线性代数出土

skc

2023 年 8 月 15 日

这玩意儿有什么用

~~这玩意儿没有用（确信~~

这玩意儿有什么用

~~这玩意儿没有用（确信~~

虽然**线性代数**几乎不会在联赛中出现，但在省选及以上的比赛中，线性代数的出现频率并不低。

这玩意儿有什么用

~~这玩意儿没有用（确信~~

虽然**线性代数**几乎不会在联赛中出现，但在省选及以上的比赛中，线性代数的出现频率并不低。

线性代数本身是一个数学内容，所以它的结构比较完整，内容关联性较强。如果能学好线性代数，并形成较为完整的数学体系，那么在解决相关问题时还是非常有用的。

参考

本课件参考了 Michael Artin 的《Algebra》(Second Edtion)，3Blue1Brown 的“线性代数的本质”系列视频，以及 yhx-12243 的两份线性代数课件。

参考

本课件参考了 Michael Artin 的《Algebra》(Second Edtion)，3Blue1Brown 的“线性代数的本质”系列视频，以及 yhx-12243 的两份线性代数课件。

由于篇幅原因，本课件不可能将线性代数的基础部分讲的非常仔细。如果你需要自学线性代数基础，这里推荐观看 3b1b 的视频（B 站上有官方翻译版本），可以形成一套完整的对线性代数的直观理解。之后，可以阅读《Algebra》的第 $1, 2, 4$ 章（中文版电子书已上传至soj/uploads/realskc/代数.pdf），该书会严谨而友好地完善线性代数的理论基础，如果想学习抽象代数也可以阅读该书，该书很适合作为代数入门书。

参考

本课件参考了 Michael Artin 的《Algebra》(Second Edtion), 3Blue1Brown 的“线性代数的本质”系列视频，以及 yhx-12243 的两份线性代数课件。

由于篇幅原因，本课件不可能将线性代数的基础部分讲的非常仔细。如果你需要自学线性代数基础，这里推荐观看 3b1b 的视频（B 站上有官方翻译版本），可以形成一套完整的对线性代数的直观理解。之后，可以阅读《Algebra》的第 $1, 2, 4$ 章（中文版电子书已上传至soj/uploads/realskc/代数.pdf），该书会严谨而友好地完善线性代数的理论基础，如果想学习抽象代数也可以阅读该书，该书很适合作为代数入门书。

但 3b1b 的视频几乎涵盖了所有的线性代数基础，所以如果你对理论基础和抽象代数没有兴趣，也完全可以不阅读此书。

Warning

假如你购买了同济大学数学系编的紫色封面的《工程数学线性代数》，请立即将其扔进垃圾桶或卖给废品收购处。

Warning

假如你购买了同济大学数学系编的紫色封面的《工程数学线性代数》，请立即将其扔进垃圾桶或卖给废品收购处。

出于缩减课时安排，压缩教师授课时长（以占用学生课后精力为代价）等原因，这本书的课程编排顺序极度不合理。无论是自学还是他人教学，这本书都不适合作为教材使用，且这本书相比于其它书也毫无特长。

由于这本书对学生极度不负责任，因此通过这本书来学习线性代数（尤其是自学）会有非常大的负收益，包括但不限于消耗更长时间，更加不容易理解，更容易思维混乱等等。

除非某课程使用这本书作为教材且你不得不上这门课，否则建议直接扔掉这本书。

Q&A

Q: 为什么是你来讲

Q&A

Q: 为什么是你来讲
A: 因为鱼大跑路了

Q&A

Q: 为什么是你来讲
A: 因为鱼大跑路了
Q: 为什么你要讲这个

Q&A

Q: 为什么是你来讲
A: 因为鱼大跑路了
Q: 为什么你要讲这个
A: 因为跑路的是鱼大

Q&A

Q: 为什么是你来讲
A: 因为鱼大跑路了
Q: 为什么你要讲这个
A: 因为跑路的是鱼大
Q: 课件为什么写得那么屑

Q&A

Q: 为什么是你来讲
A: 因为鱼大跑路了
Q: 为什么你要讲这个
A: 因为跑路的是鱼大
Q: 课件为什么写得那么屑
A: 欢迎大佬来指导我写课件

Q&A

Q: 为什么是你来讲
A: 因为鱼大跑路了
Q: 为什么你要讲这个
A: 因为跑路的是鱼大
Q: 课件为什么写得那么屑
A: 欢迎大佬来指导我写课件
Q: 内容太难了怎么办

Q&A

Q: 为什么是你来讲
A: 因为鱼大跑路了
Q: 为什么你要讲这个
A: 因为跑路的是鱼大
Q: 课件为什么写得那么屑
A: 欢迎大佬来指导我写课件
Q: 内容太难了怎么办
A: 你可以坐到前面来问我

Q&A

Q: 为什么是你来讲
A: 因为鱼大跑路了
Q: 为什么你要讲这个
A: 因为跑路的是鱼大
Q: 课件为什么写得那么屑
A: 欢迎大佬来指导我写课件
Q: 内容太难了怎么办
A: 你可以坐到前面来问我
Q: 内容太水了怎么办

Q&A

Q: 为什么是你来讲
A: 因为鱼大跑路了
Q: 为什么你要讲这个
A: 因为跑路的是鱼大
Q: 课件为什么写得那么屑
A: 欢迎大佬来指导我写课件
Q: 内容太难了怎么办
A: 你可以坐到前面来问我
Q: 内容太水了怎么办
A: 你可以坐到前面来替我回答问题

线性函数

线性一般指一个函数的性质

线性函数

线性一般指一个函数的性质

Definition 1.2.1 (线性)

函数 $f(x)$ （在实数域上）是线性的，当且仅当：对于任意 x, y, c，有 $f(x+y) = f(x) + f(y)$ 和 $f(cx) = cf(x)$（c 是任意实数）。

线性函数

线性一般指一个函数的性质

Definition 1.2.1 (线性)

函数 $f(x)$ （在实数域上）是线性的，当且仅当：对于任意 x, y, c，有 $f(x+y) = f(x) + f(y)$ 和 $f(cx) = cf(x)$（c 是任意实数）。

对于值域和定义域都是 $\mathbb{R}$ 的情况，一定有 $f(x) = kx$ 成立，也就是说 $\mathbb{R} \to \mathbb{R}$ 的线性函数一定是正比例函数。

线性函数

线性一般指一个函数的性质

Definition 1.2.1 (线性)

函数 $f(x)$ （在实数域上）是线性的，当且仅当：
对于任意 x, y, c，有 $f(x+y)=f(x)+f(y)$ 和 $f(cx)=cf(x)$（c 是任意实数）。

对于值域和定义域都是 $\mathbb{R}$ 的情况，一定有 $f(x)=kx$ 成立，也就是说 $\mathbb{R}\to\mathbb{R}$ 的线性函数一定是正比例函数。

注意：假如线性的条件中只有前一条而没有后一条，上述结论其实是不成立的。

标量

其实刚才的定义不是很严谨，主要原因是我们没有区分“数”和其它可运算元素，我们来重新定义一下。

标量

其实刚才的定义不是很严谨，主要原因是我们没有区分“数”和其它可运算元素，我们来重新定义一下。

首先定义“数”，“数”就是通常意义下的数，有时也被称为标量 (scalar)，英文字面意思就是缩放比。

我们要选取一个域作为数域，例如 $\mathbb{Q}$ 或 $\mathbb{R}$，然后矩阵和向量之类的其它可运算元素都会被定义在这个数域上。

标量

其实刚才的定义不是很严谨，主要原因是我们没有区分“数”和其它可运算元素，我们来重新定义一下。

首先定义“数”，“数”就是通常意义下的数，有时也被称为标量 (scalar)，英文字面意思就是缩放比。

我们要选取一个域作为数域，例如 $\mathbb{Q}$ 或 $\mathbb{R}$，然后矩阵和向量之类的其它可运算元素都会被定义在这个数域上。

比如说我们现在选取实数作为数域，那么“数”就指代全体实数。在之后的讨论中，如果没有特别指出，则数域默认是实数域。

我们对数的要求是构成域，即可以进行加法和乘法，有所有结合律、交换律、分配律，所有元素有加法逆，除 0 外的所有元素有乘法逆。

这其实是个很严格的要求，不过一般情况下数域都是指定的，所以不需要考虑数域的限制。

我们对数的要求是构成域，即可以进行加法和乘法，有所有结合律、交换律、分配律，所有元素有加法逆，除 0 外的所有元素有乘法逆。

这其实是个很严格的要求，不过一般情况下数域都是指定的，所以不需要考虑数域的限制。

对于其它可运算的元素，也有一些限制。例如我们要求元素能够进行数乘，数乘就是指元素和数进行的乘法，例如向量和数的乘法，注意数乘是不需要区分左乘和右乘的。我们还要求同类型元素能够进行加法，例如同样大小的矩阵能够相加。

我们对数的要求是构成域，即可以进行加法和乘法，有所有结合律、交换律、分配律，所有元素有加法逆，除 0 外的所有元素有乘法逆。

这其实是个很严格的要求，不过一般情况下数域都是指定的，所以不需要考虑数域的限制。

对于其它可运算的元素，也有一些限制。例如我们要求元素能够进行数乘，数乘就是指元素和数进行的乘法，例如向量和数的乘法，注意数乘是不需要区分左乘和右乘的。我们还要求同类型元素能够进行加法，例如同样大小的矩阵能够相加。

还有一些限制，例如数乘结合律和数乘分配律（名字是我瞎取的）。假如 $\vec{v}$ 是元素，a 和 b 是数，我们要求：

$$(ab)\vec{v} = a(b\vec{v})$$

$$(a+b)\vec{v} = a\vec{v} + b\vec{v}$$

线性

我们来重新定义一下“线性”

线性

我们来重新定义一下“线性”

Definition 1.2.2 (线性)

我们称函数 $f(x)$ 是线性的，当且仅当：
对于任意 x, y, c，有 $f(x+y) = f(x) + f(y)$ 和 $f(cx) = cf(x)$。
其中 c 是“数”，x 和 $f(x)$ 均为“可运算的元素”。

线性

我们来重新定义一下“线性”

Definition 1.2.2 (线性)

我们称函数 $f(x)$ 是线性的，当且仅当：
对于任意 x, y, c，有 $f(x+y) = f(x) + f(y)$ 和 $f(cx) = cf(x)$。
其中 c 是“数”，x 和 $f(x)$ 均为“可运算的元素”。

这个定义的核心内容跟前一个没什么区别，不过这回明确了定义域和值域。

我们来看一些线性函数（映射）的例子：

我们来看一些线性函数（映射）的例子：

Example 1.2.1

令 $f(x) = 114514x$，则 f 为线性函数。

我们来看一些线性函数（映射）的例子：

Example 1.2.1

令 $f(x) = 114514x$，则 f 为线性函数。

Example 1.2.2

令 S 为 wzf 买的东西，$f(S)$ 表示他买 S 所花的钱，则 f 是线性函数。

我们来看一些线性函数（映射）的例子：

Example 1.2.1

令 $f(x) = 114514x$，则 f 为线性函数。

Example 1.2.2

令 S 为 wzf 买的东西，$f(S)$ 表示他买 S 所花的钱，则 f 是线性函数。

在这个例子中，我们定义东西的加法表示将两堆东西合并，东西乘上实数 c 表示将这堆东西复制成为原来的 c 倍（部分情况可以当成对现实情况的扩展）。

前面的内容没看懂没关系

前面的内容没看懂没关系

因为接下来更加抽象

代数的定义

代数是一个较为基础的数学分支。它的研究对象有许多。诸如数、数量、代数式、关系、方程理论、代数结构等等都是代数学的研究对象。初等代数一般在中学时讲授，介绍代数的基本思想：研究当我们对数字作加法或乘法时会发生什么，以及了解变量的概念和如何建立多项式并找出它们的根。代数的研究对象不仅是数字，还有各种抽象化的结构。例如整数集作为一个带有加法、乘法和序关系的集合就是一个代数结构。在其中我们只关心各种关系及其性质，而对于“数本身是什么”这样的问题并不关心。常见的代数结构类型有群、环、域、模、线性空间等。

——摘自维基百科

在上一页中也写到了，代数所关注的是元素之间的关系而非元素本身，是一些更为本质的东西，称作代数结构。

在上一页中也写到了，代数所关注的是元素之间的关系而非元素本身，是一些更为本质的东西，称作代数结构。

例如我们在幼儿园时就学过，对于我们熟知的自然数而言，有 $1+1=2$。这个算式的成立，完全依赖于 1 和 2 这两个数的性质以及全体自然数的性质，所以一般不认为 $1+1=2$ 是一个代数定理。

在上一页中也写到了，代数所关注的是元素之间的关系而非元素本身，是一些更为本质的东西，称作代数结构。

例如我们在幼儿园时就学过，对于我们熟知的自然数而言，有 $1+1=2$。这个算式的成立，完全依赖于 1 和 2 这两个数的性质以及全体自然数的性质，所以一般不认为 $1+1=2$ 是一个代数定理。

但由于代数的定义本身就是模糊的，所以我们没法明确区分代数和非代数。

不过我们还是可以通过具体的例子来形成大概的感觉，比如下面这个问题就是一个非常“代数”的问题。

Example 1.3.1

给定一个数的集合，并且在该集合上定义了加法和乘法两种运算，它们关于这个集合都是封闭的。$a = b$ 仍然表示 a 和 b 相同，因此等号不需要重新定义。

我们把加法单位元记为 0，乘法单位元记为 1。集合中的所有元素都有加法逆元（a 的加法逆元记为 $-a$）。

该集合上还有加法结合律，乘法结合律，乘法分配律成立。

求证加法交换律成立。

Problem 1.3.1

对于在该集合上封闭的加法和乘法运算，有：

1. 存在 0，满足对于任意 a，都有 $0+a=a+0=a$（**加法单位元**）

2. 存在 1，满足对于任意 a，都有 $1\times a=a\times 1=a$（**乘法单位元**）

3.$a+b+c=a+(b+c)$（**加法结合律**）

4.$a\times b\times c=a\times(b\times c)$（**乘法结合律**）

5. 对任意 a，存在 $-a$ 满足 $a+(-a)=(-a)+a=0$（**加法逆元**）

6.$c\times(a+b)=c\times a+c\times b$（**左分配律**）

7.$(a+b)\times c=a\times c+b\times c$（**右分配律**）

求证： $a+b=b+a$（**加法交换律**）

Proof 1.3.1

我们考虑计算 $(a+b)\times(1+1)$

Proof 1.3.1

我们考虑计算 $(a+b)\times(1+1)$

如果先展开左边，后展开右边，则有

$$(a+b)\times(1+1)=a\times(1+1)+b\times(1+1)=a+a+b+b$$

Proof 1.3.1

我们考虑计算 $(a+b)\times(1+1)$

如果先展开左边，后展开右边，则有

$$(a+b)\times(1+1)=a\times(1+1)+b\times(1+1)=a+a+b+b$$

如果先展开右边，后展开左边，则有

$$(a+b)\times(1+1)=(a+b)+(a+b)=a+b+a+b$$

Proof 1.3.1

我们考虑计算 $(a+b)\times(1+1)$

如果先展开左边，后展开右边，则有

$(a+b)\times(1+1)=a\times(1+1)+b\times(1+1)=a+a+b+b$

如果先展开右边，后展开左边，则有

$(a+b)\times(1+1)=(a+b)+(a+b)=a+b+a+b$

所以有 $a+a+b+b=(a+b)\times(1+1)=a+b+a+b$

Proof 1.3.1

我们考虑计算 $(a+b)\times(1+1)$

如果先展开左边，后展开右边，则有

$$(a+b)\times(1+1)=a\times(1+1)+b\times(1+1)=a+a+b+b$$

如果先展开右边，后展开左边，则有

$$(a+b)\times(1+1)=(a+b)+(a+b)=a+b+a+b$$

所以有 $a+a+b+b=(a+b)\times(1+1)=a+b+a+b$

在等式左侧加 $-a$，等式右侧加 $-b$，可得

$$(-a)+a+a+b+b+(-b)=(-a)+a+b+a+b+(-b)$$

即 $a+b=b+a$

以上就是一个代数问题的例子。我们可以从中发现，代数更关心元素之间的关系，运算的性质等等，而很少关心特定的元素或者特定的集合。

以上就是一个代数问题的例子。我们可以从中发现，代数更关心元素之间的关系，运算的性质等等，而很少关心特定的元素或者特定的集合。

不过上述例子更接近抽象代数，线性代数中不会有这么搞脑子的东西，请大家放心。

相信大家都知道向量是什么。

在高中数学中，向量是平面直角坐标系上带箭头的有向线段；在物理中，向量是具有大小和方向的量。

相信大家都知道向量是什么。

在高中数学中，向量是平面直角坐标系上带箭头的有向线段；在物理中，向量是具有大小和方向的量。

但我们是 OIer，在 OI 中，我们不常使用这两种定义方式，所以**请忘记所有和有向线段有关的向量内容**。

相信大家都知道向量是什么。

在高中数学中，向量是平面直角坐标系上带箭头的有向线段；在物理中，向量是具有大小和方向的量。

但我们是 OIer，在 OI 中，我们不常使用这两种定义方式，所以**请忘记所有和有向线段有关的向量内容**。

如果你不知道向量是什么，那么恭喜你，你可以跳过上一行的步骤了。

好的，我们假装大家都已经成功忘记了，那么我们来定义一下向量。

好的，我们假装大家都已经成功忘记了，那么我们来定义一下向量。

Definition 2.1.1 (向量)

向量是一个纵向的列表，列表的每个元素都是一个数。

好的，我们假装大家都已经成功忘记了，那么我们来定义一下向量。

Definition 2.1.1 (向量)

向量是一个纵向的列表，列表的每个元素都是一个数。

比如 $\begin{bmatrix} 1 \\ 1 \\ 4 \end{bmatrix}$ 就是一个向量，由于一般意义下向量是竖着记的，所以有时也把向量称作**列向量**。

好的，我们假装大家都已经成功忘记了，那么我们来定义一下向量。

Definition 2.1.1 (向量)

向量是一个纵向的列表，列表的每个元素都是一个数。

比如 $\begin{bmatrix}1\\1\\4\end{bmatrix}$ 就是一个向量，由于一般意义下向量是竖着记的，所以有时也把向量称作**列向量**。

每个向量都有一个长度，也就是列表的元素个数，~~为了显得更专业一些~~，我们把向量的长度称为向量的**维数**。换句话说，我们一般把长度为 n 的向量称作 n 维向量，例如 $\begin{bmatrix}1\\1\\4\end{bmatrix}$ 就是一个 3 维向量，它的第一维和第二维都是 1，第三维是 4。

既然都说了向量是列表，那我们显然要把它当成列表用。

既然都说了向量是列表，那我们显然要把它当成列表用。
~~事实上大家都拿 vector 当列表用。~~

既然都说了向量是列表，那我们显然要把它当成列表用。
~~事实上大家都拿 vector 当列表用。~~

我们可以用向量来列出一样事物的所有属性，具体来说，我们可以认为某一件事物具有若干属性，每个属性都可以用一个实数来描述。

向量

既然都说了向量是列表，那我们显然要把它当成列表用。
~~事实上大家都拿 vector 当列表用。~~

我们可以用向量来列出一样事物的所有属性，具体来说，我们可以认为某一件事物具有若干属性，每个属性都可以用一个实数来描述。

假设该事物一共有 n 个属性，那么我们就可以用一个 n 维向量来描述这样事物，向量的每个元素对应一种属性。

Example 2.1.1

lhx 是一个逆天男，他的逆天之处在于很喜欢往群里发其它人的照片。

Example 2.1.1

lhx 是一个逆天男，他的逆天之处在于很喜欢往群里发其它人的照片。

我们假设他发的照片只有大老师、妹子、领导人这三种类别，那么我们就可以用一个三维向量来描述 lhx 一天之内发的照片。

比如我们可以用向量的第一维表示 lhx 发的大老师的照片数量，第二维表示 lhx 发的妹子的照片数量，第三维表示 lhx 发的领导人及其卡通形象的照片数量。

向量作为数学对象，自然会有一些运算。

Definition 2.1.2 (向量运算)

对于任意维数相同的向量 $\vec{u}$ 和 $\vec{v}$，假设它们的维度都是 n，则向量的加法定义如下：

$$\vec{u}+\vec{v}=\begin{bmatrix}u_1\\u_2\\\vdots\\u_n\end{bmatrix}+\begin{bmatrix}v_1\\v_2\\\vdots\\v_n\end{bmatrix}=\begin{bmatrix}u_1+v_1\\u_2+v_2\\\vdots\\u_n+v_n\end{bmatrix}$$

向量的数乘运算定义如下：

$$c\vec{v}=c\begin{bmatrix}v_1\\v_2\\\vdots\\v_n\end{bmatrix}=\begin{bmatrix}cv_1\\cv_2\\\vdots\\cv_n\end{bmatrix}$$

在明确了向量运算之后，我们就可以在向量上做其它事情。

在明确了向量运算之后，我们就可以在向量上做其它事情。

我们可以定义一个线性函数 $f(\vec{v})$（$\vec{v}$ 表示 lhx 的逆天向量），它的返回值是一个实数，表示 lhx 的逆天值。

例如令 $f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = x_1 + 3x_2 + 10x_3$，则 f 就是一个线性函数。

线性函数的检验非常简单，我们只要检验是否有 (1) 式和 (2) 式成立即可。

$$f(\vec{u}) + f(\vec{v}) = f(\vec{u} + \vec{v}) \tag{1}$$

$$cf(\vec{v}) = f(c\vec{v}) \tag{2}$$

线性函数的检验非常简单，我们只要检验是否有 (1) 式和 (2) 式成立即可。

$$f(\vec{u}) + f(\vec{v}) = f(\vec{u} + \vec{v}) \tag{1}$$

$$cf(\vec{v}) = f(c\vec{v}) \tag{2}$$

例如 $f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \frac{x_1}{4} + x_2 + \frac{x_3}{10}$ 和 $f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = 0$ 都是线性函数，而 $f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \max(x_1, x_2, x_3)$ 和 $f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = (x_1 + x_2)^2 - \sqrt{x_3}$ 则都不是线性函数。

我们注意到，上一页的两个线性函数都是形如 $f\left(\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\right)=c_1x_1+c_2x_2+c_3x_3$ 的形式，所以我们猜想，如果 $f(\vec{v})$ 是线性函数，则必有 $f\left(\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\right)=c_1x_1+c_2x_2+c_3x_3$.

我们注意到，上一页的两个线性函数都是形如 $f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = c_1x_1 + c_2x_2 + c_3x_3$ 的形式，所以我们猜想，如果 $f(\vec{v})$ 是线性函数，则必有 $f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = c_1x_1 + c_2x_2 + c_3x_3$.

Problem 2.1.1 (线性函数的形式)

假设函数 f 满足 $f(\vec{u}) + f(\vec{v}) = f(\vec{u} + \vec{v})$ 和 $cf(\vec{v}) = f(c\vec{v})$。

求证：$f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$.

Proof 2.1.1 (线性函数的形式)

设 $c_1 = f\left(\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}\right), c_2 = f\left(\begin{bmatrix}0\\1\\\vdots\\0\end{bmatrix}\right), \cdots, c_n = f\left(\begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix}\right)$

则有

$$f\left(\begin{bmatrix}x_1\\0\\\vdots\\0\end{bmatrix}\right) = x_1 c_1, f\left(\begin{bmatrix}0\\x_2\\\vdots\\0\end{bmatrix}\right) = x_2 c_2, \cdots, f\left(\begin{bmatrix}0\\0\\\vdots\\x_n\end{bmatrix}\right) = x_n c_n,$$

易得 $f\left(\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}\right) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$。

向量表示

事实上，所有支持相加和数乘的东西都可以被抽象成向量。

比如引言中 wzf 买东西的例子，wzf 买的东西就可以看成一个很大的向量，其维数等于商品的种类数，每一维的值即为该商品的数量，原先的加法和数乘即为现在的向量加法和向量数乘。

甚至一个 $n \times m$ 的矩阵也可以看成一个 nm 维向量，只要我们只考虑矩阵的加法和数乘而不考虑矩阵乘法。

上一节中举出的所有线性函数有一个共同点，它们的函数返回值都是实数。

但线性函数的返回值不一定要是实数，线性函数的返回值也可以是一个向量。

上一节中举出的所有线性函数有一个共同点，它们的函数返回值都是实数。

但线性函数的返回值不一定要是实数，线性函数的返回值也可以是一个向量。

我们一般把从向量到向量的线性函数称作线性变换。

Definition 2.2.1 (线性变换)

如果线性函数 f 满足 $f(\vec{v}) = \vec{w}$，即参数和返回值都是向量，则 f 是一个线性变换。

我们来看一些线性变换的例子（还是以 lhx 先生的逆天程度为例）。

我们来看一些线性变换的例子（还是以 lhx 先生的逆天程度为例）。

Example 2.2.1

由于 lhx 实在是过于逆天了，因此他有时候会被 D。

假设 lhx 被大老师 D 了 a 次，被其它人 D 了 b 次（a 和 b 视为常数），则 lhx 第二天的逆天向量会发生以下变化：

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} ax_1 \\ x_2 \\ \frac{x_3}{b} \end{bmatrix}$$

显然 f 是一个线性变换。

不过不只有这样的函数是线性变换，

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ x_1 + 2x_2 + x_3 \\ 0 \end{bmatrix}$$ 也是一个线性变换。

不过不只有这样的函数是线性变换，

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ x_1 + 2x_2 + x_3 \\ 0 \end{bmatrix}$$ 也是一个线性变换。

Problem 2.2.1

求证：把一个三维向量映射为另一个三维向量的线性函数必然具有以下形式

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} c_{11}x_1 + c_{12}x_2 + c_{13}x_3 \\ c_{21}x_1 + c_{22}x_2 + c_{23}x_3 \\ c_{31}x_1 + c_{32}x_2 + c_{33}x_3 \end{bmatrix}$$

Proof 2.2.1

令

$$\begin{bmatrix} c_{11} \\ c_{21} \\ c_{31} \end{bmatrix} = f\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\right), \begin{bmatrix} c_{12} \\ c_{22} \\ c_{32} \end{bmatrix} = f\left(\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\right), \begin{bmatrix} c_{13} \\ c_{23} \\ c_{33} \end{bmatrix} = f\left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right),$$ 则

由线性函数的定义易得结论成立。

定义域和值域的向量维数不同时也有这一结论成立。假设 f 是从 n 维向量到 m 维向量的线性变换，即

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} c_{11}x_1 + c_{12}x_2 + \cdots + c_{1n}x_n \\ c_{21}x_1 + c_{22}x_2 + \cdots + c_{2n}x_n \\ \vdots \\ \vdots \\ c_{m1}x_1 + c_{m2}x_2 + \cdots + c_{mn}x_n \end{bmatrix}$$

定义域和值域的向量维数不同时也有这一结论成立。假设 f 是从 n 维向量到 m 维向量的线性变换，即

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} c_{11}x_1 + c_{12}x_2 + \cdots + c_{1n}x_n \\ c_{21}x_1 + c_{22}x_2 + \cdots + c_{2n}x_n \\ \vdots \\ \vdots \\ c_{m1}x_1 + c_{m2}x_2 + \cdots + c_{mn}x_n \end{bmatrix}$$

这也导出了我们描述线性变换的一般方法——矩阵。

矩阵

一个 $n \times m$ 的矩阵是一个 $n \times m$ 的表格，表格的每个元素都是一个数。

两个同样大小的矩阵相加定义为对应位相加，一个矩阵数乘定义为其每个元素数乘。

一个 $n \times m$ 的矩阵可以和一个长度为 m 的向量相乘，定义为：

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} c_{11}x_1 + c_{12}x_2 + \cdots + c_{1m}x_m \\ c_{21}x_1 + c_{22}x_2 + \cdots + c_{2m}x_m \\ \vdots \\ c_{n1}x_1 + c_{n2}x_2 + \cdots + c_{nm}x_m \end{bmatrix}$$

。

矩阵乘法

矩阵 A 和向量 $\vec{v}$ 的积可以直接写成 $A\vec{v}$，再在左侧乘上矩阵 B 则得到 $BA\vec{v}$。

显然 $f(\vec{v}) = BA\vec{v}$ 是一个线性变换，因此 $f(\vec{v})$ 会等于某个矩阵 C 乘上 $\vec{v}$，也就自然的定义出了矩阵乘法 $C = BA$。

矩阵乘法

矩阵 A 和向量 $\vec{v}$ 的积可以直接写成 $A\vec{v}$，再在左侧乘上矩阵 B 则得到 $BA\vec{v}$。

显然 $f(\vec{v}) = BA\vec{v}$ 是一个线性变换，因此 $f(\vec{v})$ 会等于某个矩阵 C 乘上 $\vec{v}$，也就自然的定义出了矩阵乘法 $C = BA$。

由于矩阵乘法的定义式太长了，并且大家都会矩阵乘法，所以这里就不放式子了。

矩阵乘法

矩阵 A 和向量 $\vec{v}$ 的积可以直接写成 $A\vec{v}$，再在左侧乘上矩阵 B 则得到 $BA\vec{v}$。

显然 $f(\vec{v}) = BA\vec{v}$ 是一个线性变换，因此 $f(\vec{v})$ 会等于某个矩阵 C 乘上 $\vec{v}$，也就自然的定义出了矩阵乘法 $C = BA$。

由于矩阵乘法的定义式太长了，并且大家都会矩阵乘法，所以这里就不放式子了。

考虑函数 $f(\vec{v}) = AB\vec{v}$，由于它是个线性函数，我们可以进行之前的线性拆分操作，即分别考虑

$$f\left(\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}\right), f\left(\begin{bmatrix}0\\1\\\vdots\\0\end{bmatrix}\right), \cdots, f\left(\begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix}\right)$$

考虑函数 $f(\vec{v}) = AB\vec{v}$，由于它是个线性函数，我们可以进行之前的线性拆分操作，即分别考虑

$$f\left(\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}\right), f\left(\begin{bmatrix}0\\1\\\vdots\\0\end{bmatrix}\right), \cdots, f\left(\begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix}\right)$$

我们会发现，把 $\vec{v}$ 拆成若干个只有一位是 1 其余全是 0 的向量再与 AB 相乘，对应把 B 拆成若干列再与 A 相乘。因此，矩阵乘法也可以看成先把右侧的矩阵拆成若干列向量，然后与左边矩阵相乘后再拼起来。

我们回顾一下刚才的内容，会发现我其实没怎么介绍矩阵乘法，甚至矩阵乘法的定义都是用向量与矩阵的乘法来自然导入的。

我们回顾一下刚才的内容，会发现我其实没怎么介绍矩阵乘法，甚至矩阵乘法的定义都是用向量与矩阵的乘法来自然导入的。

但事实上，这正是我想表达的内容。在 OI 中，矩阵很多时候都是作为线性变换出现的，而矩阵的乘法便是线性变换复合的自然定义。

我们回顾一下刚才的内容，会发现我其实没怎么介绍矩阵乘法，甚至矩阵乘法的定义都是用向量与矩阵的乘法来自然导入的。

但事实上，这正是我想表达的内容。在 OI 中，矩阵很多时候都是作为线性变换出现的，而矩阵的乘法便是线性变换复合的自然定义。

因此，我们很多时候关心的都是如何构造向量和矩阵，使得向量与矩阵的乘法能达到我们想要的结果。在构造完向量和矩阵的乘法之后，剩下的矩阵乘法就只是自然定义的事情。

其它定义

n 和 m 相同的矩阵称为方阵。只有主对角线是 1 其它位置都是 0 的 n 阶方阵称为 n 阶单位矩阵，记作 I，任何矩阵乘上单位矩阵均不变。如果两个方阵 A 和 B 满足 $AB = I$，那么我们称 A 和 B 互为逆元，也可记作 $B = A^{-1}$ 或 $A = B^{-1}$。

将矩阵行列交换后得到的矩阵称为它的转置，矩阵 A 的转置记作 A^T。对于两个矩阵 A 和 B，有
$(AB)^T = B^T A^T, (AB)^{-1} = B^{-1} A^{-1}$。

当题目中出现变换或者转移的时候，我们就可以尝试用向量和矩阵来描述这个变换或者转移。

当题目中出现变换或者转移的时候，我们就可以尝试用向量和矩阵来描述这个变换或者转移。

我们以求斐波那契数列的第 n 项为例：

首先，先找出转移所需的全部内容，列在一个向量里。例如本题的转移方程是 $f_n = f_{n-2} + f_{n-1}$，则转移所需的全部内容就是 f_{n-2} 和 f_{n-1}。我们把它们列在一个向量里，得到 $\begin{bmatrix} f_{n-2} \\ f_{n-1} \end{bmatrix}$。

当题目中出现变换或者转移的时候，我们就可以尝试用向量和矩阵来描述这个变换或者转移。

我们以求斐波那契数列的第 n 项为例：

首先，先找出转移所需的全部内容，列在一个向量里。例如本题的转移方程是 $f_n = f_{n-2} + f_{n-1}$，则转移所需的全部内容就是 f_{n-2} 和 f_{n-1}。我们把它们列在一个向量里，得到 $\begin{bmatrix} f_{n-2} \\ f_{n-1} \end{bmatrix}$。

然后考虑描述转移，我们希望从 $\begin{bmatrix} f_{n-2} \\ f_{n-1} \end{bmatrix}$ 转移到 $\begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix}$。由于向量里包含了转移所需的全部内容，因此我们可以直接构造矩阵。即有 $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} f_{n-2} \\ f_{n-1} \end{bmatrix} = \begin{bmatrix} f_{n-1} \\ f_{n-2} + f_{n-1} \end{bmatrix} = \begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix}$。

当题目中出现变换或者转移的时候，我们就可以尝试用向量和矩阵来描述这个变换或者转移。

我们以求斐波那契数列的第 n 项为例:

首先，先找出转移所需的全部内容，列在一个向量里。例如本题的转移方程是 $f_n = f_{n-2} + f_{n-1}$，则转移所需的全部内容就是 f_{n-2} 和 f_{n-1}。我们把它们列在一个向量里，得到 $\begin{bmatrix} f_{n-2} \\ f_{n-1} \end{bmatrix}$。

然后考虑描述转移，我们希望从 $\begin{bmatrix} f_{n-2} \\ f_{n-1} \end{bmatrix}$ 转移到 $\begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix}$。由于向量里包含了转移所需的全部内容，因此我们可以直接构造矩阵。即有 $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} f_{n-2} \\ f_{n-1} \end{bmatrix} = \begin{bmatrix} f_{n-1} \\ f_{n-2} + f_{n-1} \end{bmatrix} = \begin{bmatrix} f_{n-1} \\ f_n \end{bmatrix}$。

最后矩阵快速幂加速转移，这道题就做完了，是不是非常简单。

这里提一句，当我们需要多次查询一个矩阵的幂的时候，如果我们每次都用快速幂，则时间复杂度是 $O(qn^3 \log v)$ 的。

我们考虑预处理这个矩阵的 2^k 次幂，这样每次查询的时候直接将 $\log$ 个矩阵相乘，复杂度还是 $O(qn^3 \log v)$ 的。

但假如每次查询的是这个矩阵的幂与一个向量的乘法，则 $\log$ 个矩阵依次与向量相乘的复杂度会变为 $n^2 \log v$，总复杂度降至 $O(n^3 \log v + qn^2 \log v)$。

这个套路应该是很广为人知的，不过为了水时长，所以这里还是提一句。

[NOIP2022] 比赛

由于今天的主题是线性代数，因此我们可以忽略此题的一些数据结构部分。

给定三个长度为 n 的序列 a, b, sum，初始全为 0，要求支持区间 $a_i{+}{=}c$，区间 $b_i{+}{=}c$，区间 $sum_i{+}{=}a_i \times b_i$，查询 sum 区间和。

对 2^{32} 取模。

我们考虑用矩阵来表示对某一个下标进行的操作，这样区间操作就是区间乘上同一个矩阵，易于维护。

我们考虑用矩阵来表示对某一个下标进行的操作，这样区间操作就是区间乘上同一个矩阵，易于维护。

首先把所有转移需要的部分列在向量上，比如本题中的 a_i, b_i, sum_i，但似乎这还不够，得带上 a_ib_i 和 1 才能支持转移，因此我们把向量记为 $\begin{bmatrix} 1 \\ a_i \\ b_i \\ a_ib_i \\ sum_i \end{bmatrix}$。

我们考虑用矩阵来表示对某一个下标进行的操作，这样区间操作就是区间乘上同一个矩阵，易于维护。

首先把所有转移需要的部分列在向量上，比如本题中的 a_i, b_i, sum_i，但似乎这还不够，得带上 $a_i b_i$ 和 1 才能支持转移，因此我们把向量记为 $\begin{bmatrix} 1 \\ a_i \\ b_i \\ a_i b_i \\ sum_i \end{bmatrix}$。

然后把每一种操作都写成矩阵的形式，区间加 a 是

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ c & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & c & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ a_i \\ b_i \\ a_i b_i \\ sum_i \end{bmatrix} = \begin{bmatrix} 1 \\ a_i + c \\ b_i \\ a_i b_i + c b_i \\ sum_i \end{bmatrix},$$

区间加 b 是 $$\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\c&0&1&0&0\\0&c&0&1&0\\0&0&0&0&1\end{bmatrix}\begin{bmatrix}1\\a_i\\b_i\\a_ib_i\\sum_i\end{bmatrix}=\begin{bmatrix}1\\a_i\\b_i+c\\a_ib_i+ca_i\\sum_i\end{bmatrix},$$

区间 $sum{+}{=}a\times b$ 是
$$\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&1&1\end{bmatrix}\begin{bmatrix}1\\a_i\\b_i\\a_ib_i\\sum_i\end{bmatrix}=\begin{bmatrix}1\\a_i\\b_i\\a_ib_i\\sum_i+a_ib_i\end{bmatrix}。$$

现在能维护了，但常数似乎很大，用线段树维护五阶矩阵会达到 $125n\log n$，这是我们无法接受的，我们考虑优化常数。

现在能维护了，但常数似乎很大，用线段树维护五阶矩阵会达到 $125n\log n$，这是我们无法接受的，我们考虑优化常数。

注意到该矩阵是一个下三角矩阵，那么我们在写矩阵乘法的时候可以只枚举满足 $i \le j \le k$ 的下标，因为其它情况都会是 0，这大约可以把常数除 6。在矩阵阶数为 5 的情况下会把 125 变成 35。

但这样还不够，我们考虑继续减小常数。

但这样还不够，我们考虑继续减小常数。

注意到矩阵中会有很多无效项，例如对角线上的值永远是 1，a_i 对 b_i 的贡献永远是 0。因此在计算的时候我们可以跳过这些项，矩阵也可以从五阶变成四阶。

[NOI2021] 密码箱

定义连分数函数

$$f(a_0, a_1, \ldots, a_{k-1}, a_k) = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\cdots + \cfrac{1}{a_{k-1} + \cfrac{1}{a_k}}}}。$$

你现在有一个数列，数列初始有两项 $a_0 = 0, a_1 = 1$，你还有两种类型的操作：

W 类型：给数列的最后一项加 1。

E 类型：若数列的最后一项为 1，则给倒数第二项加 1；否则先给数列的最后一项减 1，接着在数列的结尾添加两项 1。

[NOI2021] 密码箱

你还有一个操作序列，你会对这个操作序列进行三种修改：

APPEND c 表示在现有操作序列后追加一次 c 类型操作，其中 c 为字符 W 或 E。

FLIP l r 反转现有操作序列中第 l 个至第 r 个（下标从 1 开始，修改包含端点 l 和 r，下同）操作，即所有 W 变为 E，所有 E 变为 W。

REVERSE l r 翻转现有操作序列中第 l 个至第 r 个操作，也就是将这个区间中的操作逆序。

你需要在每次修改完成后，求出依次执行操作序列得到的数列的连分数值（分子分母分别取模）。

按照套路，我们需要维护一个向量，使得这个向量能够用矩阵转移，并且能用向量得到答案。但是，由于连分数实在是太奇怪了，所以我们暂时还无法判断维护什么内容。

按照套路，我们需要维护一个向量，使得这个向量能够用矩阵转移，并且能用向量得到答案。但是，由于连分数实在是太奇怪了，所以我们暂时还无法判断维护什么内容。

我们仔细观察连分数的形式，会发现连分数似乎支持单端修改，在数列最前面插入元素或删去数列第一项都能比较方便的维护。假如当前连分数的值是 $\frac{p}{q}$，在最前面添加一项 c 后就会变成 $\frac{q+cp}{p}$。

按照套路，我们需要维护一个向量，使得这个向量能够用矩阵转移，并且能用向量得到答案。但是，由于连分数实在是太奇怪了，所以我们暂时还无法判断维护什么内容。

我们仔细观察连分数的形式，会发现连分数似乎支持单端修改，在数列最前面插入元素或删去数列第一项都能比较方便的维护。假如当前连分数的值是 $\frac{p}{q}$，在最前面添加一项 c 后就会变成 $\frac{q+cp}{p}$。

注意到从 (p,q) 到 $(q+cp,p)$ 是一个线性变换的过程，这启发我们用向量和矩阵来描述这一过程，即

$$\begin{bmatrix} c & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} q+cp \\ p \end{bmatrix}。$$

当我们把数列的每一项都变成一个矩阵之后，原先的数列就变成了一个矩阵序列，我们关心的内容就是这个序列的积。

当我们把数列的每一项都变成一个矩阵之后，原先的数列就变成了一个矩阵序列，我们关心的内容就是这个序列的积。

题目中需要我们对连分数的末端进行操作，这很困难。但在我们转化为矩阵之后，就变成了对矩阵序列的一段进行操作，这是很简单的。

这就是矩阵的另一大优势，**矩阵能让原本只支持单向修改的东西支持双向修改甚至随机修改**。

来看题面中的两个操作。先是 W 操作，给序列的最后一项加 1。

来看题面中的两个操作。先是 W 操作，给序列的最后一项加 1。

这是相对简单的，因为它没有分类讨论。我们考虑观察进行一次 W 操作之后的结果，设最后一项为原本 a，则得到 $(a+1)+\cfrac{1}{\frac{p}{q}}=a+\cfrac{1}{\frac{p}{p+q}}$，也就是在矩阵序列的右端添加一个矩阵 $\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$。

再来看 E 操作，E 操作看起来有点复杂，它甚至带有分类讨论。但仔细观察，会发现"若数列的最后一项为 1，则给倒数第二项加 1"这半句可以删去，它和后面一种情况是等同的。

当倒数第二项为 a，最后一项为 1 时，我们有：

$$a+\cfrac{1}{(1-1)+\cfrac{1}{1+\cfrac{1}{1}}}=a+2=(a+1)+\frac{1}{1}$$

现在分类讨论统一了，我们就可以用与 W 操作相同的方法分析 E 操作对应的矩阵。

现在分类讨论统一了，我们就可以用与 W 操作相同的方法分析 E 操作对应的矩阵。

$$(x-1)+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{\left(\cfrac{p}{q}\right)}}}=x+\frac{-p}{2p+q}=x+\cfrac{1}{\left(\cfrac{2p+q}{-p}\right)}$$

因此 E 操作相当于在矩阵序列右端添加矩阵 $\begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}$

现在，W 操作和 E 操作都变成了矩阵，那么直接用平衡树维护矩阵序列即可。

矩阵的形式来源于线性变换，但矩阵的形式不一定要用于刻画线性变换，也可以更改运算来刻画其它东西。

矩阵的形式来源于线性变换，但矩阵的形式不一定要用于刻画线性变换，也可以更改运算来刻画其它东西。

最为常见的一种是把加法改为 $\max$，把乘法改为加法，这样仍然能保证分配律，也能保证矩阵的许多基本性质。

最大子段和

给定一个序列，要求支持单点修改，查询区间最大子段和。

$n, q \leq 10^5$

显然可以用线段树维护一个手写的结构体来实现。

显然可以用线段树维护一个手写的结构体来实现。

但假如我懒得思考如何写一个支持合并的结构体，我也可以选择无脑用矩阵。在前一道题里也说了，矩阵能用于维护只支持单向修改的东西。我们用一个向量存储需要的内容：当前答案和当前最大后缀和。当我们在序列末尾添加了一个数 c 之后，则会发生以下转移：

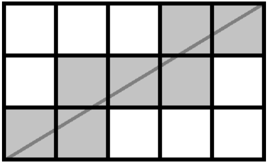
$$\begin{bmatrix} 0 & c & -\infty \\ -\infty & c & c \\ -\infty & -\infty & 0 \end{bmatrix} \begin{bmatrix} ans \\ suf \\ 0 \end{bmatrix} = \begin{bmatrix} \max(ans, suf + c) \\ \max(suf + c, c) \\ 0 \end{bmatrix}$$

使用线段树维护 $\max, +$ 矩阵即可。

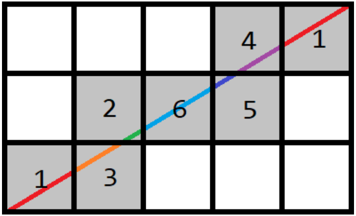
你以为只有 $\max, +$ 矩阵？

给定一个方格图，它由 n 行 m 列边长相同的正方形格子构成，其中 n 和 m 互质，且方格表的对角线长度为 nm。

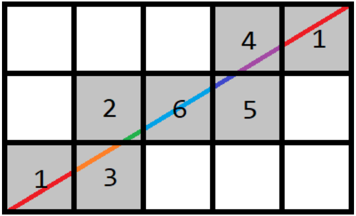
你面前有一个骰子，这个骰子的每个面都有一个颜色，且互不相同。现在这个骰子在左下角，它会沿着对角线与方格相交的轮廓一直滚到右上角，如图所示。



之后，对角线的每一段都会被骰子染上接触面所对应的颜色。



你需要对每种颜色求出该颜色对应的所有段长度的 k 次方和。$0 \le k \le 2$，$1 \le n, m \le 10^{18}$，$\gcd(n, m) = 1$，$1 \le T \le 10^5$



你需要对每种颜色求出该颜色对应的所有段长度的 k 次方和。$0 \le k \le 2$，$1 \le n, m \le 10^{18}$，$\gcd(n, m) = 1$，$1 \le T \le 10^5$

此处给出算法概要：构造出包含所有有效信息的向量，和两个转移矩阵，分别用于描述跨过横线和跨过竖线时的转移。把跨过横线的矩阵记为 U，跨过竖线的矩阵记为 R，则骰子滚动的过程即为一串 U 和 R 的乘积，如上图的过程为 $RURRUR$。这一 UR 序列的乘积可以用欧几里得算法快速计算。

最困难的部分在于如何维护每条线段的长度。

最困难的部分在于如何维护每条线段的长度。

首先先通过交换 n, m，来确保 $n < m$，即宽比高长。我们考虑定义一个辅助变量 t，表示当前格右轮廓线与对角线的交点位置到上一条横线的距离。那么每当翻过一条竖线的时候，t 就会增加 n；翻过横线的时候，t 就会减少 m。

最困难的部分在于如何维护每条线段的长度。

首先先通过交换 n, m，来确保 $n < m$，即宽比高长。我们考虑定义一个辅助变量 t，表示当前格右轮廓线与对角线的交点位置到上一条横线的距离。那么每当翻过一条竖线的时候，t 就会增加 n；翻过横线的时候，t 就会减少 m。

每当翻过一条竖线的时候，我们就假装新线段的长度是 m，线段长度不是 m 的情况在翻横线时进行修正。翻过横线的时候，新线段的长度显然是 t，而上一条线段的长度不是假定的 m 而是 $m - t$，这时候就对其进行修正。

于是我们考虑构造我们的向量，$\begin{bmatrix}(g,t)\\ ans\end{bmatrix}$。向量一共两维，第一维是骰子状态 g 和辅助变量 t 构成的二元组，第二维是六种颜色的答案构成的数组。

于是我们考虑构造我们的向量，$\begin{bmatrix}(g,t)\\ ans\end{bmatrix}$。向量一共两维，第一维是骰子状态 g 和辅助变量 t 构成的二元组，第二维是六种颜色的答案构成的数组。

矩阵 $\begin{bmatrix}(r,c) & \\ S & 1\end{bmatrix}$，表示经过这次翻转后，骰子状态 g 会复合上骰子状态 r（g 和 r 都是骰子状态的编号，复合就是置换复合），S 会在代入 (g,t) 之后加入答案 ans。

其中 S 有 6×3 个元素，表示要把当前骰子的第 i 面的颜色对应的答案加上 $a_2t^2+a_1t+a_0$。

矩阵乘法仍然通过自然定义得到，这里略过。

线性表示

如果存在系数 $c_1, c_2, \cdots, c_n$ 满足 $\vec{v} = c_1\vec{v_1} + c_2\vec{v_2} + \cdots + c_n\vec{v_n}$，则称 $\vec{v}$ 是 $\vec{v_1}, \vec{v_2}, \cdots, \vec{v_n}$ 的一个线性组合。

线性表示

如果存在系数 $c_1, c_2, \cdots, c_n$ 满足 $\vec{v} = c_1\vec{v_1} + c_2\vec{v_2} + \cdots + c_n\vec{v_n}$，则称 $\vec{v}$ 是 $\vec{v_1}, \vec{v_2}, \cdots, \vec{v_n}$ 的一个线性组合。

可以看到线性组合集加法和数乘于一身，因此它与“线性”这一概念有着密切的联系。我们可以用线性组合来定义线性空间：

一个关于线性组合封闭的集合即为线性空间（线性空间也简称空间），由全体 n 维向量构成的空间叫做 n 维列向量空间。

线性无关

对于一组向量 $\vec{v_i}$，如果它们中的任何一个向量都不能表示成其它向量的线性组合，那么称这组向量是线性无关的，否则这组向量是线性相关的。

线性无关

对于一组向量 $\vec{v_i}$，如果它们中的任何一个向量都不能表示成其它向量的线性组合，那么称这组向量是线性无关的，否则这组向量是线性相关的。

一组向量线性无关，当且仅当零向量只有一个作为线性组合的表达式，即系数全为 0。

基

对于一个向量集合 $S=\{\vec{v_1},\vec{v_2},\cdots,\vec{v_n}\}$，它其中的所有向量的线性组合构成一个线性空间，称为集合 S 张成的空间，记作 $\mathrm{Span}\,S$。这个空间也是包含集合 S 的最小的线性空间。

基

对于一个向量集合 $S=\{\vec{v_1},\vec{v_2},\cdots,\vec{v_n}\}$，它其中的所有向量的线性组合构成一个线性空间，称为集合 S 张成的空间，记作 $\operatorname{Span} S$。这个空间也是包含集合 S 的最小的线性空间。

对于一个线性空间 V，如果一组向量线性无关，且张成 V，则称这组向量是空间 V 的一个基。

基

对于一个向量集合 $S=\{\vec{v_1},\vec{v_2},\cdots,\vec{v_n}\}$，它其中的所有向量的线性组合构成一个线性空间，称为集合 S 张成的空间，记作 $\operatorname{Span} S$。这个空间也是包含集合 S 的最小的线性空间。

对于一个线性空间 V，如果一组向量线性无关，且张成 V，则称这组向量是空间 V 的一个基。

易证，集合 $\boldsymbol{B}=\{\vec{v_1},\vec{v_2},\cdots,\vec{v_n}\}$ 是空间 V 的基当且仅当 V 中的每个向量都能以唯一形式写成 B 的线性组合。基 $\boldsymbol{B}$ 具有矩阵形式，即从左到右每一列分别为 $\vec{v_1},\vec{v_2},\cdots,\vec{v_n}$ 的 $m\times n$ 矩阵（m 是向量维数）。

维数

一个线性空间的基集合的大小总是一个固定的值（通过线性方程组证明，此处略去），我们称这个值为线性空间的维数。线性空间 V 的维数记作 $\dim V$。

维数

一个线性空间的基集合的大小总是一个固定的值（通过线性方程组证明，此处略去），我们称这个值为线性空间的维数。线性空间 V 的维数记作 $\dim V$。

由 n 维向量张成的空间维数至多为 n（取到 n 时即为 n 维列向量空间），只包含零向量的空间维数为 0。

线性变换

一个线性变换会把线性空间映射成另一个线性空间，因为变换前后向量的线性组合关系不变。

新的线性空间维数不可能大于原先的线性空间，因为原先的基在映射后仍然能张成整个空间。

线性变换

一个线性变换会把线性空间映射成另一个线性空间，因为变换前后向量的线性组合关系不变。

新的线性空间维数不可能大于原先的线性空间，因为原先的基在映射后仍然能张成整个空间。

对于任意 m 维的线性空间，都一定存在一个线性变换，能把它映射到所有 m 维向量构成的线性空间。

秩

秩是矩阵的一个属性，矩阵 A 的秩记作 $\mathrm{Rank}(A)$，用于描述线性变换的“维数”。

秩

秩是矩阵的一个属性，矩阵 A 的秩记作 $\mathrm{Rank}(A)$，用于描述线性变换的“维数”。

定义一个 $n \times m$ 矩阵的秩为线性无关的列的数量，即所有列张成空间的维数，也就是映射 m 维列向量空间得到的新空间的维数。

秩

秩是矩阵的一个属性，矩阵 A 的秩记作 $\mathrm{Rank}(A)$，用于描述线性变换的“维数”。

定义一个 $n \times m$ 矩阵的秩为线性无关的列的数量，即所有列张成空间的维数，也就是映射 m 维列向量空间得到的新空间的维数。

对该矩阵进行任意可逆的线性变换不会改变它的秩，因为秩等于列向量空间映射后的维数，而可逆操作必然不会降低空间的维数。

秩

秩是矩阵的一个属性，矩阵 A 的秩记作 $\mathrm{Rank}(A)$，用于描述线性变换的“维数”。

定义一个 $n \times m$ 矩阵的秩为线性无关的列的数量，即所有列张成空间的维数，也就是映射 m 维列向量空间得到的新空间的维数。

对该矩阵进行任意可逆的线性变换不会改变它的秩，因为秩等于列向量空间映射后的维数，而可逆操作必然不会降低空间的维数。

因此，我们可以通过对矩阵进行可逆行变换与列变换的方式进行消元，借助消元即可得出重要结论：**矩阵线性无关的行的数量等于线性无关的列的数量**，即矩阵的秩等于其转置的秩。

3b1b

通过 3b1b 的视频来形象地理解线性代数概念!
https://www.bilibili.com/video/BV1ys411472E?p=4

如果你想以较为形象的方式入门线性代数，那么这里强烈推荐观看 3b1b 的视频，胜过绝大部分的书和课程。初次看可以 0.75 倍速，更加便于理解。

降秩

如果一个 $n \times m$ 的矩阵 A，满足 $\mathrm{Rank}(A) = \min(n, m)$，则称 A 是一个满秩矩阵，否则称 A 是一个降秩矩阵。

降秩

如果一个 $n \times m$ 的矩阵 A，满足 $\mathrm{Rank}(A) = \min(n, m)$，则称 A 是一个满秩矩阵，否则称 A 是一个降秩矩阵。

如果一个线性变换的矩阵表示是降秩的，则它映射时可能会使得空间维数减小（秩等于 m 的矩阵必然保持维数不变）。对此，我们有维数公式：

降秩

如果一个 $n \times m$ 的矩阵 A，满足 $\mathrm{Rank}(A) = \min(n, m)$，则称 A 是一个满秩矩阵，否则称 A 是一个降秩矩阵。

如果一个线性变换的矩阵表示是降秩的，则它映射时可能会使得空间维数减小（秩等于 m 的矩阵必然保持维数不变）。对此，我们有维数公式：

Theorem 3.1.1 (维数公式)

设 T 是从空间 V 到空间 W 的一个线性变换，记 $\ker T = \{\vec{v} \in V \mid T(\vec{v}) = 0\}$ 表示 T 的核（也称零空间），$\mathrm{im}\ T = \{\vec{w} \in W \mid \exists \vec{v} \in V, \vec{w} = T(\vec{v})\}$ 表示 T 的像，则有 $\dim(\ker T) + \dim(\mathrm{im}\ T) = \dim V$。

基变换

给定同一个线性空间 V 的两个基，比如 $\boldsymbol{B}=\{\vec{v_1},\vec{v_2},\cdots,\vec{v_n}\}$ 和 $\boldsymbol{B'}=\{\vec{v_1}',\vec{v_2}',\cdots,\vec{v_n}'\}$。考虑一个线性变换 P，它会把基 $\boldsymbol{B}$ 映射到基 $\boldsymbol{B'}$，即把 $c_1\vec{v_1}+c_2\vec{v_2}+\cdots+c_n\vec{v_n}$ 映射为 $c_1\vec{v_1}'+c_2\vec{v_2}'+\cdots+c_n\vec{v_n}'$，写成矩阵形式即为 $\boldsymbol{B'}=\boldsymbol{B}P$。

基变换

给定同一个线性空间 V 的两个基，比如 $\boldsymbol{B}=\{\vec{v_1},\vec{v_2},\cdots,\vec{v_n}\}$ 和 $\boldsymbol{B}'=\{\vec{v_1}',\vec{v_2}',\cdots,\vec{v_n}'\}$。考虑一个线性变换 P，它会把基 $\boldsymbol{B}$ 映射到基 $\boldsymbol{B}'$，即把 $c_1\vec{v_1}+c_2\vec{v_2}+\cdots+c_n\vec{v_n}$ 映射为 $c_1\vec{v_1}'+c_2\vec{v_2}'+\cdots+c_n\vec{v_n}'$，写成矩阵形式即为 $\boldsymbol{B}'=\boldsymbol{B}P$。

对于给定的两个基 $\boldsymbol{B}$ 和 $\boldsymbol{B}'$，矩阵 P 是唯一的，即 $\boldsymbol{B}'$ 在 $\boldsymbol{B}$ 上的线性表示，我们称矩阵 P 为基变换矩阵。原先的向量 $\boldsymbol{B}X$ 会在基变换之后变为 $\boldsymbol{B}'X=\boldsymbol{B}PX$，即在原基上的坐标从 X 变为 PX。

线性算子

线性算子指从一个线性空间到其自身的线性变换，因此，线性算子的矩阵一定是一个方阵。例如上文中的基变换 P 会把向量的坐标从 X 变为 PX，因此可以视为线性算子。

线性算子

线性算子指从一个线性空间到其自身的线性变换，因此，线性算子的矩阵一定是一个方阵。例如上文中的基变换 P 会把向量的坐标从 X 变为 PX，因此可以视为线性算子。

考虑进行一次基变换后线性算子会发生什么变化。设 T 是一个线性算子，对于原本的基 $\boldsymbol{B}$，我们有 T 关于 $\boldsymbol{B}$ 的矩阵为 A，即 $T\boldsymbol{B}X = \boldsymbol{B}Y \Leftrightarrow AX = Y$。而当基从 $\boldsymbol{B}$ 变为 $\boldsymbol{B}' = \boldsymbol{B}P$ 后，线性算子 T 关于 $\boldsymbol{B}'$ 的矩阵就会变成 A'，满足 $T\boldsymbol{B}'X = \boldsymbol{B}'Y \Leftrightarrow A'X = Y$。

线性算子

线性算子指从一个线性空间到其自身的线性变换，因此，线性算子的矩阵一定是一个方阵。例如上文中的基变换 P 会把向量的坐标从 X 变为 PX，因此可以视为线性算子。

考虑进行一次基变换后线性算子会发生什么变化。设 T 是一个线性算子，对于原本的基 $\boldsymbol{B}$，我们有 T 关于 $\boldsymbol{B}$ 的矩阵为 A，即 $T\boldsymbol{B}X = \boldsymbol{B}Y \Leftrightarrow AX = Y$。而当基从 $\boldsymbol{B}$ 变为 $\boldsymbol{B}' = \boldsymbol{B}P$ 后，线性算子 T 关于 $\boldsymbol{B}'$ 的矩阵就会变成 A'，满足 $T\boldsymbol{B}'X = \boldsymbol{B}'Y \Leftrightarrow A'X = Y$。

对此，我们有 $A'X = Y \Leftrightarrow T\boldsymbol{B}'X = \boldsymbol{B}'Y \Leftrightarrow T\boldsymbol{B}PX = \boldsymbol{B}PY \Leftrightarrow APX = PY$，因此有 $A' = P^{-1}AP$。

相似矩阵

如果矩阵 A, A' 和可逆矩阵 P 满足 $P^{-1}AP = A'$，则称矩阵 A 和 A' 是相似的（又称共轭）。由上一页可知，相似矩阵是同一个线性变换在不同基下的矩阵表示。因此，对于那些不涉及基的性质，相似矩阵是完全相同的，比如相似矩阵的行列式与特征多项式均相等。

相似矩阵

如果矩阵 A, A' 和可逆矩阵 P 满足 $P^{-1}AP = A'$，则称矩阵 A 和 A' 是相似的（又称共轭）。由上一页可知，相似矩阵是同一个线性变换在不同基下的矩阵表示。因此，对于那些不涉及基的性质，相似矩阵是完全相同的，比如相似矩阵的行列式与特征多项式均相等。

相似还是一种等价关系，即具有传递性，因此我们可以通过相似关系把矩阵分成若干个共轭类，每一类对应一种线性变换。相似关系 $f(A) = P^{-1}AP$ 也具有良好的性质，它关于加法和乘法都是同态，即 $f(A+B) = f(A) + f(B), f(AB) = f(A)f(B)$。

1 引入

2 向量表示与矩阵

3 线性空间
- 理论内容
- 高斯消元
- 来点题

4 行列式与特征多项式

5 Thanks

高斯消元的过程大家都会，所以这里不讲了。

高斯消元的过程大家都会，所以这里不讲了。

高斯消元的过程相当于对矩阵进行若干行变换，而行变换就相当于在矩阵左侧乘上另一个矩阵。我们可以把矩阵的每一行缩起来，写成 $\begin{bmatrix} \vec{v_1}^T \\ \vec{v_2}^T \\ \vdots \\ \vec{v_n}^T \end{bmatrix}$ 的类向量形式，左乘矩阵就会得到 $\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \vec{v_1}^T \\ \vec{v_2}^T \end{bmatrix} = \begin{bmatrix} c_{11}\vec{v_1}^T + c_{12}\vec{v_2}^T \\ c_{21}\vec{v_1}^T + c_{22}\vec{v_2}^T \end{bmatrix}$，即对矩阵 A 进行的任意行变换都可以看作左乘一个方阵 E，得到 EA。

高斯消元的过程大家都会，所以这里不讲了。

高斯消元的过程相当于对矩阵进行若干行变换，而行变换就相当于在矩阵左侧乘上另一个矩阵。我们可以把矩阵的每一行缩起来，写成 $\begin{bmatrix} \vec{v_1}^T \\ \vec{v_2}^T \\ \vdots \\ \vec{v_n}^T \end{bmatrix}$ 的类向量形式，左乘矩阵就会得到 $\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \vec{v_1}^T \\ \vec{v_2}^T \end{bmatrix} = \begin{bmatrix} c_{11}\vec{v_1}^T + c_{12}\vec{v_2}^T \\ c_{21}\vec{v_1}^T + c_{22}\vec{v_2}^T \end{bmatrix}$，即对矩阵 A 进行的任意行变换都可以看作左乘一个方阵 E，得到 EA。

例如 $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a & b \\ a+c & b+d \end{bmatrix}$ 就描述了把第一行加到第二行上的变换。

线性方程组的矩阵表示是 $AX=B$，当我们用高斯消元法解方程组时，相当于在等式两边左乘矩阵 P，变成 $PAX=PB$，使得 PA 为单位矩阵（或行阶梯形矩阵），然后得到 $X=PB$（或类似结果）。

线性方程组的矩阵表示是 $AX=B$，当我们用高斯消元法解方程组时，相当于在等式两边左乘矩阵 P，变成 $PAX=PB$，使得 PA 为单位矩阵（或行阶梯形矩阵），然后得到 $X=PB$（或类似结果）。

当我们用高斯消元法对矩阵 A 求逆的时候，相当于对它左乘一个矩阵 P 使得 $PA=I$，因此 $P=A^{-1}$，把 P 描述的行变换对单位矩阵进行得到 $PI=P$，即完成了求逆。

线性方程组的矩阵表示是 $AX = B$，当我们用高斯消元法解方程组时，相当于在等式两边左乘矩阵 P，变成 $PAX = PB$，使得 PA 为单位矩阵（或行阶梯形矩阵），然后得到 $X = PB$（或类似结果）。

当我们用高斯消元法对矩阵 A 求逆的时候，相当于对它左乘一个矩阵 P 使得 $PA = I$，因此 $P = A^{-1}$，把 P 描述的行变换对单位矩阵进行得到 $PI = P$，即完成了求逆。

由于在乘上可逆矩阵后矩阵的秩不变，因此高斯消元也可以用来求矩阵的秩。

上海森堡矩阵

上海森堡矩阵是指一个 $n\times n$ 的矩阵 A，满足对于任意 $i>j+1$，都有 $a_{ij}=0$。即只有主对角线右上的这半部分和主对角线左下的一条斜线有值，如
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{bmatrix}$$
。

上海森堡矩阵

上海森堡矩阵是指一个 $n \times n$ 的矩阵 A，满足对于任意 $i > j+1$，都有 $a_{ij} = 0$。即只有主对角线右上的这半部分和主对角线左下的一条斜线有值，如 $\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{bmatrix}$。

对于一些特殊情况，我们不能在高斯消元时左乘任意矩阵 P，必须同时右乘矩阵 P^{-1}，来保证方阵 A 变为它的相似矩阵 PAP^{-1}。这时，我们很难把 A 消成上三角矩阵，但仍然能消成上海森堡矩阵。

动态高斯消元

每次插入一个向量，维护它们张成的线性空间，如支持查询一个向量是否在该线性空间中，查询张成的线性空间的维数。

动态高斯消元

每次插入一个向量，维护它们张成的线性空间，如支持查询一个向量是否在该线性空间中，查询张成的线性空间的维数。

就是线性基而已。

动态高斯消元

每次插入一个向量，维护它们张成的线性空间，如支持查询一个向量是否在该线性空间中，查询张成的线性空间的维数。

就是线性基而已。

如需支持删除可以使用线段树分治。

PKUSC2023 试机题

给定一个 n 阶方阵 A，定义一次操作为把每个 a_{ij} 变为第 i 行所有元素的和加上第 j 列所有元素的和，求 k 次操作后的矩阵。

$n \leq 2000, k \leq 10^9$，对 998244353 取模。

我们想办法用矩阵语言描述题目中的变换。

我们想办法用矩阵语言描述题目中的变换。

令 E 表示一个 n 阶方阵，它的每个位置都是 1。那么把 A 每个元素变成它所在行的元素和之后，矩阵 A 就会变成 AE。列的情况同理，会从 A 变成 EA。因此一次操作会把 A 变成 $AE + EA$。

我们想办法用矩阵语言描述题目中的变换。

令 E 表示一个 n 阶方阵，它的每个位置都是 1。那么把 A 每个元素变成它所在行的元素和之后，矩阵 A 就会变成 AE。列的情况同理，会从 A 变成 EA。因此一次操作会把 A 变成 $AE+EA$。

因为有 $E^2=nE$，所以我们可以列出一个矩阵变换若干次后的结果：$A \Rightarrow AE+EA \Rightarrow nAE+nEA+2EAE \Rightarrow n^2AE+n^2EA+6nEAE \Rightarrow \cdots \Rightarrow n^{k-1}(AE+EA)+(2^k-2)n^{k-2}EAE$

该方法在非方阵的情况下也生效。

[THUPC2023 决赛] Freshman Dream

给定一个 $n \times n$ 的矩阵 A，它的每一位都有 50% 的概率为 0，50% 的概率为 1，且不同位之间相互独立。

你需要构造一个 $n \times n$ 的 01 矩阵 B，满足 B 中恰好有 k 个 1，且对于任意 $1 \le i,j \le n$ 有 $(AB)_{ij} = A_{ij}B_{ij}$。（这里的矩阵乘法在模 2 意义下进行）

$n = 100, 0 \le k \le n^2$

来点题

第二章提到，矩阵乘法中右边的矩阵是可以按列拆分的，因此我们把 B 拆成若干列，一列一列考虑。

第二章提到，矩阵乘法中右边的矩阵是可以按列拆分的，因此我们把 B 拆成若干列，一列一列考虑。

比如现在考虑 B 的第一列，即需要构造 $\vec{b_1} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$，使得 $a_{i1}x_i = (A\vec{b_1})_i$，也就是

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} =$$

$$\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 \\ a_{21}x_2 \\ \vdots \\ a_{n1}x_n \end{bmatrix}。$$

移项一下可得 $\begin{bmatrix} (a_{11}-a_{11})x_1+a_{12}x_2+\cdots+a_{1n}x_n \\ a_{21}x_1+(a_{22}-a_{21})x_2+\cdots+a_{2n}x_n \\ \vdots \\ a_{n1}x_1+a_{n2}x_2+\cdots+(a_{nn}-a_{n1})x_n \end{bmatrix}$，因此

原问题改写为

$$\begin{bmatrix} a_{11}-a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}-a_{21} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}-a_{n1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = 0$$，即构造 $\vec{b_1}$

使得 $A_1\vec{b_1}=0$。

移项一下可得 $\begin{bmatrix} (a_{11}-a_{11})x_1+a_{12}x_2+\cdots+a_{1n}x_n \\ a_{21}x_1+(a_{22}-a_{21})x_2+\cdots+a_{2n}x_n \\ \vdots \\ a_{n1}x_1+a_{n2}x_2+\cdots+(a_{nn}-a_{n1})x_n \end{bmatrix}$，因此

原问题改写为

$$\begin{bmatrix} a_{11}-a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}-a_{21} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}-a_{n1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = 0$$

，即构造 $\vec{b_1}$ 使得 $A_1\vec{b_1}=0$。

这也相当于选取 A_1 的若干行使得它们相加等于 0。由于 A 是随机生成的，所以 A_1 的秩不会太小，我们可以利用线性基枚举出每一个满足 $A_1\vec{b_1}=0$ 的 $\vec{b_1}$，之后对所有列跑一个背包即可得到 1 的总个数恰为 k 的方案。

线性空间计数

假设我们现在的数域是 F_p（即矩阵与向量的每个元素都是 $[0, p-1]$ 的整数，且所有运算在模 p 意义下进行）。

线性空间计数

假设我们现在的数域是 F_p（即矩阵与向量的每个元素都是 $[0, p-1]$ 的整数，且所有运算在模 p 意义下进行）。

Problem 3.3.1 (可逆矩阵计数)

阶为 n 的可逆矩阵有多少个？

线性空间计数

假设我们现在的数域是 F_p（即矩阵与向量的每个元素都是 $[0, p-1]$ 的整数，且所有运算在模 p 意义下进行）。

Problem 3.3.1 (可逆矩阵计数)

阶为 n 的可逆矩阵有多少个？

Solution 3.3.1 (可逆矩阵计数)

第一列不能是零向量，共 p^n-1 种方案；第二列不能是第一列的线性组合，共 p^n-p 种方案；第三列不能是前两列的线性组合，共 p^n-p^2 种方案。

以此类推，总方案数为 $\prod_{i=0}^{n-1}(p^n-p^i)$。

Problem 3.3.2 (满秩矩阵计数)

$n \times m$ 的满秩矩阵有多少个？$(n \geq m)$

Problem 3.3.2 (满秩矩阵计数)

$n \times m$ 的满秩矩阵有多少个？($n \geq m$)

Solution 3.3.2 (满秩矩阵计数)

和上一个问题完全一致，只是列数只有 m 而不是 n，故答案为 $\prod_{i=0}^{m-1}(p^n - p^i)$。

Problem 3.3.3 (线性空间计数)

n 维列向量构成的 m 维线性空间有多少种？$(n \geq m)$

Problem 3.3.3 (线性空间计数)

n 维列向量构成的 m 维线性空间有多少种？($n \geq m$)

Solution 3.3.3 (线性空间计数)

一个线性空间的一个基与一个 $n \times m$ 的满秩矩阵一一对应，一个 m 维线性空间的基的个数等于 m 阶可逆矩阵的个数。

因此线性空间的个数等于 $n \times m$ 的满秩矩阵的个数除以 m 阶可逆矩阵的个数，即答案等于 $\dfrac{\prod_{i=0}^{m-1}(p^n - p^i)}{\prod_{i=0}^{m-1}(p^m - p^i)}$。

Problem 3.3.4 (定秩矩阵计数)

秩为 r 的 $n \times m$ 矩阵有多少种？($n \geq m$)

Problem 3.3.4 (定秩矩阵计数)

秩为 r 的 $n \times m$ 矩阵有多少种？($n \geq m$)

Solution 3.3.4 (定秩矩阵计数)

矩阵的秩为 r，即所有列张成的线性空间维数为 r，那么先枚举这个空间。在确定了空间之后，需要在这个空间中选取 m 个向量，要求这 m 个向量能够张成这个空间，这部分的方案数等于秩为 r 的 $r \times m$ 矩阵的个数。

总方案数就是两者相乘，即 $\dfrac{\prod_{i=0}^{r-1}(p^n - p^i)}{\prod_{i=0}^{r-1}(p^r - p^i)} \cdot \prod_{i=0}^{r-1}(p^m - p^i)$。

[UOJ453/loj6040] 围绕着我们的圆环

给定一个 $p \times r$ 的 01 矩阵 C，共 m 次修改，每次修改 C 的一行，并求有多少 $p \times q$ 的 01 矩阵 A 和 $q \times r$ 的 01 矩阵 B 满足 A 和 B 在模 2 意义下的乘积等于 C。

$p, q, r, m \leq 10^3$

假如 C 的秩为 k，那么我们一定可以通过消元使得 C 只有 $c_{ii}(1 \leq i \leq k)$ 的位置是 1，其它位置都是 0。

假如 C 的秩为 k，那么我们一定可以通过消元使得 C 只有 $c_{ii}(1 \le i \le k)$ 的位置是 1，其它位置都是 0。

而消元只会用到行变换和列变换，因此消元的过程可以看成 $C \Rightarrow PCQ$，其中 P 和 Q 分别是 p 阶和 r 阶的可逆矩阵。因此可以把等式从 $AB = C$ 变成 $(PA)(BQ) = PCQ$，对 A 和 B 的方案数统计转化为对 PA 和 BQ 的方案数统计。

由于 P 和 Q 都是可逆矩阵，因此方案数不变，我们也通过这一过程得出了方案数只和 C 的秩有关的结论。

来点题

接下来问题可以拆成两部分，动态维护 C 的秩和对每一种秩求出答案。

接下来问题可以拆成两部分，动态维护 C 的秩和对每一种秩求出答案。

第一部分非常简单，只要线段树分治 + 线性基即可。

接下来问题可以拆成两部分，动态维护 C 的秩和对每一种秩求出答案。

第一部分非常简单，只要线段树分治 + 线性基即可。

接着看第二部分，假设当前的问题是求 $AB = C$ 的方案数，其中 C 只有 $c_{ii}(1 \le i \le k)$ 的位置是 1，其它位置都是 0。我们考虑把矩阵 A 看成若干列向量，即 $A = (\vec{a_1}, \vec{a_2}, \cdots, \vec{a_q})$，那么 B 的第 i 列的内容就是 “C 的第 i 列关于 A 中列向量的线性表示”。

接下来问题可以拆成两部分，动态维护 C 的秩和对每一种秩求出答案。

第一部分非常简单，只要线段树分治 + 线性基即可。

接着看第二部分，假设当前的问题是求 $AB = C$ 的方案数，其中 C 只有 $c_{ii}(1 \le i \le k)$ 的位置是 1，其它位置都是 0。我们考虑把矩阵 A 看成若干列向量，即 $A = (\vec{a_1}, \vec{a_2}, \cdots, \vec{a_q})$，那么 B 的第 i 列的内容就是 “C 的第 i 列关于 A 中列向量的线性表示”。

假设 A 的秩是 i，则 A 中会有 $q - i$ 个非基向量，当 C 的某一列在 $\operatorname{Span} A$ 中时，它就会有 2^{q-i} 种表示方法。

因此，我们考虑先枚举 A 的秩 i，再计算 A 的方案数，使得 C 的每一列都在 $\mathrm{Span}\,A$ 中，然后这一方案数乘上 $2^{(q-i)r}$ 即为 A 和 B 的总方案数。

因此，我们考虑先枚举 A 的秩 i，再计算 A 的方案数，使得 C 的每一列都在 $\text{Span}\, A$ 中，然后这一方案数乘上 $2^{(q-i)r}$ 即为 A 和 B 的总方案数。

那么现在的唯一问题就是计算用 q 个 p 维列向量构成的张成 i 维空间（且必须包含一个 k 维子空间）的方案数是多少。

因此，我们考虑先枚举 A 的秩 i，再计算 A 的方案数，使得 C 的每一列都在 $\operatorname{Span} A$ 中，然后这一方案数乘上 $2^{(q-i)r}$ 即为 A 和 B 的总方案数。

那么现在的唯一问题就是计算用 q 个 p 维列向量构成的张成 i 维空间（且必须包含一个 k 维子空间）的方案数是多少。

令记号 $f(n,m,r)$ 表示秩为 r 的 $n\times m$ 矩阵数，$g(i,j)$ 表示一个 i 维空间中的 j 维子空间的个数，则答案等于 $\dfrac{f(p,q,i)g(i,k)}{g(p,k)}$。（因为每个 k 维子空间都是等价的）

1 引入

2 向量表示与矩阵

3 线性空间

4 行列式与特征多项式
- 行列式
- 特征多项式
- 行列式相关定理
- 其它应用

5 Thanks

直观定义

回顾一下定义，线性算子是指一个线性空间到其自身的线性变换。

直观定义

回顾一下定义，线性算子是指一个线性空间到其自身的线性变换。

行列式是一个线性算子的属性，用于描述这个变换将空间放大了几倍。

https://www.bilibili.com/video/BV1ys411472E?p=7

如果你觉得这个定义不够严谨，那么这里给出一个严谨的定义。

如果你觉得这个定义不够严谨，那么这里给出一个严谨的定义。

Definition 4.1.1 (行列式)

存在唯一的从 $n \times n$ 的矩阵空间到数的函数 δ，满足以下性质：

(1) 对于恒等矩阵 I，有 $\delta(I) = 1$。

(2) 函数 δ 对于矩阵的各行是线性的。

(3) 若矩阵的两个相邻行相等，则 $\delta(A) = 0$。

该函数 δ 被称为行列式，矩阵 A 的行列式一般记作 $\det A$ 或 $|A|$。

显然上述定义符合“将空间放大了几倍”的直观定义，且它还具有一些其它性质：

显然上述定义符合“将空间放大了几倍”的直观定义，且它还具有一些其它性质：

(i) 交换矩阵的任意两行，行列式会变成它的相反数。

显然上述定义符合“将空间放大了几倍”的直观定义，且它还具有一些其它性质：

(i) 交换矩阵的任意两行，行列式会变成它的相反数。

(ii) $\det(A) = \sum_{p \text{ is perm}} (-1)^{\sigma(p)} \prod_{i=1}^{n} a_{ip_i}$。($\sigma(p)$ 表示 p 的逆序数)

显然上述定义符合“将空间放大了几倍”的直观定义，且它还具有一些其它性质：

(i) 交换矩阵的任意两行，行列式会变成它的相反数。

(ii) $\det(A) = \sum_{p \text{ is perm}} (-1)^{\sigma(p)} \prod_{i=1}^{n} a_{ip_i}$。($\sigma(p)$ 表示 p 的逆序数)

(iii) $\det A = \det A^T$

显然上述定义符合“将空间放大了几倍”的直观定义，且它还具有一些其它性质：

(i) 交换矩阵的任意两行，行列式会变成它的相反数。

(ii) $\det(A) = \sum_{p \text{ is perm}} (-1)^{\sigma(p)} \prod_{i=1}^{n} a_{ip_i}$。($\sigma(p)$ 表示 p 的逆序数)

(iii) $\det A = \det A^T$

由第二条性质即可得出定义的唯一性，第二条性质也称为行列式的展开式。

由于行列式表示这个变换将空间放大了几倍，那么两个变换复合后，它们的放大倍数就会相乘，因此自然得到 $\det(AB) = \det A \det B$，这一公式也可以通过初等矩阵展开（即高斯消元）来严谨地证明。

由于行列式表示这个变换将空间放大了几倍，那么两个变换复合后，它们的放大倍数就会相乘，因此自然得到 $\det(AB) = \det A \det B$，这一公式也可以通过初等矩阵展开（即高斯消元）来严谨地证明。

一个矩阵的行列式非 0，当且仅当它是一个满秩矩阵。

由于行列式表示这个变换将空间放大了几倍，那么两个变换复合后，它们的放大倍数就会相乘，因此自然得到 $\det(AB) = \det A \det B$，这一公式也可以通过初等矩阵展开（即高斯消元）来严谨地证明。

一个矩阵的行列式非 0，当且仅当它是一个满秩矩阵。

一个矩阵的行列式可以通过高斯消元来 $O(n^3)$ 地计算。

对于一个 $n \times n$ 的矩阵 A，它关于 i, j 的余子式是删去第 i 行和第 j 列后剩下的 $(n-1) \times (n-1)$ 的矩阵的行列式，关于 i, j 的代数余子式是相应余子式乘上 $(-1)^{i+j}$ 得到的值。

对于一个 $n \times n$ 的矩阵 A，它关于 i, j 的余子式是删去第 i 行和第 j 列后剩下的 $(n-1) \times (n-1)$ 的矩阵的行列式，关于 i, j 的代数余子式是相应余子式乘上 $(-1)^{i+j}$ 得到的值。

由行列式的展开式很容易得到以下结论：令 M_{ij} 表示 A 关于 i, j 的代数余子式，则对于任意 i 有 $|A| = \sum_{j=1}^{n} a_{ij} M_{ij}$。上式称为行列式关于第 i 行的按行展开，按列展开同理。

对于一个 $n \times n$ 的矩阵 A，它关于 i, j 的余子式是删去第 i 行和第 j 列后剩下的 $(n-1) \times (n-1)$ 的矩阵的行列式，关于 i, j 的代数余子式是相应余子式乘上 $(-1)^{i+j}$ 得到的值。

由行列式的展开式很容易得到以下结论：令 M_{ij} 表示 A 关于 i, j 的代数余子式，则对于任意 i 有 $|A| = \sum_{j=1}^{n} a_{ij} M_{ij}$。上式称为行列式关于第 i 行的按行展开，按列展开同理。

令 M_{ij} 表示 A 关于 i, j 的代数余子式，如果一个矩阵 B，满足 $B_{ij} = M_{ji}$，则称 B 是 A 的伴随矩阵，记作 A^*，我们有结论 $AA^* = |A|I$，因此 $A^{-1} = \dfrac{A^*}{|A|}$。

Cauchy-Binet 定理

对于 $n \times n$ 的方阵，我们有 $\det(AB) = \det A \det B$，对于 $n \times m$ 的非方阵，我们也有类似的结论。

Cauchy-Binet 定理

对于 $n \times n$ 的方阵，我们有 $\det(AB) = \det A \det B$，对于 $n \times m$ 的非方阵，我们也有类似的结论。

设 A 是一个 $n \times m$ 的矩阵，B 是一个 $m \times n$ 的矩阵。假如 $n > m$，则显然有 $\det(AB) = 0$；假如 $n \leq m$，则有以下定理成立：

Cauchy-Binet 定理

对于 $n\times n$ 的方阵，我们有 $\det(AB)=\det A\det B$，对于 $n\times m$ 的非方阵，我们也有类似的结论。

设 A 是一个 $n\times m$ 的矩阵，B 是一个 $m\times n$ 的矩阵。假如 $n>m$，则显然有 $\det(AB)=0$；假如 $n\leq m$，则有以下定理成立：

Theorem 4.1.1 (Cauchy-Binet 定理)

设 S 是集合 $\{1,2,\cdots,m\}$ 的一个大小为 n 的子集，则有 $\det(AB)=\sum\limits_S\det(A_S)\det(B_S)$。

其中 A_S 表示只保留 A 中下标 S 对应的列后得到的方阵，B_S 表示只保留 B 中下标 S 对应的行后得到的方阵。

证明

$$AB = \begin{bmatrix} \sum_{i=1}^{m} a_{1i}b_{i1} & \sum_{i=1}^{m} a_{1i}b_{i2} & \cdots & \sum_{i=1}^{m} a_{1i}b_{in} \\ \sum_{i=1}^{m} a_{2i}b_{i1} & \sum_{i=1}^{m} a_{2i}b_{i2} & \cdots & \sum_{i=1}^{m} a_{2i}b_{in} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{m} a_{ni}b_{i1} & \sum_{i=1}^{m} a_{ni}b_{i2} & \cdots & \sum_{i=1}^{m} a_{ni}b_{in} \end{bmatrix}。$$

证明

$$AB = \begin{bmatrix} \sum_{i=1}^m a_{1i}b_{i1} & \sum_{i=1}^m a_{1i}b_{i2} & \cdots & \sum_{i=1}^m a_{1i}b_{in} \\ \sum_{i=1}^m a_{2i}b_{i1} & \sum_{i=1}^m a_{2i}b_{i2} & \cdots & \sum_{i=1}^m a_{2i}b_{in} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^m a_{ni}b_{i1} & \sum_{i=1}^m a_{ni}b_{i2} & \cdots & \sum_{i=1}^m a_{ni}b_{in} \end{bmatrix}。$$

由于行列式关于所有行都呈线性，因此我们关于所有行展开，得到 $\sum_{i_1=1}^m \sum_{i_2=1}^m \cdots \sum_{i_n=1}^m \det \begin{bmatrix} a_{1i_1}b_{i_11} & a_{1i_1}b_{i_12} & \cdots & a_{1i_1}b_{i_1n} \\ a_{2i_2}b_{i_21} & a_{2i_2}b_{i_22} & \cdots & a_{2i_2}b_{i_2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{ni_n}b_{i_n1} & a_{ni_n}b_{i_n2} & \cdots & a_{ni_n}b_{i_nn} \end{bmatrix}$。

提取每一行中的 a，得到

$$\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\cdots\sum_{i_n=1}^{m} a_{1i_1}a_{2i_2}\cdots a_{ni_n}\det\begin{bmatrix} b_{i_1 1} & b_{i_1 2} & \cdots & b_{i_1 n} \\ b_{i_2 1} & b_{i_2 2} & \cdots & b_{i_2 n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{i_n 1} & b_{i_n 2} & \cdots & b_{i_n n} \end{bmatrix}。$$

提取每一行中的 a，得到

$$\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\cdots\sum_{i_n=1}^{m} a_{1i_1}a_{2i_2}\cdots a_{ni_n}\det\begin{bmatrix} b_{i_1 1} & b_{i_1 2} & \cdots & b_{i_1 n} \\ b_{i_2 1} & b_{i_2 2} & \cdots & b_{i_2 n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{i_n 1} & b_{i_n 2} & \cdots & b_{i_n n} \end{bmatrix}。$$

如果 $i_1, i_2, \cdots, i_n$ 中有重复的数，则该行列式一定为 0，否则可以将 $i_1, i_2, \cdots, i_n$ 排序，并乘上 $(-1)^{\sigma(i_1,i_2,\cdots,i_n)}$，因此上式变为 $\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\cdots\sum_{i_n=1}^{m}\delta(i_1, i_2, \cdots, i_n)a_{1i_1}a_{2i_2}\cdots a_{ni_n}\det B_S$。

（当 p 中有重复数时，$\delta(p)=0$，否则 $\delta(p)=(-1)^{\sigma(p)}$）

考虑枚举 $i_1, i_2, \cdots, i_n$ 排序后的结果，并将上式改写为

$$\sum_{i_1<i_2<\cdots<i_n}^{m} \sum_{p \text{ is } \mathrm{perm}(i)} (-1)^{\sigma(p)} a_{1p_1} a_{2p_2} \cdots a_{np_n} \det B_S \text{。}$$

考虑枚举 $i_1, i_2, \cdots, i_n$ 排序后的结果，并将上式改写为

$$\sum_{i_1<i_2<\cdots<i_n}^{m} \sum_{p \text{ is perm}(i)} (-1)^{\sigma(p)} a_{1p_1} a_{2p_2} \cdots a_{np_n} \det B_S\text{。}$$

显然上式等于 $\sum_{S} \det A_S \det B_S$，证毕。

特征向量

https://www.bilibili.com/video/BV1ys411472E?p=14

特征向量

https://www.bilibili.com/video/BV1ys411472E?p=14

当我们对一个空间进行线性变换后，会有一些向量，它们发生的变化只相当于乘了一个系数 c。我们把这样的向量叫作这个矩阵的特征向量，这个系数 c 叫作关于这个特征向量的特征值。

即，方阵 A 的特征向量 $\vec{v}$ 和对应的特征值 λ 满足 $A\vec{v}=\lambda\vec{v}$。

特征多项式

把刚才的 $A\vec{v} = \lambda\vec{v}$ 改写为 $A\vec{v} = \lambda I\vec{v}$，即 $(\lambda I - A)\vec{v} = 0$。

特征多项式

把刚才的 $A\vec{v}=\lambda\vec{v}$ 改写为 $A\vec{v}=\lambda I\vec{v}$，即 $(\lambda I-A)\vec{v}=0$。

由于存在非零向量 $\vec{v}$ 满足 $(\lambda I-A)\vec{v}=0$，因此 $\lambda I-A$ 是一个降秩矩阵，即 $|\lambda I-A|=0$。

特征多项式

把刚才的 $A\vec{v} = \lambda\vec{v}$ 改写为 $A\vec{v} = \lambda I\vec{v}$，即 $(\lambda I - A)\vec{v} = 0$。

由于存在非零向量 $\vec{v}$ 满足 $(\lambda I - A)\vec{v} = 0$，因此 $\lambda I - A$ 是一个降秩矩阵，即 $|\lambda I - A| = 0$。

可以证明 λ 是一个特征值当且仅当 $|\lambda I - A| = 0$。

特征多项式

把刚才的 $A\vec{v}=\lambda\vec{v}$ 改写为 $A\vec{v}=\lambda I\vec{v}$，即 $(\lambda I-A)\vec{v}=0$。

由于存在非零向量 $\vec{v}$ 满足 $(\lambda I-A)\vec{v}=0$，因此 $\lambda I-A$ 是一个降秩矩阵，即 $|\lambda I-A|=0$。

可以证明 λ 是一个特征值当且仅当 $|\lambda I-A|=0$。

因此，我们把 $|xI-A|$ 叫作 A 的特征多项式，特征多项式的根即为特征值。由于特征多项式一定有根，因此矩阵一定存在特征值和特征向量。

特征多项式

把刚才的 $A\vec{v}=\lambda\vec{v}$ 改写为 $A\vec{v}=\lambda I\vec{v}$，即 $(\lambda I-A)\vec{v}=0$。

由于存在非零向量 $\vec{v}$ 满足 $(\lambda I-A)\vec{v}=0$，因此 $\lambda I-A$ 是一个降秩矩阵，即 $|\lambda I-A|=0$。

可以证明 λ 是一个特征值当且仅当 $|\lambda I-A|=0$。

因此，我们把 $|xI-A|$ 叫作 A 的特征多项式，特征多项式的根即为特征值。由于特征多项式一定有根，因此矩阵一定存在特征值和特征向量。

提问：描述把平面逆时针旋转 90° 的矩阵的特征向量是什么？

我们前面提到过，相似矩阵描述的是同一个线性变换。而特征值很显然是与基的选取无关的，因此我们有重要结论：

相似矩阵的特征多项式相同

我们前面提到过，相似矩阵描述的是同一个线性变换。而特征值很显然是与基的选取无关的，因此我们有重要结论：

相似矩阵的特征多项式相同

提问：特征多项式相同的矩阵是否一定是相似矩阵？

观察 $xI - A = \begin{bmatrix} -a_{11}+x & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & -a_{22}+x & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & -a_{nn}+x \end{bmatrix}$。

我们可以得到 $[x^0]\det(xI - A) = (-1)^n|A|, [x^{n-1}]\det(xI - A) = -\operatorname{trace}(A), [x^n]\det(xI - A) = 1$，其中 $\operatorname{trace}(A)$ 表示 A 的迹，也就是对角线上的元素之和。

观察 $xI - A = \begin{bmatrix} -a_{11}+x & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & -a_{22}+x & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & -a_{nn}+x \end{bmatrix}$。

我们可以得到 $[x^0]\det(xI-A) = (-1)^n|A|, [x^{n-1}]\det(xI-A) = -\operatorname{trace}(A), [x^n]\det(xI-A) = 1$，其中 $\operatorname{trace}(A)$ 表示 A 的迹，也就是对角线上的元素之和。

由于相似矩阵的特征多项式相同，因此相似矩阵的行列式和迹也都相同。

观察 $xI - A = \begin{bmatrix} -a_{11}+x & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & -a_{22}+x & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & -a_{nn}+x \end{bmatrix}$。

我们可以得到 $[x^0]\det(xI-A) = (-1)^n|A|, [x^{n-1}]\det(xI-A) = -\operatorname{trace}(A), [x^n]\det(xI-A) = 1$，其中 $\operatorname{trace}(A)$ 表示 A 的迹，也就是对角线上的元素之和。

由于相似矩阵的特征多项式相同，因此相似矩阵的行列式和迹也都相同。

又因为 $\det(xI-A) = (x-\lambda_1)(x-\lambda_2)\cdots(x-\lambda_n)$，所以有 $\operatorname{trace}(A) = \lambda_1+\lambda_2+\cdots+\lambda_n$ 和 $\det A = \lambda_1\lambda_2\cdots\lambda_n$。

计算特征多项式

由于相似矩阵的特征多项式相同，因此我们可以在保持相似关系的前提下对矩阵进行消元。

计算特征多项式

由于相似矩阵的特征多项式相同，因此我们可以在保持相似关系的前提下对矩阵进行消元。

前文中也提到了，我们可以在保持相似关系的情况下，把任意一个方阵消成上海森堡矩阵，然后在求上海森堡矩阵的 $\det(xI - A)$。

计算特征多项式

由于相似矩阵的特征多项式相同，因此我们可以在保持相似关系的前提下对矩阵进行消元。

前文中也提到了，我们可以在保持相似关系的情况下，把任意一个方阵消成上海森堡矩阵，然后在求上海森堡矩阵的 $\det(xI-A)$。

对于一个所有元素均为常数的上海森堡矩阵，它的行列式可以以 $O(cnt)$ 的复杂度求出，其中 cnt 表示矩阵中的非零元素个数。具体方法就是枚举最后一列选择的元素，然后转移到前面元素的行列式。

计算特征多项式

由于相似矩阵的特征多项式相同，因此我们可以在保持相似关系的前提下对矩阵进行消元。

前文中也提到了，我们可以在保持相似关系的情况下，把任意一个方阵消成上海森堡矩阵，然后在求上海森堡矩阵的 $\det(xI-A)$。

对于一个所有元素均为常数的上海森堡矩阵，它的行列式可以以 $O(cnt)$ 的复杂度求出，其中 cnt 表示矩阵中的非零元素个数。具体方法就是枚举最后一列选择的元素，然后转移到前面元素的行列式。

上海森堡矩阵的 $\det(xI-A)$ 也可以用同样的方法求出，复杂度 $O(n^3)$。

Cayley-Hamilton 定理

Theorem 4.2.1 (Cayley-Hamilton 定理)

设 $n \times n$ 的方阵 A 的特征多项式是 $f(x)$，有 $f(A) = 0$ 恒成立。

Cayley-Hamilton 定理

Theorem 4.2.1 (Cayley-Hamilton 定理)

设 $n \times n$ 的方阵 A 的特征多项式是 $f(x)$，有 $f(A) = 0$ 恒成立。

Proof 4.2.1

令 B 表示 $xI - A$ 的伴随矩阵，因此有
$B(xI - A) = \det(xI - A)I = f(x)I$。

把等式两侧都看成关于 x 的系数为矩阵的多项式，把 x 用 A 代入，立刻得到 $f(A) = 0$。

对角化

假如一个 $n \times n$ 的方阵 A 有 n 个线性无关的特征向量，那么我们可以让这些特征向量构成一个基，当我们把线性变换 A 放到这个基上考虑的时候，会发现线性变换对于每个基都只构成常数倍的缩放，因此 A 在这个基下的矩阵会是一个对角矩阵。

对角化

假如一个 $n \times n$ 的方阵 A 有 n 个线性无关的特征向量，那么我们可以让这些特征向量构成一个基，当我们把线性变换 A 放到这个基上考虑的时候，会发现线性变换对于每个基都只构成常数倍的缩放，因此 A 在这个基下的矩阵会是一个对角矩阵。

以上过程称之为矩阵的对角化，如果用 P 表示把所有特征向量拼起来得到的矩阵，则 $P^{-1}AP$ 是一个对角矩阵。

对角化

假如一个 $n \times n$ 的方阵 A 有 n 个线性无关的特征向量，那么我们可以让这些特征向量构成一个基，当我们把线性变换 A 放到这个基上考虑的时候，会发现线性变换对于每个基都只构成常数倍的缩放，因此 A 在这个基下的矩阵会是一个对角矩阵。

以上过程称之为矩阵的对角化，如果用 P 表示把所有特征向量拼起来得到的矩阵，则 $P^{-1}AP$ 是一个对角矩阵。

理论上，对角化是非常强的工具，因为对角矩阵具有很多很好的性质。例如，对角矩阵的快速幂非常好计算，而 $(P^{-1}AP)^n = P^{-1}A^nP$，因此可以通过对角化来计算 A^n，该方法可用于得出线性递推数列的通项公式。

对角化

假如一个 $n \times n$ 的方阵 A 有 n 个线性无关的特征向量，那么我们可以让这些特征向量构成一个基，当我们把线性变换 A 放到这个基上考虑的时候，会发现线性变换对于每个基都只构成常数倍的缩放，因此 A 在这个基下的矩阵会是一个对角矩阵。

以上过程称之为矩阵的对角化，如果用 P 表示把所有特征向量拼起来得到的矩阵，则 $P^{-1}AP$ 是一个对角矩阵。

理论上，对角化是非常强的工具，因为对角矩阵具有很多很好的性质。例如，对角矩阵的快速幂非常好计算，而 $(P^{-1}AP)^n = P^{-1}A^nP$，因此可以通过对角化来计算 A^n，该方法可用于得出线性递推数列的通项公式。

不过对角化在 OI 中并不常用，因为特征值和特征向量一般不是整数，且非常难求，此外还不是每个矩阵都能对角化。

若尔当形

若尔当形指这样的方阵：$\begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_t \end{bmatrix}$，其中 $J_1, J_2, \cdots, J_t$ 都是若尔当块。

若尔当块也是一个方阵，其主对角线元素相同，且主对角线左下的斜线的每个元素都是 1，也就是 $\begin{bmatrix} \lambda & & & \\ 1 & \lambda & & \\ & \ddots & \ddots & \\ & & 1 & \lambda \end{bmatrix}$ 的形式。

例如，$\begin{bmatrix} \lambda_1 & & \\ 1 & \lambda_1 & \\ & & \lambda_2 \end{bmatrix}$ 就是一个若尔当形。

例如，$\begin{bmatrix} \lambda_1 & & \\ 1 & \lambda_1 & \\ & & \lambda_2 \end{bmatrix}$ 就是一个若尔当形。

Definition 4.2.1 (若尔当分解)

对于任意 $n \times n$ 复矩阵 A，存在可逆复矩阵 P 使得 $P^{-1}AP$ 有若尔当形，其中每个若尔当块对应一个特征向量，该若尔当块的对角线即为对应特征值。

且线性变换 A 的若尔当形除若尔当块的顺序外是唯一的。

证明可在《代数》的定理 4.7.10 处找到，但较为复杂且困难，故此处略去。

证明可在《代数》的定理 4.7.10 处找到，但较为复杂且困难，故此处略去。

若尔当形对于分析线性变换的结构有重要意义。每个若尔当块对应一个特征值和特征向量，还有若干广义特征向量（定义见《代数》)。

若尔当块的结构也更好的解释了之前“特征多项式相同矩阵是否相似”的问题，现在我们知道了矩阵相似的等价类即为若尔当形。

CF923E Perpetual Subtraction

有一个整数 x，初始时它的值为 $[0,n]$ 中的某一个整数，其中为 i 的概率是 p_i。

每一轮你会把 x 随机替换成 $[0,x]$ 中的一个数，进行 m 轮后，对于每个 $i \in [0,n]$，$x = i$ 的概率。

$n \le 10^5, m \le 10^{18}$，对 998244353 取模。

如果把 $x = i$ 的概率列成向量，那么每一轮的过程就可以看成一个线性变换，写成矩阵形式即为：

$$\vec{p}_i = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ 0 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ 0 & 0 & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{n+1} \end{bmatrix} \vec{p}_{i-1}。$$

把上面的矩阵记为 A，则答案为 $A^m \cdot \vec{p}_0$，其中 $\vec{p}_0$ 即为题目给出的概率分布数组。

现在的核心问题在于处理这个 A^m，我们考虑对 A 进行对角化。

现在的核心问题在于处理这个 A^m，我们考虑对 A 进行对角化。

先考虑这个矩阵的特征多项式，由于它是上三角矩阵，因此 $\det(xI - A)$ 非常好求，就是 $(x-1)(x-\frac{1}{2})\cdots(x-\frac{1}{n+1})$。

现在的核心问题在于处理这个 A^m，我们考虑对 A 进行对角化。

先考虑这个矩阵的特征多项式，由于它是上三角矩阵，因此 $\det(xI - A)$ 非常好求，就是 $(x-1)(x-\frac{1}{2})\cdots(x-\frac{1}{n+1})$。

然后我们需要求出这个矩阵的特征向量，这样才能得出用于对角化的矩阵 P。

上三角矩阵的特征向量也有一些比较好的性质，比如我们考虑前 $n-1$ 行和前 $n-1$ 列构成的矩阵，假设我们已经求出了这个矩阵的所有 $n-1$ 个特征向量，那么这些向量只要多添一维 0，就会成为新的 n 阶矩阵的特征向量，这样我们只需要每次找到一个特征向量即可。

上三角矩阵的特征向量也有一些比较好的性质，比如我们考虑前 $n-1$ 行和前 $n-1$ 列构成的矩阵，假设我们已经求出了这个矩阵的所有 $n-1$ 个特征向量，那么这些向量只要多添一维 0，就会成为新的 n 阶矩阵的特征向量，这样我们只需要每次找到一个特征向量即可。

我们从小到大考虑，当 n 等于 1 时只有一个特征向量 $[1]$。

上三角矩阵的特征向量也有一些比较好的性质，比如我们考虑前 $n-1$ 行和前 $n-1$ 列构成的矩阵，假设我们已经求出了这个矩阵的所有 $n-1$ 个特征向量，那么这些向量只要多添一维 0，就会成为新的 n 阶矩阵的特征向量，这样我们只需要每次找到一个特征向量即可。

我们从小到大考虑，当 n 等于 1 时只有一个特征向量 $\begin{bmatrix}1\end{bmatrix}$。

n 等于 2 时，我们也可以利用上三角矩阵的性质，直接解出特征向量。显然特征向量进行数乘后仍然是特征向量，因此我们先钦定其最后一位为 1，得到方程 $\begin{bmatrix}1 & \frac{1}{2} \\ 0 & \frac{1}{2}\end{bmatrix}\begin{bmatrix}x \\ 1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}x \\ 1\end{bmatrix}$，即 $x+\frac{1}{2}=\frac{1}{2}x$，解得 $x=-1$，即特征向量为 $\begin{bmatrix}-1 \\ 1\end{bmatrix}$。

$n=3$ 时也可以通过相同的方法，逐个递推得到。

设特征向量为 $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$，有 $\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$。

先通过 $\frac{1}{2}y+\frac{1}{3}=\frac{1}{3}y$ 解得 $y=-2$，再通过 $x-1+\frac{1}{3}=\frac{1}{3}x$

解得 $x=1$，因此 $n=3$ 时的特征向量为 $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$。

继续计算更大的特征向量并找规律，会发现第 $i+1$ 个特征向量为 $\left[(-1)^{i+0}\binom{i}{0} \quad (-1)^{i+1}\binom{i}{1} \quad \cdots \quad (-1)^{i+i}\binom{i}{i}\right]^T$。

因此矩阵 P 等于
$$\begin{bmatrix} 1 & -1 & 1 & -1 & \cdots & (-1)^n \\ & 1 & -2 & 3 & \cdots & (-1)^{n+1}\binom{n}{1} \\ & & 1 & -3 & \cdots & (-1)^{n+2}\binom{n}{2} \\ & & & 1 & \cdots & (-1)^{n+3}\binom{n}{3} \\ & & & & \ddots & \vdots \\ & & & & & 1 \end{bmatrix}$$。

可以发现矩阵 P 就是二项式反演的矩阵形式，因此 P^{-1} 就是把 P 的所有元素去掉负号得到的结果。

一个向量左乘 P 和 P^{-1} 都可以通过卷积来快速计算，因此本题复杂度 $O(n\log n)$。

矩阵树定理

Theorem 4.3.1 (矩阵树定理（Matrix-Tree 定理）)

对于任意有向无权图 G，定义矩阵 A 满足 $a_{ij}=\begin{cases}\text{从}i\text{ 到}j\text{ 的边数的相反数} & i\neq j\\ \text{指向}i\text{ 的边数} & i=j\end{cases}$。令 A_i 表示删去 A 的第 i 行第 i 列得到的矩阵，则 $|A_i|$ 等于 G 的以 i 为根的外向树数量。

显然有向图版本的矩阵树定理可以直接推出无向图版本的矩阵树定理。

设该图的点数为 n，边数为 m。构造 $n \times m$ 的矩阵 B 和 $m \times n$ 的矩阵 C，对于第 i 条边，如果它是 $u \to v$ 的，则有 $B_{ui} = 1, B_{vi} = -1, C_{iv} = -1$，$B$ 和 C 的其它位置均为 0。

即 B 的每一列表示一条边，起点填 1 终点填 -1，其它填 0；C 的每一行表示一条边，终点填 -1，其它填 0。

设该图的点数为 n，边数为 m。构造 $n \times m$ 的矩阵 B 和 $m \times n$ 的矩阵 C，对于第 i 条边，如果它是 $u \to v$ 的，则有 $B_{ui} = 1, B_{vi} = -1, C_{iv} = -1$，$B$ 和 C 的其它位置均为 0。

即 B 的每一列表示一条边，起点填 1 终点填 -1，其它填 0；C 的每一行表示一条边，终点填 -1，其它填 0。

直接将 B 和 C 相乘，可以得到 $A = BC$。将 B 删掉第 r 行的矩阵 B_r 与 C 删掉第 r 列的矩阵 C_r 相乘，可得 $A_r = B_r C_r$。

选取图 G 中的 $n-1$ 条边，提取 B_r 中这些边对应的 $n-1$ 列得到一个 $(n-1)\times(n-1)$ 的方阵，记作 B_r'。

如果这 $n-1$ 条边构成以 r 为根的外向树，则 $|B_r'|$ 等于 1 或 -1，否则 $|B_r'|$ 等于 0。

选取图 G 中的 $n-1$ 条边，提取 B_r 中这些边对应的 $n-1$ 列得到一个 $(n-1)\times(n-1)$ 的方阵，记作 B'_r。

如果这 $n-1$ 条边构成以 r 为根的外向树，则 $|B'_r|$ 等于 1 或 -1，否则 $|B'_r|$ 等于 0。

如果 $|B'_r|$ 不为 0，那么提取 C_r 中对应的 $n-1$ 行得到的 C'_r 满足 $|C_r|=|B_r|$。

选取图 G 中的 $n-1$ 条边，提取 B_r 中这些边对应的 $n-1$ 列得到一个 $(n-1)\times(n-1)$ 的方阵，记作 B_r'。

如果这 $n-1$ 条边构成以 r 为根的外向树，则 $|B_r'|$ 等于 1 或 -1，否则 $|B_r'|$ 等于 0。

如果 $|B_r'|$ 不为 0，那么提取 C_r 中对应的 $n-1$ 行得到的 C_r' 满足 $|C_r|=|B_r|$。

对 $A_r=B_rC_r$ 使用 Cauchy-Binet 定理，可知矩阵树定理成立。

LOJ3626 愚蠢的在线法官

给定一个 n 个点的树，第 i 个点的点权为 v_i，和一个长度为 k 的顶点序列 a，求以下矩阵的行列式并对 998244353 取模：

$$\begin{bmatrix} v_{\text{lca}(a_1,a_1)} & v_{\text{lca}(a_1,a_2)} & \cdots & v_{\text{lca}(a_1,a_k)} \\ v_{\text{lca}(a_2,a_1)} & v_{\text{lca}(a_2,a_2)} & \cdots & v_{\text{lca}(a_2,a_k)} \\ \vdots & \vdots & \ddots & \vdots \\ v_{\text{lca}(a_k,a_1)} & v_{\text{lca}(a_k,a_2)} & \cdots & v_{\text{lca}(a_k,a_k)} \end{bmatrix}$$

$\operatorname{lca}(u, v)$ 很容易用向量乘法来描述，因为 $\operatorname{lca}(u, v)$ 到根的链相当于 u 到根的链与 v 到根的链的交。

$\mathrm{lca}(u,v)$ 很容易用向量乘法来描述，因为 $\mathrm{lca}(u,v)$ 到根的链相当于 u 到根的链与 v 到根的链的交。

如果我们构造一个长度为 n 的向量 $\vec{b}_u$，满足对于每一个 u 的祖先 x，$\vec{b}_u$ 的第 x 位为 p_x，其它位为 0；构造令一个长度为 n 的向量 $\vec{b}_v$，对于每一个 v 的祖先 x，$\vec{b}_v$ 的第 x 位为 q_x，其它位为 0。

则有 $\vec{b}_u^T\vec{b}_v = \sum_{x \text{ is an ancestor of } v} p_x q_x$。

因此，假如我们先对点权做一个差分，得到新点权 w_i，满足原点权 $v_i = \sum_{j \text{ is an ancestor of } i} w_j$，则很容易用 w 构造出两个矩阵，第一个矩阵里填 w，第二个矩阵里填 1，使得这两个矩阵的乘积等于题目中的矩阵。

因此，假如我们先对点权做一个差分，得到新点权 w_i，满足原点权 $v_i = \sum_{j \text{ is an ancestor of } i} w_j$，则很容易用 w 构造出两个矩阵，第一个矩阵里填 w，第二个矩阵里填 1，使得这两个矩阵的乘积等于题目中的矩阵。

题目求的是行列式，因此使用 Cauchy-Binet 定理展开。枚举 S 相当于枚举每一种选取 k 个点的方案，第二个矩阵对应的行列式等于把这 k 个点与给定的 k 个点进行祖先配对，如果可以交换两个点的配对点则整个行列式为 0，即要求配对方式唯一，最终的行列式为 1 或 -1。第一个矩阵对应的行列式等于第二个矩阵的行列式乘上选定的 k 的点的 w 点权的积，因此两者相乘等于选定的 k 的点的 w 点权的积或 0，取决于配对方案数是否唯一。

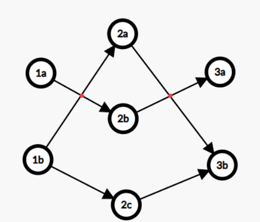
用一个树形 dp 求出配对方案数唯一对应的答案即可。

[NOI2021] 路径交点

有一张共 k 层的分层图，第 i 层有 n_i 个顶点，第 1 层与第 k 层的点数相同，且其它层的点数满足 $n_1 \le n_i \le 2n_1$。第 i 层到第 $i+1$ 层连有一些有向边，全图只有这样的边，且没有重边。

一个方案指 n_1 条从第 1 层到第 k 层的路径，且没有两条路径经过了重复点。求路径交点个数为偶数的方案数比为奇数的方案数多多少，$n_1, k \le 100$，对 998244353 取模。

下图的答案为 $2-0=2$：



注意到如果删去后一层中无用的点，则两层间的路径可以视为一个排列，产生的交点数就等于排列的逆序对数。

注意到如果删去后一层中无用的点，则两层间的路径可以视为一个排列，产生的交点数就等于排列的逆序对数。

因此，对于 $k=2$ 的情况，答案就等于二分图邻接矩阵的行列式值。

注意到如果删去后一层中无用的点，则两层间的路径可以视为一个排列，产生的交点数就等于排列的逆序对数。

因此，对于 $k=2$ 的情况，答案就等于二分图邻接矩阵的行列式值。

对于 $k=3$ 的情况，如果我们枚举第二层使用了哪 n_1 个点，则构成了 Cauchy-Binet 定理的形式，因此答案就等于两个二分图邻接矩阵相乘后的行列式值。

$k>3$ 的情况也完全相同，只是多套了几层 Cauchy-Binet 定理而已，故答案等于所有 $k-1$ 个二分图邻接矩阵相乘后的行列式值。

LGV 引理

Theorem 4.3.2 (LGV 引理)

给定一张有向无环图，和 n 个起点 $a_1, a_2, \ldots, a_n$， n 个终点 $b_1, b_2, \ldots, b_n$ 共 $2n$ 个互不相同的点。

对于一个 n 阶排列 p，我们用 $f(p)$ 表示，选取 n 条不交路径，第 i 条路径起点为 a_i 终点为 b_{p_i} 的方案数。则有 $\sum\limits_p (-1)^{\sigma(p)} f(p)$ 等于以下矩阵的行列式：

$$\begin{bmatrix} e(a_1, b_1) & e(a_1, b_2) & \cdots & e(a_1, b_n) \\ e(a_2, b_1) & e(a_2, b_2) & \cdots & e(a_2, b_n) \\ \vdots & \vdots & \ddots & \vdots \\ e(a_n, b_1) & e(a_n, b_2) & \cdots & e(a_n, b_n) \end{bmatrix}$$

其中 $e(u, v)$ 表示从 u 到 v 的路径条数。

显然该矩阵的行列式在套用行列式展开式后等于 $\sum\limits_{p}(-1)^{\sigma(p)}g(p)$，其中 $g(p)$ 表示，选取 n 条任意路径，第 i 条路径起点为 a_i 终点为 b_{p_i} 的方案数。

假如一个路径方案中有两条路径相交，那么交换这两条路径交点之后的部分就会交换它们的终点。交换后的方案的排列奇偶性一定会改变，这也意味着两个方案的 $(-1)^{\sigma(p)}$ 互为相反数，因此会相互抵消，抵消之后就会只剩下不交路径的方案。

[NOI2021] 路径交点

接着看刚才的题，显然分层图是个 DAG，一个方案就是一组 DAG 上的不交路径。

[NOI2021] 路径交点

接着看刚才的题，显然分层图是个 DAG，一个方案就是一组 DAG 上的不交路径。

设一个方案的交点个数为 c，方案终点关于起点的排列为 p，仔细思考就会发现有 $(-1)^c = (-1)^{\sigma(p)}$。

[NOI2021] 路径交点

接着看刚才的题，显然分层图是个 DAG，一个方案就是一组 DAG 上的不交路径。

设一个方案的交点个数为 c，方案终点关于起点的排列为 p，仔细思考就会发现有 $(-1)^c = (-1)^{\sigma(p)}$。

因此直接套用 LGV 引理，也会得到跟之前做法相同的结论。

1 引入

2 向量表示与矩阵

3 线性空间

4 行列式与特征多项式
- 行列式
- 特征多项式
- 行列式相关定理
- 其它应用

5 Thanks

$\det(Ax+B)$（qoj59）

给定一个 $n \times n$ 的矩阵，它的每一位都是一个 $ax+b$ 形式的一次多项式，求它的行列式值。（即一个 n 次多项式）

要求复杂度 $O(n^3)$，且对 998244353 取模。

考虑一种特殊的高斯消元，来把这个矩阵消成与特征多项式相似的形式，也就是把除主对角线以外元素的一次项都消掉。

考虑一种特殊的高斯消元，来把这个矩阵消成与特征多项式相似的形式，也就是把除主对角线以外元素的一次项都消掉。

仍然是和普通高斯消元一样的顺序，先用第一行去把第一列都消掉。如果第一行第一个元素没有一次项，就在第一行更右边的列中找一个有一次项的元素，并交换两列。如果第一行一整行都没有一次项，就整行同乘 x，并回到上一步。如果乘了 $n+1$ 个 x，则行列式为 0，直接结束消元。之后的行与第一行相同。

循环矩阵行列式

给定一个 $n \times n$ 的矩阵 A 和一个长度为 n 的序列 b，矩阵 A 满足 $a_{ij} = b_{(i+j) \bmod n}$，求 $\det A$。

由于 $\det A = \lambda_1\lambda_2\cdots\lambda_n$，因此我们考虑求出这个矩阵的所有特征值再相乘。

由于 $\det A = \lambda_1\lambda_2\cdots\lambda_n$，因此我们考虑求出这个矩阵的所有特征值再相乘。

很容易发现 $\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}$ 是这个矩阵的一个特征向量，其特征值为 $b_1+b_2+\cdots+b_n$，但其它的特征值似乎不是很好找。

由于 $\det A = \lambda_1\lambda_2\cdots\lambda_n$，因此我们考虑求出这个矩阵的所有特征值再相乘。

很容易发现 $\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}$ 是这个矩阵的一个特征向量，其特征值为 $b_1+b_2+\cdots+b_n$，但其它的特征值似乎不是很好找。

我们先考虑 $n=2$ 的情况，发现另一个特征向量是 $\begin{bmatrix}1\\-1\end{bmatrix}$，特征值为 $a-b$。

再考虑 $n=3$ 的情况，即 $\det\begin{bmatrix} x-a & -b & -c \\ -b & x-c & -a \\ -c & -a & x-b \end{bmatrix}=0$。

再考虑 $n=3$ 的情况，即 $\det\begin{bmatrix} x-a & -b & -c \\ -b & x-c & -a \\ -c & -a & x-b \end{bmatrix}=0$。

我没去算过，反正结论就是剩下两个特征向量分别是 $\begin{bmatrix} \omega \\ \omega^2 \\ 1 \end{bmatrix}$ 和 $\begin{bmatrix} \omega^2 \\ \omega \\ 1 \end{bmatrix}$，特征值分别是 $\omega a+\omega^2 b+c$ 和 $\omega^2 a+\omega b+c$。

再考虑 $n=3$ 的情况，即 $\det\begin{bmatrix} x-a & -b & -c \\ -b & x-c & -a \\ -c & -a & x-b \end{bmatrix}=0$。

我没去算过，反正结论就是剩下两个特征向量分别是 $\begin{bmatrix} \omega \\ \omega^2 \\ 1 \end{bmatrix}$ 和 $\begin{bmatrix} \omega^2 \\ \omega \\ 1 \end{bmatrix}$，特征值分别是 $\omega a+\omega^2 b+c$ 和 $\omega^2 a+\omega b+c$。

那不难发现 n 阶循环矩阵的特征向量也是这个形式，因此 $\det A=\prod_{i=0}^{n-1}\sum_{j=0}^{n-1}\omega_n^{ij}b_j$。

结式

假如恰好存在模意义下的 n 次单位根，则上述等式可以 $O(n^2)$ 求值或借助多项式 $O(n\log n)$ 求值，否则似乎还是很困难。

结式

假如恰好存在模意义下的 n 次单位根，则上述等式可以 $O(n^2)$ 求值或借助多项式 $O(n\log n)$ 求值，否则似乎还是很困难。

因此，这里引出结式的概念：

Definition 4.4.1 (结式)

给定两个多项式 f 和 g，设 g 的所有根为 $r_1, r_2, \ldots, r_k$，则 f 和 g 的结式 (resultant) 是一个数，等于 $f(r_1)f(r_2)\cdots f(r_k)$，记作 $\operatorname{res}(f,g)$。

在本题中，答案即为 $f(x) = b_0 + b_1x + b_2x^2 + \cdots + b_{n-1}x^{n-1}$ 和 $g(x) = x^n - 1$ 的结式 $\operatorname{res}(f,g)$。

结式有两条重要性质：

(1) 如果 f, g 的最高次系数均为 1，则 $\operatorname{res}(f, g) = (-1)^{\deg f \cdot \deg g} \operatorname{res}(g, f)$

(2) $\operatorname{res}(cf, g) = c^{\deg g} \operatorname{res}(f, g)$

(3) $\operatorname{res}(f, g) = \operatorname{res}(f \bmod g, g)$

结式有两条重要性质:

(1) 如果 f, g 的最高次系数均为 1，则 $\operatorname{res}(f, g) = (-1)^{\deg f \cdot \deg g} \operatorname{res}(g, f)$

(2) $\operatorname{res}(cf, g) = c^{\deg g} \operatorname{res}(f, g)$

(3) $\operatorname{res}(f, g) = \operatorname{res}(f \bmod g, g)$

其中第一条性质可以通过把 f 因式分解来得到。

结式有两条重要性质:

(1) 如果 f, g 的最高次系数均为 1，则 $\operatorname{res}(f, g) = (-1)^{\deg f \cdot \deg g} \operatorname{res}(g, f)$

(2) $\operatorname{res}(cf, g) = c^{\deg g} \operatorname{res}(f, g)$

(3) $\operatorname{res}(f, g) = \operatorname{res}(f \bmod g, g)$

其中第一条性质可以通过把 f 因式分解来得到。

于是我们可以通过类似求多项式 $\gcd$ 的方法求多项式的结式，可以 $O(n^2)$ 朴素计算，可能可以借助高科技更快计算。

Thanks