# 快速幂

## 暴力求幂

## 普通快速幂

## 多进制快速幂

## 矩阵快速幂

# 问题模型和优化

矩阵求斐波那契数列以广义斐波那契数列的通解

## **广义斐波那契数列**

### 例题和思路：

基准时间限制：1 秒 空间限制：131072 KB 分值: 0 [难度：基础题](http://www.51nod.com/onlineJudge/problemList.html#!groupId=1)

斐波那契数列的定义如下：

F(0) = 0

F(1) = 1

F(n) = F(n - 1) + F(n - 2) (n >= 2)

(1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, ...)

给出n，求F(n)，由于结果很大，输出F(n) % 1000000009的结果即可。

**Input**

输入1个数n(1 <= n <= 10^18)。

**Output**

输出F(n) % 1000000009的结果。

**Input示例**

11

**Output示例**

89

数据很大，无法用递推的方法去求解

事实上正解是通过矩阵快速幂，logn\*N\*N\*N的时间复杂度即可求得答案,很快,N代表正方形矩阵的边长，N\*N 的矩阵相乘复杂度是N^3

设斐波那契额数列第i项是F[i],且F[0]=0 F[1]=1，

i>=2时,F[i]= F[i-1]+F[i-2];

设矩阵m是

m=

存在矩阵m使得：

F[i+1] \*m=F[i+2]

[ ] \* = [ ]

根据矩阵相乘法则；

第一个矩阵行乘第二个矩阵列，  
F[i]\*+ F[i-1]\*=F[i+1]

F[i]\*+ F[i-1]\*=F[i]

这2个等式成立，a应该满足什么呢？

F[i]和F[i-1]推导出F[i+1]显然是1倍的F[i]加上1倍的F[i-1]

F[i]和F[i-1]推导出F[i]显然是1倍的F[i]加上0倍的F[i-1]

=1,=1 ，=1 =0

m=

那么对于F[0]和F[1]构成的矩阵，乘以一个m得到

F[1]和F[2],乘以两个m得到F[2]和F[3]

[ ] \* = [ ]

m 的n次方用矩阵快速幂求，和普通快速幂思路一样

因此斐波那契数列的第n项就用[ ] \*

输出结果矩阵[ ]的第0个数F[n]

### 通解

F[i]是广义的斐波那契数列第i项，F[i]= K+A\*F[i-1]+B\*F[i-2]+C\*F[I-3]+….;

A,B,C代表常数，K是一个任意与F无关的式子，可能是常数，可能是普通式子。数列的前几项给定

举个例子：F[i]= i+A\*F[i-1]+B\*F[i-2]+C\*F[i-3]+1;

这类问题还是矩阵快速幂，假设递推式有len项，那么我们构造一个len\*len的矩阵,

设存在矩阵m使得：F[i] \*m=F[i+1]

在例子中：F[i]= i+A\*F[i-1]+B\*F[i-2]+C\*F[I-3]+1;中就构造：

F[n]= 乘以

等于

=F[n+1]

i\*+\*+\*+\*+=

i\*+\*+\*+\*+=

i\*+\*+\*+\*+=

i\*+\*+\*+\*+=

i\*+\*+\*+\*+=

解出m矩阵

### 题目1：

链接：<https://www.nowcoder.com/acm/contest/105/G>  
来源：牛客网

题目描述

这是一个加强版的斐波那契数列。

给定递推式 F[i]=F[i-1]+F[i-2]+i^3+i^2+i+1

F[0]=0;F[1]=1;

求F(n)的值，由于这个值可能太大，请对109+7取模。

输入描述:

第一行是一个整数T(1 ≤ T ≤ 1000)，表示样例的个数。  
以后每个样例一行，是一个整数n(1 ≤ n ≤ 1018)。

输出描述:

每个样例输出一行，一个整数，表示F(n) mod 1000000007。

正解就是矩阵快速幂，也很容易想到矩阵快速幂，关健是怎么去得到矩阵

看到有6项那就设一个6\*6的矩阵

乘以

等于

按照递推关系，第一个矩阵行乘第二个矩阵列

最后能得到一下矩阵，这个过程中(i+1)^3,分解得到i^3+3\*i^2+3\*i+1,注意系数是3

m=

后面解法就不用说了

## 欧拉降幂

费马-欧拉定理：

若n与a都是正整数且互质(gcd(n,a)=1)：( a^phi[n] )%n == 1

对于三个任意非负整数a,b,n,有欧拉降幂公式：

%n=

当时，显然b很小，用普通快速幂计算即可

### 经典题目

计蒜客41299

大意：x=0时,f(x)=1，x=1时,f(x)=x，x>=2时,f(x)=x^f(x-1)

这个是超指数函数，递归的欧拉降幂即可，由于一个数反复求欧拉函数的复杂度是logn级别，所以欧拉降幂的递归深度不会太大。且判断 很容易，这里的“b”代表f(x-1)，phi[n]很小1e6以内，仅仅有限的几个数的超指数函数值可能比phi[n]小，暴力枚举娇小的几个超指数函数值就好。