矩阵初步

Chaigidel

效实中学

矩阵基础

运算法则

A 是 $m \times n$ 矩阵，B 是 $n \times p$ 矩阵

$C = AB$ 是 $m \times p$ 矩阵

$$C = (c_{ij}), \ (1 \leq i \leq m, \ 1 \leq j \leq p)$$
$$c_{ij} = \sum_{k=1}^{n} a_{ik} \times b_{kj}$$

$C = c_{ij} = A$ 的第 i 行逐个乘 B 的第 j 列

计算复杂度 $O(mnp)$

计算训练

$$\begin{bmatrix} 2 & -1 & 1 \\ 1 & 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 & 0 \\ 3 & 1 & 0 & -1 \\ -2 & 0 & 1 & 2 \end{bmatrix} =$$

计算训练

$$\begin{bmatrix} 2 & -1 & 1 \\ 1 & 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 & 0 \\ 3 & 1 & 0 & -1 \\ -2 & 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} -3 & 3 & -1 & 3 \\ 6 & 5 & 1 & 1 \end{bmatrix}$$

基本性质

显然

$$ABC = A(BC)$$
$$A(B + C) = AB + AC$$

但是

$$AB \neq BA$$

基本性质

显然

$$ABC = A(BC)$$
$$A(B+C) = AB + AC$$

但是

$$AB \neq BA$$

由于结合律成立，故计算 A^k 可以用快速幂

$O(n^3 \log k)$

提问时间

在实数乘法里 $a \times 1 = a$

若 A 为 $n \times n$ 的矩阵

矩阵 I_n 对任意 A 均有 $A\mathrm{I}_n = A$

$$\mathrm{I}_n =$$

提问时间

在实数乘法里 $a \times 1 = a$

若 A 为 $n \times n$ 的矩阵

矩阵 I_n 对任意 A 均有 $A\mathrm{I}_n = A$

$$\mathrm{I}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

扩展

Min-Plus 矩阵

$$c_{i,j}=\min_{k=1}^{n}\{a_{i,k}+b_{k,j}\}$$

易证其满足结合律

加速转移

矩阵的本质是**转移**，从一个状态转移到另一个状态

当转移可以复用的时候就可以用快速幂

题目中的转移矩阵常常比较小且有特殊规律

可以利用手动循环展开的方式减小常数

练一练

洛谷 P1939

定义数列 a 满足

$$a_n = a_{n-1} + a_{n-3}$$
$$a_1 = a_2 = a_3 = 1$$

求

$$a_n \bmod 1000000007 \quad (n \leq 2 \times 10^9)$$

由于每项只和前 3 项有关，可用 1×3 的矩阵表示数列

然后考虑转移

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \to \begin{bmatrix} a_2 & a_3 & a_4 \end{bmatrix} = \begin{bmatrix} a_2 & a_3 & a_1 + a_3 \end{bmatrix}$$

得到转移矩阵 $\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}^n = \begin{bmatrix} a_{n+1} & a_{n+2} & a_{n+3} \end{bmatrix}$$

数列

$$f_1 = f_2 = 0$$
$$f_n = 7f_{n-1} + 6f_{n-2} + 5n + 4 \times 3^n$$

$$\begin{bmatrix} f_n & f_{n-1} & n & 3^n & 1 \end{bmatrix} \times \begin{bmatrix} 7 & 1 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 \\ 5 & 0 & 1 & 0 & 0 \\ 12 & 0 & 0 & 3 & 0 \\ 5 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{n+1} & f_n & n+1 & 3^{n+1} & 1 \end{bmatrix}$$

「THUSCH 2017」大魔法师

~~中二病患者~~ 大魔法师小 L 制作了 n 个魔力水晶球，每个水晶球有水、火、土三个属性的能量值。小 L 把这 n 个水晶球在地上从前向后排成一行，然后开始今天的魔法表演。

我们用 A_i, B_i, C_i 分别表示从前向后第 i 个水晶球（下标从 1 开始）的水、火、土的能量值。

小 L 计划施展 m 次魔法。每次，他会选择一个区间 $[l, r]$，然后施展以下 3 大类、7 种魔法之一：

1. 魔力激发：令区间里每个水晶球中**特定属性**的能量爆发，从而使另一个**特定属性**的能量增强。具体来说，有以下三种可能的表现形式：

    - 火元素激发水元素能量：令 $A_i = A_i + B_i$。
    - 土元素激发火元素能量：令 $B_i = B_i + C_i$。
    - 水元素激发土元素能量：令 $C_i = C_i + A_i$。

    需要注意的是，增强一种属性的能量并不会改变另一种属性的能量，例如 $A_i = A_i + B_i$ 并不会使 B_i 增加或减少。

2. 魔力增强：小 L 挥舞法杖，消耗自身 v 点法力值，来改变区间里每个水晶球的**特定属性**的能量。具体来说，有以下三种可能的表现形式：

    - 火元素能量定值增强：令 $A_i = A_i + v$。
    - 水元素能量翻倍增强：令 $B_i = B_i \cdot v$。
    - 土元素能量吸收融合：令 $C_i = v$。

3. 魔力释放：小 L 将区间里所有水晶球的能量聚集在一起，融合成一个新的水晶球，然后送给场外观众。生成的水晶球每种属性的能量值等于区间内所有水晶球对应能量值的代数和。**需要注意的是，魔力释放的过程不会真正改变区间内水晶球的能量**。

值得一提的是，小 L 制造和融合的水晶球的原材料都是定制版的 OI 工厂水晶，所以这些水晶球有一个能量阈值 998244353。当水晶球中某种属性的能量值大于等于这个阈值时，能量值会自动对阈值取模，从而避免水晶球爆炸。

小 W 为小 L（唯一的）观众，围观了整个表演，并且收到了小 L 在表演中融合的每个水晶球。小 W 想知道，这些水晶球蕴涵的三种属性的能量值分别是多少。

先只考虑一个，如何写成矩阵的形式

- $A_i = A_i + B_i$
- $B_i = B_i + C_i$
- $C_i = C_i + A_i$

先只考虑一个，如何写成矩阵的形式

- $A_i = A_i + B_i$
- $B_i = B_i + C_i$
- $C_i = C_i + A_i$

$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a+b & b & c \end{bmatrix}$$

- $A_i = A_i + v$
- $B_i = B_i \cdot v$
- $C_i = v$

- $A_i = A_i + v$
- $B_i = B_i \cdot v$
- $C_i = v$

$$\begin{bmatrix} a & b & c & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ v & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a+v & b & c & 1 \end{bmatrix}$$

- $A_i = A_i + v$
- $B_i = B_i \cdot v$
- $C_i = v$

$$\begin{bmatrix} a & b & c & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ v & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a+v & b & c & 1 \end{bmatrix}$$

由于矩阵的结合律和分配律，可以轻易地推广到序列之中

杂题选讲

摆花

一排 n 个位置，m 种花。每个位置可以摆花也可以不摆花，有些花如果摆在相邻的位置（隔着一个空的位置不算相邻），就不好看了。假定每种花数量无限，求摆花的方案数。

第一行有两个用空格隔开的正整数 n， m 。

接下来的 m 行，每行有 m 个字符 1 或 0 , 若第 i 行第 j 列为 1 ，则表示第 i 种花和第 j 种花不能排在相邻的位置，输入保证对称（提示：同一种花可能不能排在相邻位置）。

$n \leq 1000000000$， $m \leq 100$

定义一种花，叫做不摆花 ，它能和任意花相邻

定义一种花，叫做不摆花，它能和任意花相邻

根据加法原理，可以得到转移方程，n 是位置，i 是花的种类

定义一种花，叫做不摆花，它能和任意花相邻

根据加法原理，可以得到转移方程，n 是位置，i 是花的种类

$$f(n,i)=\sum_{j=0}^{m} f(n-1,j)[a_{ij}=0]$$

非常自然地可以想到用矩阵进行快速转移

定义一种花，叫做不摆花，它能和任意花相邻

根据加法原理，可以得到转移方程，n 是位置，i 是花的种类

$$f(n,i)=\sum_{j=0}^{m} f(n-1,j)[a_{ij}=0]$$

非常自然地可以想到用矩阵进行快速转移

定义 $g(x)=\begin{cases}1, & x=0\\0, & x=1\end{cases}$

则转移矩阵 $T_{i,j}=g(a_{ij})$

「NOIP 2018」保卫王国

n 点的联通树，可以花 p_i 代价把 i 点染色，要求任 2 个相邻点至少有 1 个被染色。给出 m 组询问，每次强制两个点的状态 (染/不染)，求出每次的最小花费。

$n,\ m \le 10^5$

首先一个显然的 DP

$$f_{u0} = \sum f_{v1}$$
$$f_{u1} = \sum \min\{f_{v0}, f_{v1}\}$$

SPOJ GSS3 / P4513 小白逛公园

给定长为 $n \le 10^5$ 的序列 a ，支持两种操作

- 单点修改

  即给定 i, d ，$a_i = d$

- 求区间最大子段和

  即给定 a, b ，输出 $\max_{[l,r]\in[a,b]}\{\sum_{i=l}^{r} a_i\}$

操作数 $m \le 10^5$

SPOJ GSS3 / P4513 小白逛公园

给定长为 $n \le 10^5$ 的序列 a ，支持两种操作

- 单点修改

  即给定 $i, \ d$ ， $a_i = d$

- 求区间最大子段和

  即给定 $a, \ b$ ，输出 $\max_{[l,r]\in[a,b]}\{\sum_{i=l}^{r} a_i\}$

操作数 $m \le 10^5$

这道题之前讲过，所以待会 random

考虑无修，$a = 1,\ b = n$ 如何做

考虑无修，$a=1$，$b=n$ 如何做

```
for(int i = 1, ans = -inf, cur = 0 ; i <= n ; ++ i) {
    ans = max(ans, cur + a[i], a[i]);
    cur = max(cur + a[i], a[i]);
}
```

考虑无修，$a = 1, \ b = n$ 如何做

```
for(int i = 1, ans = -inf, cur = 0 ; i <= n ; ++ i) {
    ans = max(ans, cur + a[i], a[i]);
    cur = max(cur + a[i], a[i]);
}
```

$$\begin{bmatrix} cur_{i-1} & ans_{i-1} \end{bmatrix} \to \begin{bmatrix} cur_i & ans_i \end{bmatrix}$$

发现 1×2 的矩阵不能表达

发现 1×2 的矩阵不能表达

$$\begin{bmatrix} cur_{i-1} & ans_{i-1} & 0 \end{bmatrix} \begin{bmatrix} a_i & a_i & -\infty \\ -\infty & 0 & -\infty \\ a_i & a_i & 0 \end{bmatrix} = \begin{bmatrix} cur_i & ans_i & 0 \end{bmatrix}$$

这里的乘法为 max-plus

需要维护区间矩阵乘积，线段树即可

「LibreOJ 6208」树上询问

有一棵 n 节点的树，根为 1 号节点。每个节点有两个权值 k_i, t_i，初始值均为 0。

给出三种操作：1. $\mathrm{Add}(x, d)$ 操作：将 x 到根的路径上所有点的 $k_i \leftarrow k_i + d$ 2. $\mathrm{Mul}(x, d)$ 操作：将 x 到根的路径上所有点的 $t_i \leftarrow t_i + d \times k_i$ 3. $\mathrm{Query}(x)$ 操作：询问点 x 的权值 t_x

$n,\ m \leq 100000,\ -10 \leq d \leq 10$

「LibreOJ 6208」树上询问

有一棵 n 节点的树，根为 1 号节点。每个节点有两个权值 k_i, t_i，初始值均为 0。

给出三种操作：1. $\mathrm{Add}(x, d)$ 操作：将 x 到根的路径上所有点的 $k_i \leftarrow k_i + d$ 2. $\mathrm{Mul}(x, d)$ 操作：将 x 到根的路径上所有点的 $t_i \leftarrow t_i + d \times k_i$ 3. $\mathrm{Query}(x)$ 操作：询问点 x 的权值 t_x

$n,\ m \le 100000,\ -10 \le d \le 10$

对于树上的问题，树链剖分即可转换为序列上的问题

树链剖分不是本次的重点，有兴趣可以自学

所以把树看成序列即可，即树退化为链

解法一

直接用线段树维护

维护 (a, b, c)，$t = ab + c$，a 相当于 k

对于 2 操作，$(a, b + d, c)$

对于 1 操作，$(a + d, b, c - bd)$

下放呢?

解法一

直接用线段树维护

维护 (a, b, c) ，$t = ab + c$ ，a 相当于 k

对于 2 操作，$(a, b + d, c)$

对于 1 操作，$(a + d, b, c - bd)$

下放呢?

父为 (a, b, c)

子从 (d, e, f) 更新为 $(d + a, b + e, c + f - ae)$

解法一

直接用线段树维护

维护 (a, b, c)，$t = ab + c$，a 相当于 k

对于 2 操作，$(a, b + d, c)$

对于 1 操作，$(a + d, b, c - bd)$

下放呢?

父为 (a, b, c)

子从 (d, e, f) 更新为 $(d + a, b + e, c + f - ae)$

你会发现我根本讲不清楚 ~~因为我没写过~~

解法二

这是一个比较复杂的线段树维护问题

对于 1 操作 $\begin{bmatrix} k & t \end{bmatrix} \to \begin{bmatrix} k+d & t \end{bmatrix}$

对于 2 操作 $\begin{bmatrix} k & t \end{bmatrix} \to \begin{bmatrix} k & t+kd \end{bmatrix}$

解法二

这是一个比较复杂的线段树维护问题

对于 1 操作 $\begin{bmatrix} k & t \end{bmatrix} \to \begin{bmatrix} k+d & t \end{bmatrix}$

对于 2 操作 $\begin{bmatrix} k & t \end{bmatrix} \to \begin{bmatrix} k & t+kd \end{bmatrix}$

$$\begin{bmatrix} k & t & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ d & 0 & 1 \end{bmatrix} = \begin{bmatrix} k+d & t & 1 \end{bmatrix}$$

$$\begin{bmatrix} k & t & 1 \end{bmatrix} \begin{bmatrix} 1 & d & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} k & t+d\times k & 1 \end{bmatrix}$$

「Codeforces 718C」Sasha and Array

长度为 n 的序列 a，支持两种操作

1. 区间加法，即 `rep(i, l, r) a[i] += x;`
2. 输出 $\sum_{i=l}^{r} f(a_i)$

$f_0 = 0,\ f_1 = 1,\ f_n = f_{n-1} + f_{n-2}$

$$\begin{bmatrix} f_{i-1} & f_i \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} f_i & f_{i+1} \end{bmatrix}$$

$$\begin{bmatrix} f_{i-1} & f_i \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} f_i & f_{i+1} \end{bmatrix}$$

$$\begin{bmatrix} f_{a-1} + f_{b-1} & f_a + f_b \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} f_a + f_b & f_{a+1} + f_{b+1} \end{bmatrix}$$

每个节点维护 $s = \begin{bmatrix} \sum f_{i-1} & \sum f_i \end{bmatrix}$，lazytag

s_{12} 即为区间的答案

合并 `s = ls->s + rs->s`

下放 `ls/rs->s *= pow(tag), ls/rs->tag + tag, tag = 0`

卡常

「SDOI2009」HH 去散步

给定 n 个点，m 条边的无向图，问从 s 到 t 走 t 的长度的方案数。

边长均为 1，$n \le 50,\ m \le 60,\ t \le 2^{30}$

$$f_{t,e} = \sum f_{t,e'}$$

$f_{t,e}$ 表示总共走 t 步，最后走的边为 e

e' 需要满足 `e'.u = e.v, e'.v != e.u`

$$f_{t,e} = \sum f_{t,e'}$$

$f_{t,e}$ 表示总共走 t 步，最后走的边为 e

e' 需要满足 `e'.u = e.v, e'.v != e.u`

f_t 只和 f_{t-1} 有关，并且转移方式固定

尝试构造矩阵