## 斐波那契数列

1. def f(n):

a,b=1,1

for i in range(n-1):

a,b=b,a+b

return a

1. #使用递归

def fib(n):

if n==1 or n==2:

return 1

return fib(n-1)+fib(n-2)

# 输出了第10个斐波那契数列

print (fib(10))

## 冒泡排序

比较相邻元素，若后>前则交换，对每一组相邻元素做相同工作，从开始到结束，做完后末尾的元素就是最大值。针对除了末尾元素外所有的元素重复步骤直到没有元素需要对比

## 简单选择排序

每一次在n-i+1个记录中选最小的记录并和第i个交换

例：第一次排序时（i=1）从n-1+1个记录中找到最小的值与第1个交换，第2次（i=2）从n-2+1个记录中找最小的与第2个交换直到整个序列呈有序排列

## 快速排序



## 网络号与主机号计算

▲异或：同0异1

▲相与：同1为1，不同为0

▲相或：有1为1，否则为0

例：IP:192.168.1.135,子网掩码：255.255.255.192

将ip和子网掩码转为2进制

IP: 11000000.10101000.00000001.10000111

子网掩码：11111111.11111111.11111111.11000000

求网络号：直接相与 得 192.168.1.128

求子网号：先判断是哪类地址，此处因IP在192—223之间，顾为C类地址，前两个字节为子网号，得192.168.1.2（10的十进制就是2）

求主机号：子网掩码取反后相与

OR

后六位000111 转为十进制就是7

192.168.1.7

故主机号(子网掩码取反后相与)结果为：11111111.11111111.11111111.00000111，转换为10进制为192.168.1.7

例题：某校园网地址为186.115.192.0/21，若将该网络分成30个子网，则子网掩码为？

分析：原子网掩码（网络位21位+主机位11位）：11111111.11111111.11111000.00000000

现需要分成30个子网，25=32>30，则需要向主机位借5位

给定的网络是186.115.192.0/21，默认情况下有21位用于网络部分。

从主机部分借用5位后，新的子网掩码有：

网络部分：21位（原有的） + 5位（借用的）= 26位

新的子网掩码为/26，对应的十进制表示为：

二进制形式：11111111.11111111.11111111.11000000

十进制形式：255.255.255.192

结果

因此，将网络186.115.192.0/21分成30个子网的子网掩码是 255.255.255.192（/26）。

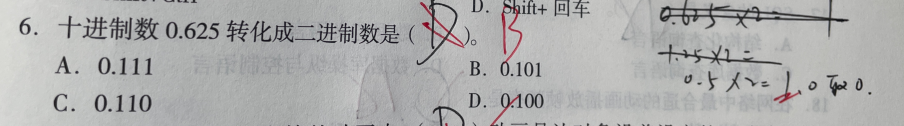
## 进出栈

1进栈0出栈，进栈顺序是1，2，3，4，5，6，为了得到出栈序列325461，操作为？

123进32出，45进54出，6进6出，1出

所以操作为111001100100

## 十进制转二进制（小数点）



0.625×2=1.25 取1

0.25×2=0.5 取0

0.5×2=1.0 取1

故 (0.625)10=(0.101)2

## 输出质数

输出100以内质数，用’,’隔开

def is\_prime(n):  
 if n<=1:  
 return False  
 #2到n的平方根之间的任意一个数如果能整除n那么n就不是质数  
 #任意一个非质数都能被一个小于或等于它的平方根的数整除  
 for i in range(2,int(pow(n,2)+1)):  
 if n%i==0:  
 return False  
 return True  
prime\_num=[str(num) for num in range(2,101) if is\_prime(num)]  
print(','.join(prime\_num))

先序：根左右

中序：左根右

后序：左右根