利用矩阵乘法优化快速求斐波那契数列

wcz

2018-02-04

目录

1 前言

为了方便表示, 我们称斐波那契数列的第 x 项称为 $Fib(x)$. 这篇文章可能会系统的介绍一下线性代数的部分内容, 用'*' 标记.

2 为什么需要有更优秀的方法求解斐波那契数列

对于斐波那契数列, 可以通过递归和递推求解. 前者需要计算 2^n 次, 后者需要 n 次. 可是如果求解的是任意第 n 项或者是前 n 项怎么办.

这时我们便需要寻找更快的方法.

3 如何用矩阵乘法优化求斐波那契数列

3.1 将递推变成若干次矩阵乘法

考虑其递推式,

$$Fib(x) = Fib(x-1) + Fib(x-2) \tag{1}$$

学过代数的同学们可能会发现, 式 (1) 中的 $Fib(x)$和$Fib(x-1)$ 满足,

$$\begin{bmatrix} Fib(x) \\ Fib(x-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} Fib(x-1) \\ Fin(x-2) \end{bmatrix} \tag{2}$$

那么, 我们亦可以通过这个来递推斐波那契数列, 每递推一次进行一次矩阵乘法, 这样递推效率和前面的普通的递推相同.

3.2 将矩阵乘法表示成矩阵幂的形式

为了方便后面的推导. 不如直接将式 (2) 中间的矩阵定义为 $\mathbf{A}$.

有没有办法加快进行 n 次相同的矩阵乘法的效率呢?

需要从矩阵乘法的性质入手. 矩阵乘法不满足交换律但是满足结合律和分配率. 这是一个显而易见的结论, 因为矩阵乘法的本质是线性变换.

这表明在进行若干次矩阵乘法, 例如:

$$\begin{bmatrix} Fib(k) \\ Fib(k-1) \end{bmatrix} = \begin{bmatrix} Fib(1) \\ Fib(2) \end{bmatrix} \times \mathbf{A} \times \mathbf{A} \cdots \mathbf{A} \tag{3}$$

因为矩阵乘法满足结合律, 所以我们可以任意改变乘法的顺序, 甚至可以表示成矩阵幂的形式. 式 (3) 可以表示成以下形式.

$$\begin{bmatrix} Fib(k) \\ Fib(k-1) \end{bmatrix} = \begin{bmatrix} Fib(1) \\ Fib(2) \end{bmatrix} \times (\mathbf{A} \times \mathbf{A} \cdots \mathbf{A}) = \begin{bmatrix} Fib(1) \\ Fib(2) \end{bmatrix} \times \mathbf{A}^k \tag{4}$$

3.3 矩阵快速幂

有了矩阵的幂, 就需要考虑矩阵的幂运算满足的性质. 因为矩阵乘法满足结合律, 所以矩阵的幂运算满足

$$\mathbf{A}^m\mathbf{A}^n = \mathbf{A}^{m+n}, (\mathbf{A}^m)^n = \mathbf{A}^{mn} \tag{5}$$

有了这些性质, 足够我们发现一种快速求解矩阵幂的方法了.

考虑我们前面学过的快速幂算法, 是将连续若干次相乘分治 [1] 来进行处理. 这种思想可以迁移到对于矩阵幂的处理上, 被称为矩阵快速幂.

同快速幂一样, 矩阵快速幂拥有几乎完全相同的步骤. 如果需要求解 $\mathbf{A}^k$

```
class Matrix
Matrix A, ans
while k!=0:
        if k/2*2=k:
                ans=A*ans
        A=A*A
        k/=2
```

这是写快速幂常见的一种写法. 容易看出其时间复杂度为 $O(\lg k)$

4 有关线代的部分内容*

线性代数大概是非常让人头疼的一门课程了, 就从我之前对线性代数的浅薄的了解, 他是一大堆巨算符叠加起来的乱七八糟的运算. 我大概真的不知道这些东西有什么用, 当然这跟线性代数这门学科的难度有关, 也和部分教材的舍本取末有关, 就拿国内很普及的也是我正看的同济大学的《线性代数》, 在读者领悟了其内容真正含义之前早就被一大堆混乱的运算吓跑了.

线代涉及的一些名词

- 线性方程组
- 线性空间
- 向量空间
- 线性变换

[1] 分治这是一种通过对问题的子问题进行分别处理从而加快效率的算法, 在信息学竞赛中应用非常广泛.

- 线性相关
- 线性无关
- 特征向量
- 特征值
- 行向量
- 列向量
- 行列式
- 空间积
- 维数
- 向量
- 矩阵
- 变换
- 坐标
- 内积
- 阶
- 基
- 秩

4.1 定义

可以说空间就是一个满足某些性质的集合, 而线性空间的元素是向量, 因此也被称为向量空间.

定义 4.1.1 (线性空间) , *设 V 是一个非空集合, R 为实数域. 如果对于任意两个元素 $\alpha, \beta \in V$, 总有唯一的一个元素 $\gamma \in V$ 与其对应, 称为 α 与 β 的和, 记为 $\gamma = \alpha + \beta$; 又对于任一数 $\lambda \in R$ 与任一元素 $\alpha \in V$, 总有唯一的一个元素 $\delta \in V$ 与之对应, 称为 λ 与 α 的积, 记做 $\delta = \lambda\alpha$; 并且这两种运算满足以下八条运算规则 (设 $\alpha, \beta, \gamma \in V, \lambda, \mu \in R$):*

(i) $\alpha+\beta=\beta+\alpha$*;*

(ii) $(\alpha+\beta)+\gamma=\alpha+(\beta+\gamma)$*;*

(iii) 在 V 中存在零元素 *0,* 对任何 $\alpha\in V$*,* 都有 $\alpha+0=\alpha$*;*

(iv) 对任何 $\alpha\in V$*,* 都有 α 的负元素 $b\in V$*,* 使 $\alpha+\beta=0$*;*

(v) $1\alpha=\alpha$*;*

(vi) $\lambda(\mu\alpha=(\lambda\mu)\alpha$*;*

(vii) $(\lambda+\mu)\alpha=\lambda\alpha+\mu\alpha$*;*

(viii) $\lambda(\alpha+\beta)=\lambda\alpha+\lambda\beta$*,*

那么*,* V 就被称为 *(*实数域 R 上的*)* 向量空间 *(*或线性空间*),*V 中的元素无论其本来的性质如何*,* 统称为 *(*实*)* 向量*.* 我们发现这个集合 *(*空间*)* 的运算是封闭的*,* 即 α 与 β 的积和和*,* 都是属于 V 的元素*.*

定义 4.1.2 在线性空间 V 中*,* 如果存在 n 个元素 $\alpha_1,\alpha_2\cdots,\alpha_n$ 满足*:*

(i) $\alpha_1,\alpha_2\cdots,\alpha_n$ 线性无关*;*

(ii) V 中任意元素总可以用 $\alpha_1,\alpha_2\cdots,\alpha_n$ 线性表示*;*

那么*,* $\alpha_1,\alpha_2\cdots,\alpha_n$ 就称为是线性空间 V 的一个基*,* n 称为线性空 s 间 V 的维数*.*

定义 4.1.3 (线性变换) 设 V_n,U_m 分别是实数域上的 n 维和 m 维线性空间*,* T 是一个从 V_n 到 U_m 的映射*,* 如果映射 T 满足

(i) 任给 $\alpha_1,\alpha_2\in V_n$*, (*从而 $\alpha_1+\alpha_2\in V_n$*),* 有 $T(\alpha_1+\alpha_2)=T(\alpha_1)+T(\alpha_2)$*;*

(ii) 任给 $\alpha\in V_n,k\in R$*, (*从而 $k\alpha\in V_n$*),* 有 $T(k\alpha)=kT(\alpha)$*,*

那么*,* T 就称为从 V_n 到 U_m 的线性映射*,* 或称为线性变换*.*

上述有关定义是完整的定义*.* 我们一般使用的都是数组向量空间*,* 也就是我们平时所提到的向量空间*.*

定义 4.1.4 如果一个向量组 $A:\alpha_1,\alpha_2,\cdots\alpha_n$ 存在一组 $k_1,k_2,\cdots,k_n$*,* 使得

$$k_1\alpha_1+k_2\alpha_2+\cdots k_n\alpha_n=0$$

则称向量组 A 是线性相关的*,* 否则称为线性无关*.*

如果向量组 A 中有某个向量能由其余 $m-1$ 个向量线性表示, 那么向量组就是线性相关的. 上述概念可以迁移到线性方程组, 如果线性方程组中一组方程是其余方程的线性组合, 那么这个方程就是多余的, 这时方程组是线性相关的; 反之为线性无关.