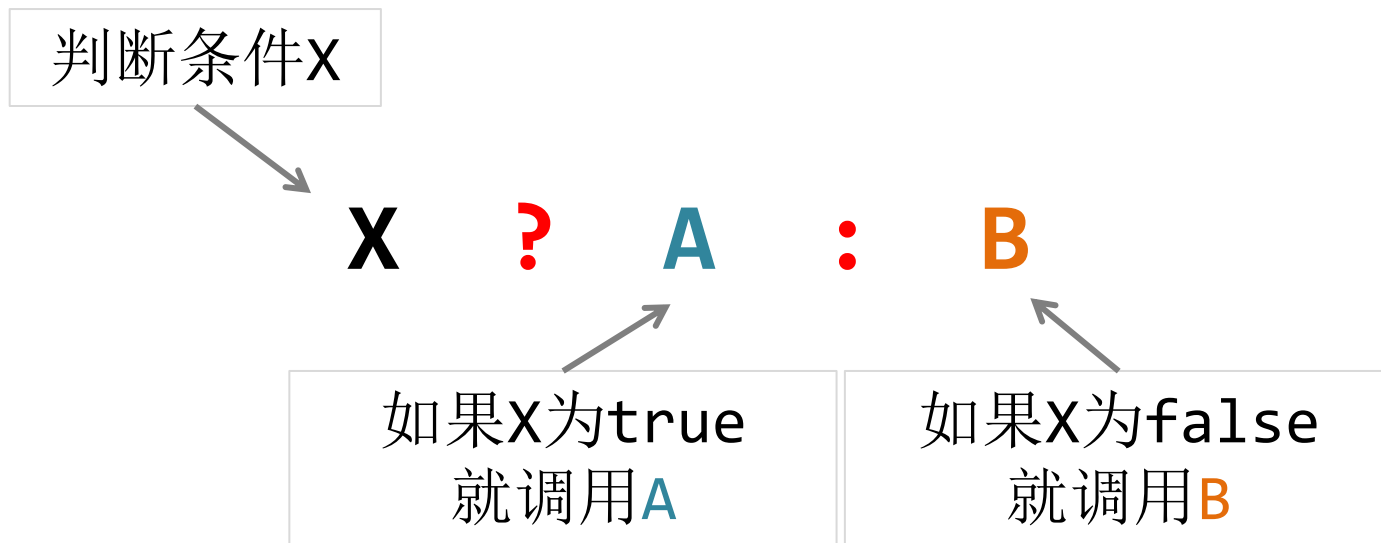
```
if(n>=100)
    price=7*n;
else
    price=10*n;
```

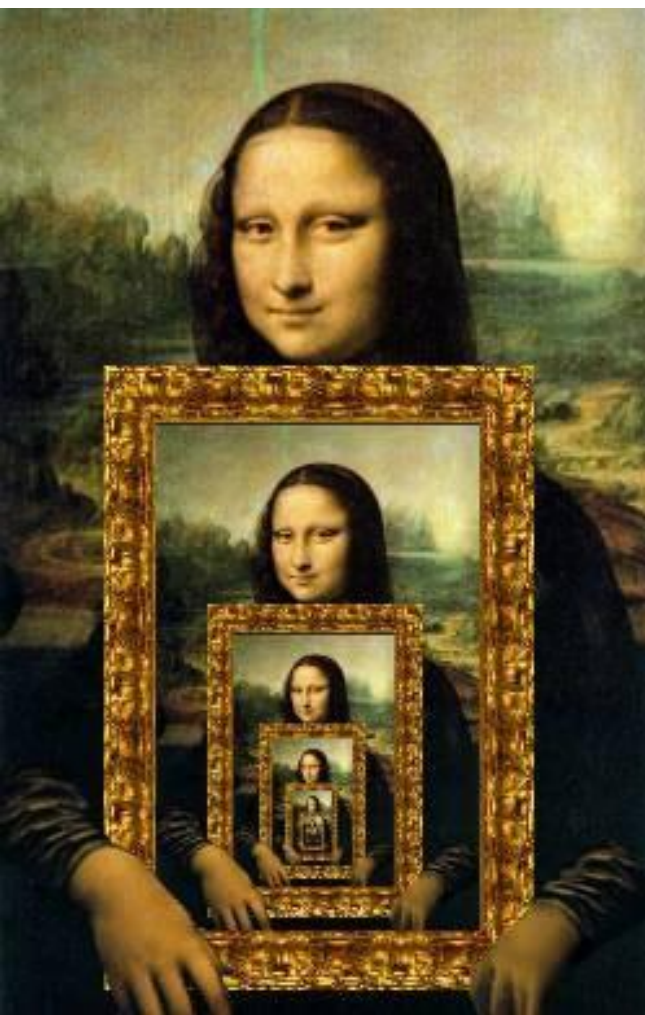


```
price=(n>=100?7*n:10*n);
```

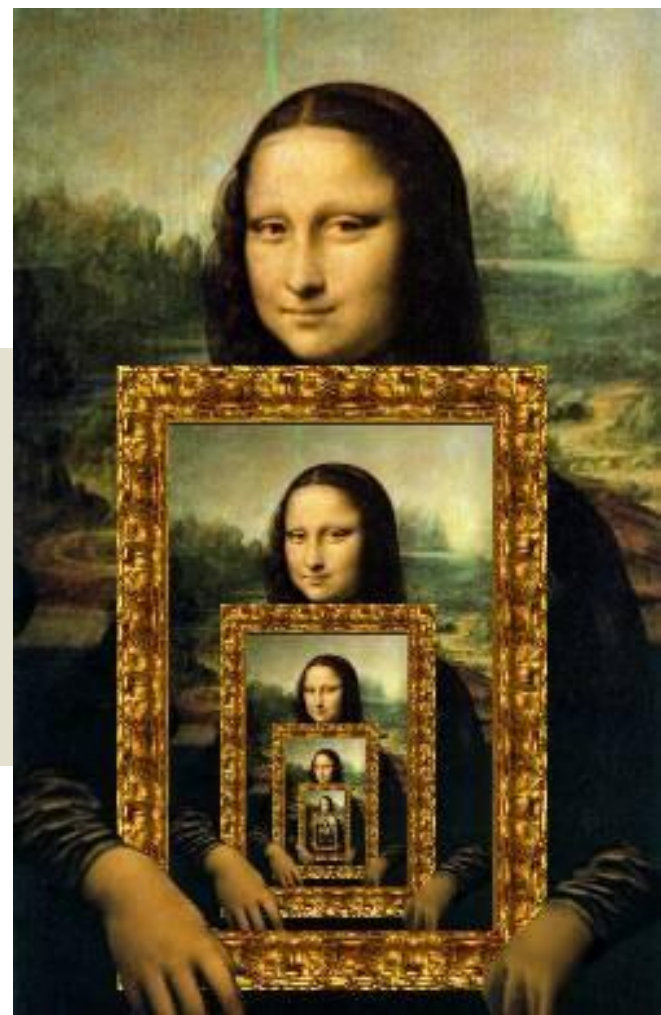
```
price=(n>=100?7:10)*n;
```

三元表达式有时 可以代替if语句





递归
(recursion)



吓得我抱起了

抱着抱着抱着我的小鲤鱼的我的我的我









Hyphaene Compressa - Doum Palm

© Shlomit Pinter

两面镜子

你乘坐电梯时，前后各有一块镜子，你会发现你在镜子的成像数不过来。那是因为你在镜子A中的成像又在镜子B中成像，镜子B的成像又在镜子A中有成像，如此反复……。

踢皮球

小明去找A经理解决问题，A经理说：这件事情不归我管，去找B经理。于是小明去找B经理。B经理说。这件事情不归我管，去找A经理。。如果两个经理的说辞不变，小明又始终听话，小明将永远往返于两者之间。

递归

函数调用自身

吓得我抱起了

抱着抱着抱着我的小鲤鱼的我的我的我



大事化小
小事化了

阶乘函数

n的阶乘 $n! = n \times (n - 1) \times \cdots \times 2 \times 1$

```
3 typedef long long ll;
4 ll f(ll x){
5     ll ans=1;
6     for(ll i=2;i<=x;i++)
7         ans*=i;
8     return ans;
9 }
10 int main(){
11     ll n;
12     cin>>n;
13     cout<<f(n);
14     return 0;
15 }
```

f(x)代表x的阶乘

阶乘函数

n的阶乘 $n! = n \times (n - 1) \times \cdots \times 2 \times 1$

f(x)代表x的阶乘

```
4 11 f(11 x){  
5     if(x==0) return 1;  
6     return f(x-1)*x;  
7 }  
8 int main(){  
9     11 n;  
10    cin>>n;  
11    cout<<f(n);  
12    return 0;  
13 }
```

终止条件

递归:函数调用自身

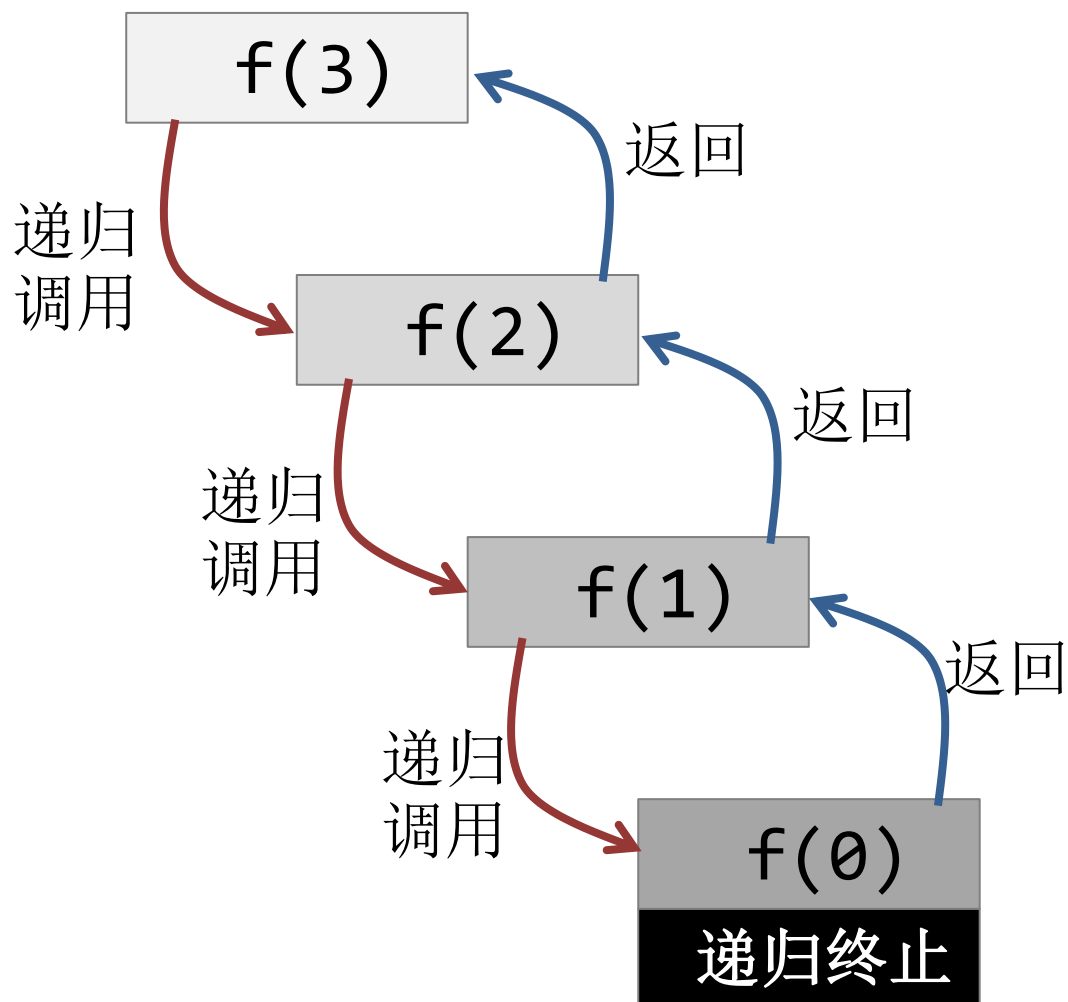
阶乘函数

n 的阶乘 $n! = n \times (n - 1) \times \cdots \times 2 \times 1$

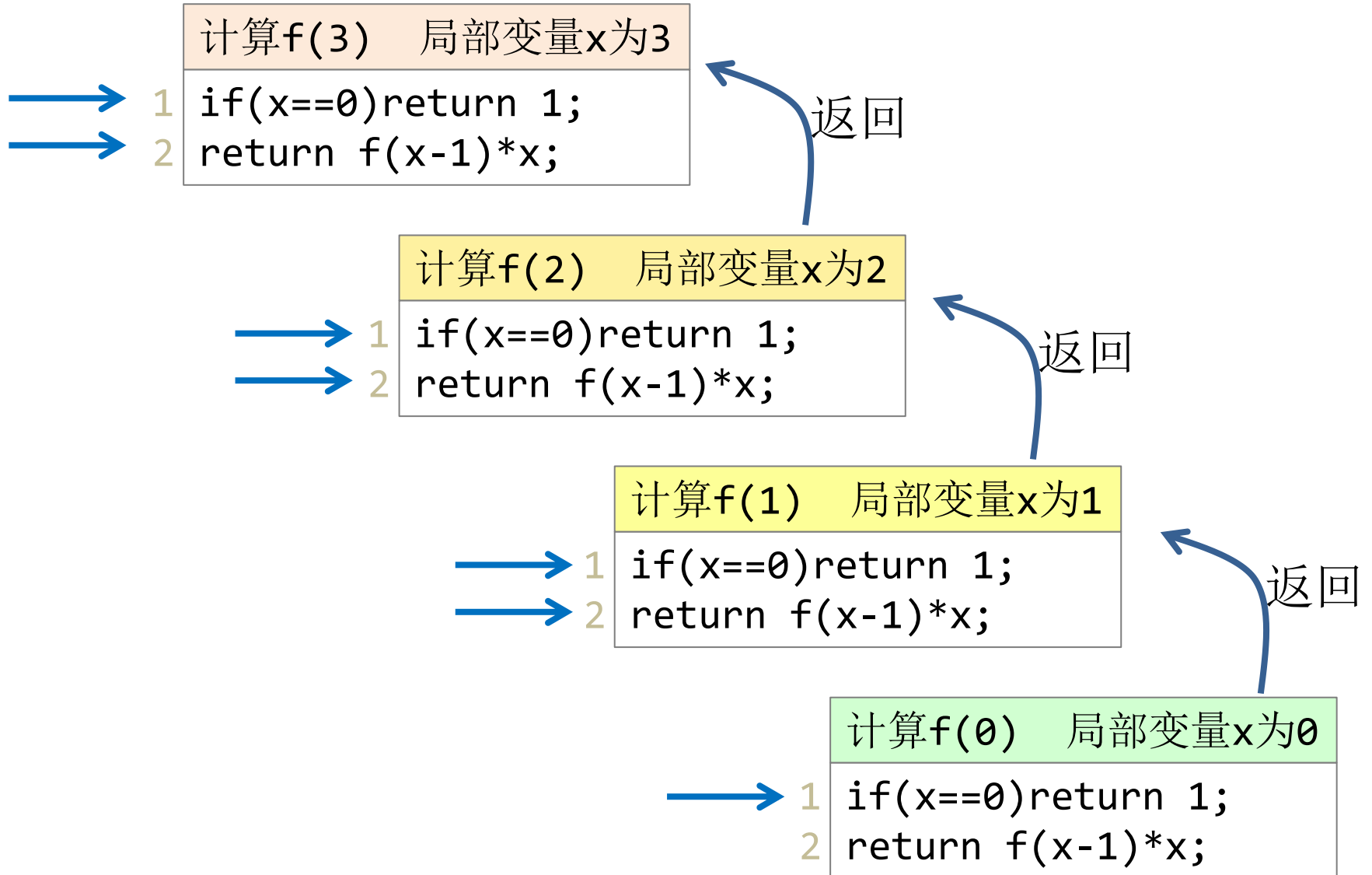
$f(x)$ 代表 x 的阶乘

```
4 11 f(11 x){  
5     return x?f(x-1)*x:1;  
6 }  
7 int main(){  
8     11 n;  
9     cin>>n;  
10    cout<<f(n);  
11    return 0;  
12 }
```

阶乘 - 递归过程



阶乘 - 递归过程



斐波那契数列

```
4 ll F[100];  
5 int main(){  
6     ll n;  
7     cin>>n;  
8     F[1]=F[2]=1;  
9     for(ll i=3;i<=n;i++)  
10         F[i]=F[i-1]+F[i-2];  
11     cout<<F[n]<<endl;  
12     return 0;  
13 }
```

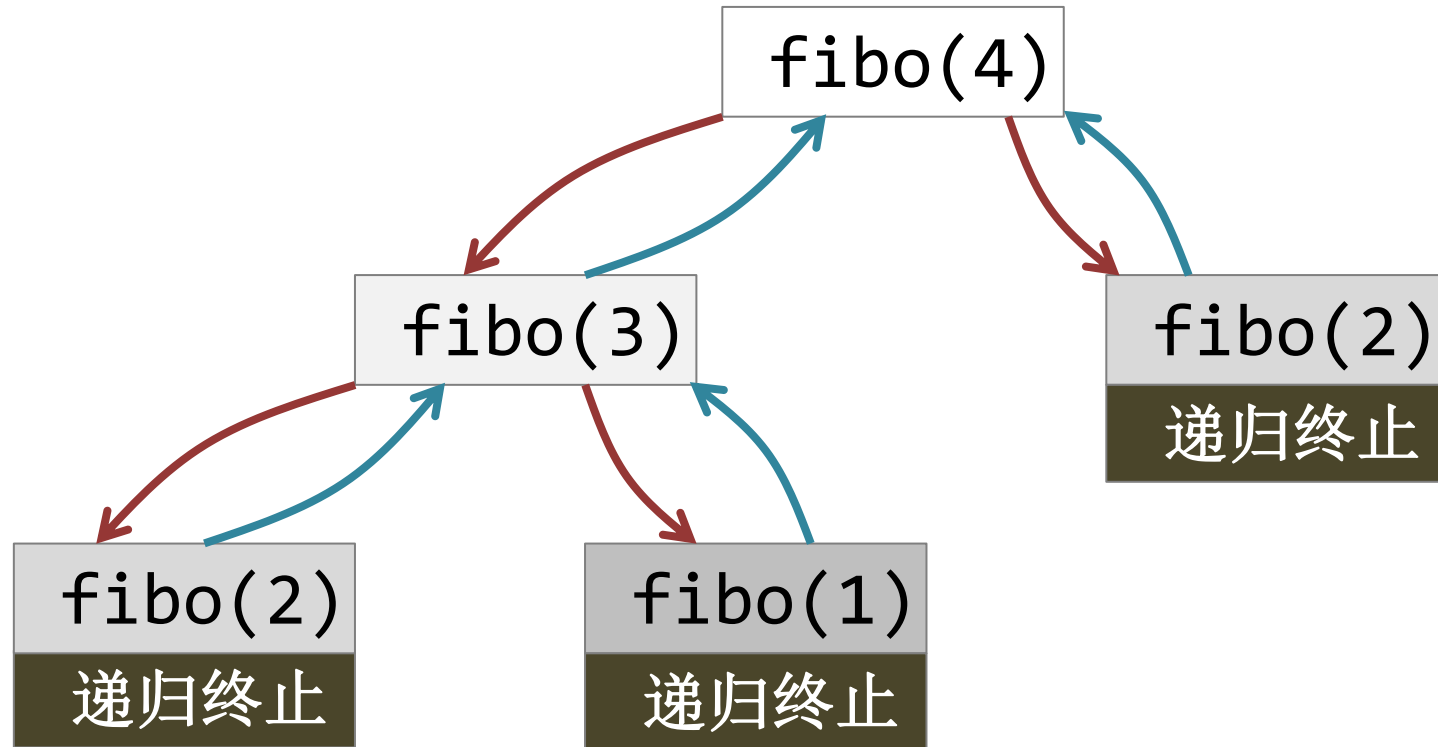
F[x]代表
斐波那契数列第x项

斐波那契数列

```
1  #include<iostream>
2  using namespace std;
3  long long fibo(int x){
4      if(x<=2) return 1;
5      return fibo( )+fibo( );
6  }
7  int main(){
8      int n;
9      cin>>n;
10     cout<<fibo(n);
11     return 0;
12 }
```

fibo(x)代表
斐波那契数列第x项

斐波那契 - 递归过程



最大公约数

如果 a 是 b 的倍数，那么 b 就是 a 的约数。

给定两个数字 x, y ，求 x 和 y 的公共约数里最大是几。

输入样例：

7 8

输出样例：

1

输入样例：

8 12

输出样例：

4

输入样例：

3 9

输出样例：

3

greatest common divisor
简称GCD

最大公约数

```
1  #include<iostream>
2  #include<algorithm>
3  using namespace std;
4  int main(){
5      int a,b,i;
6      cin>>a>>b;
7      for(i=min(a,b);i>=1;i--){
8          if(a%i==0&&b%i==0) break;
9      }
10     cout<<i<<endl;
11     return 0;
```


最大公约数

辗转相除法，即欧几里德算法Euclidean algorithm，是求两个正整数之最大公因数的算法。它是已知最古老的算法，其可追溯至公元前300年前。

求520和250的
最大公约数

a	b	a%b
520	250	20
250	20	10
20	10	0

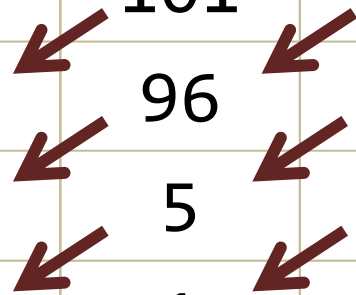
The diagram illustrates the steps of the Euclidean algorithm for finding the GCD of 520 and 250. It shows a table with three columns: 'a', 'b', and 'a%b'. The rows represent the state of the algorithm at each step. Red arrows indicate the sequence of values: from (520, 250) to (250, 20) to (20, 10) to (10, 0).

最大公约数

辗转相除法，即欧几里德算法Euclidean algorithm，是求两个正整数之最大公因数的算法。它是已知最古老的算法，其可追溯至公元前300年前。

求96和101的最大公约数

a	b	a%b
96	101	96
101	96	5
96	5	1
5	1	0



最大公约数

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      int a,b,r;
5      cin>>a>>b;
6      while(r=a%b){
7          a=b;
8          b=r;
9      }
10     cout<<b<<endl;
11     return 0;
12 }
```

第1步

把a除以b的余数赋值给r

第2步

判断当r非零时不断循环

最大公约数

```
3 int gcd(int x,int y){
4     int r;
5     while(r=x%y){
6         x=y;
7         y=r;
8     }
9     return y;
10 }
11 int main(){
12     int a,b;
13     cin>>a>>b;
14     cout<<gcd(a,b)<<endl;
15     return 0;
16 }
```

gcd(x)代表x和y的
最大公约数

最大公约数

辗转
相除

$\text{gcd}(x)$ 代表 x 和 y 的
最大公约数

```
3 int gcd(int x,int y){  
4     if(x%y==0) return y;  
5     return gcd(y,x%y);  
6 }
```

递归函数

最大公约数

```
3 int gcd(int x,int y){  
4     return y?gcd(y,x%y):x;  
5 }
```

判断条件X

X **?** **A** **:** **B**如果X为true
就调用**A**如果X为false
就调用**B**三元
运算

魔鬼的步伐

计数问题

魔鬼共有 n 级楼梯要走，魔鬼有他的步伐，每一步他只可以向上走 a 级楼梯或者 b 级楼梯，请问共有多少种不同的走法可以正好走完 n 级台阶。

输入样例：

4 3 5

输出样例：

0

输入样例：

10 2 5

输出样例：

2

$go(x)$ 代表魔鬼正在第 x 级台阶
继续往前走的过程

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int ans=0,n,a,b;
4  void go(int x){
5      if(x==n) ans++;
6      if(x>=n) return;
7      go(x+a);
8      go(x+b);
9  }
10 int main(){
11     freopen("steps.in","r",stdin);
12     freopen("steps.out","w",stdout);
13     cin>>n>>a>>b;
14     go(0);
15     cout<<ans<<endl;
16     return 0;
17 }
```

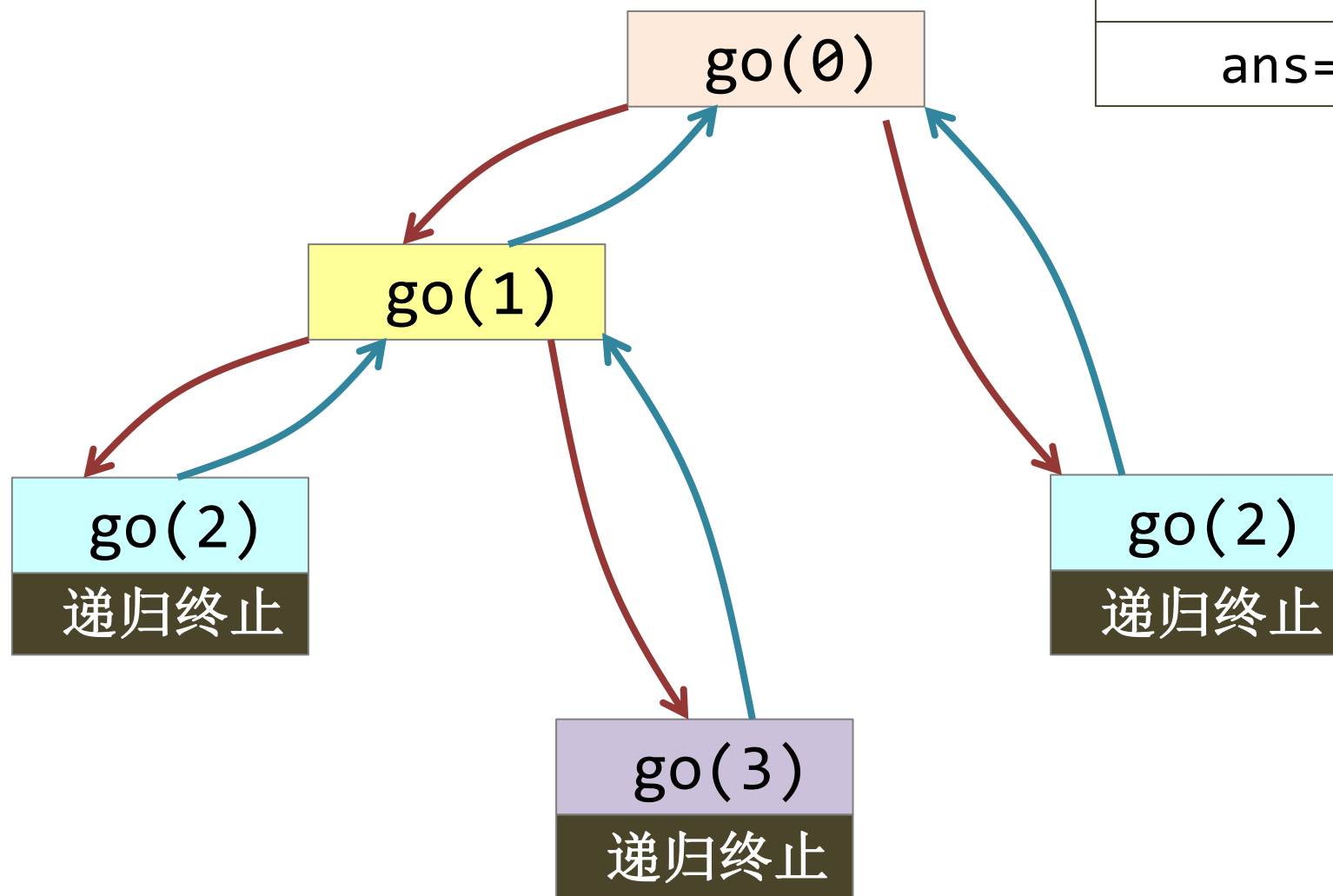
go(x)代表魔鬼正在第x级台阶
继续往前走的过程

$n=2, a=1, b=2$

$ans=0$

$ans=1$

$ans=2$



计算f(0) 局部变量x为0

→ 1 if(x==n) ans++;
→ 2 if(x>=n) return;
→ 3 go(x+a);
→ 4 go(x+b);

返回

n=2, a=1, b=2

ans=0

ans=1

计算f(1) 局部变量x为1

→ 1 if(x==n) ans++;
→ 2 if(x>=n) return;
→ 3 go(x+a);
→ 4 go(x+b);

返回

计算f(2) 局部变量x为2

→ 1 if(x==n) ans++;
→ 2 if(x>=n) return;
3 go(x+a);
4 go(x+b);

返回

计算f(3) 局部变量x为3

→ 1 if(x==n) ans++;
→ 2 if(x>=n) return;

计算go(0) 局部变量x为0

1 if(x==n) ans++;
2 if(x>=n) return;
3 go(x+a);
4 go(x+b);

n=2, a=1, b=2

ans=0

ans=1

ans=2

计算go(1) 局部变量x为1

1 if(x==n) ans++;
2 if(x>=n) return;
3 go(x+a);
4 go(x+b);

返回

计算go(2) 局部变量x为2

1 if(x==n) ans++;
2 if(x>=n) return;
3 go(x+a);
4 go(x+b);

计算go(3) 局部变量x为3

1 if(x==n) ans++;
2 if(x>=n) return;

太戈编程

本次作业都需要用递归方法撰写

重点翻译函数含义

83

118

117

拓展题

305, 129, 128