

NOIP 模拟赛

LSK

试题数目	3
时间限制	3.5h
空间限制	以硬件资源为限

题目名称	落英	撷芳	远征
题目类型	传统	传统	传统
目录	petal	herb	expedition
可执行文件名	petal	herb	expedition
输入文件名	petal.in	herb.in	expedition.in
输出文件名	petal.out	herb.out	expedition.out
单测试点时限	4.0s	6.0s	2.0s
单测试点内存限制	512MB	512MB	512MB
子任务数	10	6	6
分值	100	100	100
C++语言编译选项	-O2 -lm	-O2 -lm	-O2 -lm
C 语言编译选项	-O2 -lm	-O2 -lm	-O2 -lm
Pascal 语言编译选项	-O2	-O2	-O2

提示及鸣谢：

1. 本套试题的所有题目都采用捆绑测试，每题有若干个子任务；对于每个子任务，你只有通过了其中所有的测试点才能获得该子任务的分值。

2. 子任务可能包含依赖关系：如果一个子任务的符合限制的测试数据集合是另一个子任务的子集，那么只有通过了限制更紧的子任务才能获得限制较松子任务的分值，测试点不再重复出现；为了使你更明确，每题都标出了子任务的依赖关系。

3. 题目难度与题目顺序不一定相关。

4. 每题时间限制都在标准程序运行时间的两倍以上，如果你发现你的程序运行超过时限，请你更多考虑优化算法的时间复杂度而非常数因子。

5. 评测在 XJOI 在线评测系统上进行，你可以到 210.33.19.103:9000 的自定义测试中心测试你的程序在最终评测时的运行速度。

6. 每题都下发若干样例文件，这些数据都包含在最终测试数据中，数据强度与其他测试点大致相当，你可以合理利用这些数据全面测试你的程序。

7. 我们提供了 convolution.cpp 来帮助你解题，它的用法在 convolution.pdf 中有更详细的介绍，如果你不是 C++语言的选手，请忽略它们。

8. 由于出题人水平有限，难免会有疏忽之处，如果发现试题中的各种问题请联系出题人及时更正，希望大家能做题愉快。

9. 感谢褚写庭同学为这套比赛中的题目验题。

落英

背景

花谢花飞花满天 红消香断有谁怜
游丝软系飘春榭 落絮轻沾扑绣帘

题目描述

每每落红之期，总是伤怀之时。每个人，每个时刻目睹那鲜艳的生命坠落的一刻，都会有不同的理解，听到不同的——生命的声音。

层层叠叠的落花往往让人不忍践踏，漫天落花更容易让人感怀。

有 n 个院落，每个院落都有许多花树。初始时每个院落就有若干已落的花瓣，之后的 T 个时刻，每个时刻会有一片花瓣落下，花瓣落下的位置是在每一棵花树中等概率随机的。最终每个院落的新增的花瓣数为这个院落的所有花树新掉落的花瓣数的总和。

设第 i 个院落最终花瓣数为 num_i ， k 为一个固定参数，定义感伤值为以下公式计算出的结果：

$$\sum_{i=1}^n num_i^k$$

现在请你计算，感伤值的数学期望是多少，对 998244353 取模输出。

注：数学期望指试验中每次可能结果的概率乘以其结果的总和。

输入格式

从文件 `petal.in` 中读入数据：

第一行一个整数表示数据所属的子任务编号，子任务含义见后文；

第二行三个整数 T, n, k 分别表示时刻数，院落数，和感伤值的计算参数；

接下来一行 n 个整数，第 i 个整数表示第 i 个院落的花树数量 a_i ；

接下来一行 n 个整数，第 i 个整数表示第 i 个院落初始的已落花瓣数量 va_i 。

输出格式

输出到文件 `petal.out` 中；

输出一行，一个在 $[0, 998244352]$ 中的整数，表示感伤值的数学期望对 998244353 取模的值。

可以证明答案一定可以表示为 $\frac{p}{q}$ (q 不为 998244353 的倍数, p, q 互质) 的形式，那么你只需要输出 $p \cdot q^{-1}$ 即可，其中 $q \cdot q^{-1} \equiv 1 \pmod{998244353}$ ，可以证明这样的

q^{-1} 是唯一的。

样例输入 1

```
1
2 2 3
3 2
1 0
```

样例输出 1

```
838525272
```

样例解释

感伤值为 27 的概率为 0.36;

感伤值为 9 的概率为 0.64;

感伤值的数学期望为 15.48，取模后为 838525272。

样例输入 2~7

见选手文件下的 petal2~7.in。

样例输出 2~7

见选手文件下的 petal2~7.ans。

子任务

对于所有的测试数据，保证：

$$0 \leq T \leq 10^{18}$$

$$1 \leq n \leq 10$$

$$1 \leq k \leq 50000$$

$$1 \leq a_i \leq 50000$$

$$0 \leq va_i \leq 998244352$$

子任务编号	分值	特殊性质	依赖的子任务
1	5	$T, n, k \leq 5$	无
2	11	$T \leq 100000$	1
3	17	性质 1, 2, 3	无
4	5	性质 2, 3	3
5	4	性质 1, 2	3
6	13	性质 2	3, 4, 5
7	15	性质 1, 3	3
8	5	性质 3	3, 4, 7
9	8	性质 1	1, 3, 5, 7
10	17	无	1~9

性质 1: $k \leq 50$

性质 2: $n=2$ 且 $a_1 = a_2$

性质 3: 所有 va_i 都相等

提示

费马小定理: $a^{p-1} \equiv 1(\text{mod } p)$ (p 为质数, a, p 互质)

撷芳

背景

揽木根以结茝兮 贯薜荔之落蕊
矫菌桂以纫蕙兮 索胡绳之纚纚

题目描述

你能感受到那从几千年前留下的兰蕙的芳香吗？让我们捡拾经过历史洗礼的歌卷，聆听那悠扬悲怆的长歌，从群芳丛中走过，编织起属于自己的精神画卷。

远古的园圃里葱葱郁郁生长着 n 种香草，第 i 种香草的芳香值为 a_i 。

现在你要选择两棵香草编织在一起，根据人们的经验，芳香值为 x 和芳香值为 y 的香草编织起来的成品的芳香值为 x 和 y 的最小公倍数。

现在你要求出每一种编织方法的芳香值之和，对 998244353 取模。

形式化地说，设 $f(i,j)$ 表示第 i 种香草和第 j 种香草编织起来的芳香值，你要求的答案可以表示为：

$$(\sum_{i=1}^n \sum_{j=1}^n f(i,j)) \bmod 998244353$$

由于香草的芳香值受到各种自然环境的影响，你需要对 a 数组做区间修改的操作并回答修改后的答案(修改进行后不会撤销，后续修改在修改后的数组上进行)。

输入格式

从文件 herb.in 中读入数据：

第一行一个整数表示数据所属的子任务编号，子任务含义见后文；

第二行一个整数 n 表示香草种数；

接下来一行 n 个整数，第 i 个整数表示第 i 种香草的芳香值 a_i ；

接下来一行一个整数 q 表示修改的数量；

接下来 q 行，每行 3 个整数 l, r, va ，表示对于每个 $i \in [l, r]$ ，将 a_i 修改为 va 。

输出格式

输出到文件 herb.out 中；

输出 $q+1$ 行，每行一个在 $[0, 998244352]$ 中的整数，第一行表示没有修改时的答案，第 $2 \sim q+1$ 行，第 i 行表示进行完第 $i-1$ 次修改后的答案。

样例输入 1

```

1
3
2 3 6
3
1 2 1
2 3 3
1 1 6

```

样例输出 1

```

47
34
25
42

```

样例输入 2~5

见选手文件下的 herb2~5.in。

样例输出 2~5

见选手文件下的 herb2~5.ans。

子任务

对于所有的测试数据，保证：

$$1 \leq n \leq 100000, 0 \leq q \leq 100000, 1 \leq a_i, va \leq 10^6$$

$$1 \leq l \leq r \leq n$$

子任务编号	分值	特殊性质	依赖的子任务
1	7	$n, q \leq 10$	无
2	8	$ai, va \leq 10000, q=0$	无
3	18	$a_i = i, q=0$	无
4	27	$q=0$	2, 3
5	17	$n, q \leq 10000$	1
6	23	无	1, 2, 3, 4, 5

远征

背景

岂曰无衣 与子同裳
王于兴师 修我甲兵 与子偕行

题目描述

这里，萧瑟的秋风带来肃杀的气氛，严整的戈矛寒光逼人，一场腥风血雨的大战和并肩战斗的浴血搏杀即将展开。

王师是最能跋山涉水的军队，是最同仇敌忾的军队。

他们拥有两个大本营，营地的形状都是圆形。由于是远征讨伐，大本营之外的地域都是敌人占领的区域，军队的活动范围是这两个圆内(包括圆上)的点集的并集。

在这次战争中，王师需要完成 q 次作战任务；对于每一次任务，初始时军队的位置可以抽象地认为在点 S ，保证点 S 在军队的活动范围内。根据君王多年的作战经验，每一次行军长度不能太长，那样会造成士卒疲惫而影响战斗力，也不能太短，那样影响行军效率，因此军中规定一次行军必须行进恰好 d 单位长度。并且由于需要保证行军安全，每一次行军的起点和终点必须在军队活动范围内，但军队行动敏捷，中途可以经过敌方占领区。军队最终想要到达 T 点(也保证在军队的活动范围内)的位置来完成作战任务。

现在君王想要知道，每一次作战任务能否通过有限次行军完成，军情紧急，你能帮他快速计算吗？

注意由于战场情况瞬息万变，每次作战任务的营地形状， S 和 T 的位置以及 d 值都会改变；并且注意虽然给出的 S ， T 的坐标均为整数，但行军的起始点坐标值可以不为整数。

输入格式

从文件 `expedition.in` 中读入数据；

第一行一个整数表示数据所属的子任务编号，子任务含义见后文；

第二行一个整数 q 表示作战任务数；

接下来描述了 q 次作战任务的信息，对于每个作战任务有三行数据进行描述：

第一行六个整数 $x_1, y_1, r_1, x_2, y_2, r_2$ 分别描述了两个圆的圆心坐标及半径；

第二行四个整数 S_x, S_y, T_x, T_y 分别表示 S 和 T 的坐标；

第三行一个整数 d 表示每一次的行军长度。

输出格式

输出到文件 `expedition.out` 中；

为了减少输出量，输出时我们需要进行如下处理：

设变量 $ans_i (i \in [1, q])$ 表示第 i 次作战任务的答案(能完成任务值为 1, 否则值为 0);

你只需要输出 $(\sum_{i=1}^q ans_i \cdot 2^{i-1}) \bmod 998244353$ 的值即可。

样例输入 1

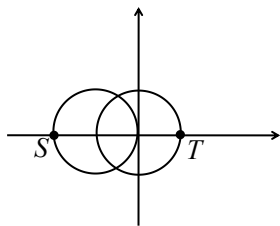
```
6
3
0 0 2 -2 0 2
-4 0 2 0
3
0 0 2 -2 0 2
-4 0 2 0
6
0 0 2 -2 0 2
-4 0 2 0
9
```

样例输出 1

```
3
```

样例解释

三次任务的营地形状和 S, T 都相同, 如图所示:



若 $d=3$, 可行的方案之一为: $(-4,0) \rightarrow (-1,0) \rightarrow (2,0)$;

若 $d=6$, 可行的方案之一为: $(-4,0) \rightarrow (2,0)$;

若 $d=9$, 显然不存在可行方案。

$ans_1=1, ans_2=1, ans_3=0$, 因此答案为 3。

样例输入 2~7

见选手文件下的 expedition2~7.in。

样例输出 2~7

见选手文件下的 expedition2~7.ans。

子任务

对于所有的测试数据，保证：

$$1 \leq q \leq 100000$$

$$-500 \leq x_1, y_1, x_2, y_2 \leq 500$$

$$1 \leq r_1, r_2 \leq 500$$

$$-1000 \leq Sx, Sy, Tx, Ty \leq 1000$$

$$1 \leq d \leq 1000$$

保证点 S ，点 T 都属于两个圆内(包括圆上)的点集的并集。

子任务编号	分值	特殊性质	依赖的子任务
1	20	输入的所有数绝对值不超过 10，数据随机	无
2	23	其中一个圆内(包括圆上)的点集是另一个的子集	无
3	17	两个圆圆内(包括圆上)的点集交集为空	无
4	8	$d=1$	无
5	17	$d > 2r_1, d > 2r_2$	无
6	15	无	1~5

注：“数据随机”的方式为： $x_1, y_1, x_2, y_2, r_1, r_2, d$ 在其范围内等概率随机， S 点和 T 点分别在合法集合内均匀随机。

提示

1. 两个点 $(x_1, y_1), (x_2, y_2)$ 的欧几里得距离计算公式为： $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ ；
2. 在程序实现过程中请尽量避免不必要的精度误差。