

学了就会做的炒鸡简单的题

Orange

2018 年 6 月 7 日

	前缀和	子集的子集	农场
题目类型	传统型	传统型	传统型
源程序文件名	prefix.cpp	subsubset.cpp	farm.cpp
输入文件名	prefix.in	subsubset.in	farm.in
输出文件名	prefix.out	subsubset.out	farm.out
每个测试点时限	1 秒	2 秒	1 秒
内存限制	512 MiB	512 MiB	128 MiB
测试点数目	(子任务)	(子任务)	10
每个测试点分值	(子任务)	(子任务)	10
编译选项	(默认选项)	(默认选项)	(默认选项)
比较答案方式	(默认方式)	(默认方式)	(默认方式)

请仔细阅读本页内容

所有测试均不打开 `-std=c++11` 编译选项

比较答案的默认方式为全文比较，忽略文末回车，但不忽略行末空格。

满分 500 分，但像往常一样，实际上你只需要得到 300 分就够了。

如果被电脑卡常，可申诉。

1 前缀和 `prefix.cpp` / `prefix.in` / `prefix.out`

1.1 题目描述

自从 lyc 学了前缀和后，他就一直在想：前缀和这么厉害，能拿来玩一年吗？

于是他给了你一个看上去要算一年的任务：对序列 A 求 m 次前缀和，求出它的第 n 项（规定下标从 0 开始）。其中，序列 A 为：

$$A = a_0, 0, 0, 0, \dots, 0$$

即第 0 个数为 a_0 ，其余数均为 0。保证 a_0 为正整数。

由于答案可能很大，所以结果对 100003 取余。

1.2 输入

本题包含多组询问。

第一行输入一个整数 T ，表示数据组数。

对于每组数据，输入一行三个整数 n, m, a_0 ，含义均在题目描述中提到。

1.3 输出

对于每组询问，输出包含一行一个正整数表示答案。

1.4 样例

1.4.1 输入

```
2
4 2 3
2 0 1
```

1.4.2 输出

```
15
0
```

1.4.3 解释

对于第一个询问，一开始，序列为：

3, 0, 0, 0, 0

求了第一次前缀和后，序列为：

3, 3, 3, 3, 3

求了第二次前缀和后，序列为：

3, 6, 9, 12, 15

所以输出 15，注意下标从 0 开始。

对于第二个询问，答案显然为 0。

1.5 数据规模与约定

本题采用子任务测试的形式。仅当你通过了子任务中所有的测试点，你才能得到该子任务对应的分数，否则不得分。

对于 100% 的数据， $T \leq 10^4$ ， $0 \leq a < 10^5 + 3$ 。

子任务 1 (24 分)： $n, m \leq 100$ 。

子任务 2 (5 分)： $n = 1$ ， $m \leq 50000$ 。

子任务 3 (30 分)： $n, m \leq 50000$ ， $T = 1$ 。

子任务 4 (41 分)： $n, m \leq 50000$ 。

子任务 5 (100 分)： $n, m \leq 10^{18}$ 。

1.6 提示

注意边界。

2 子集的子集 `subsubset.cpp` / `subsubset.in` / `subsubset.out`

2.1 题目描述

有一个大小为 n 的集合 S ，它由 n 个二元组 (i, x_i) ($1 \leq i \leq n$) 组成。我们用序列 s_i ($1 \leq i \leq n$) 来表示属于 S 的二元组 (i, s_i) 。定义一个集合的权值为属于它的二元组的 x 之积，即二元组的第二个元素之积。求出 S 的所有非空子集的所有非空子集的权值之和。

2.2 输入

输入包含两行。第一行输入一个正整数 n ，第二行输入 n 个正整数，表示 $s_{1 \sim n}$ 。

2.3 输出

输出包含一行一个整数，表示答案。答案对 $998244353 = 119 \times 2^{23} + 1$ 取模。

2.4 样例

2.4.1 输入

2
2 3

2.4.2 输出

16

2.4.3 解释

$S = \{(1, 2), (2, 3)\}$ ，为了方便，我们省略二元组的第一个元素，记 $S = \{2, 3\}$ 。

$\{2, 3\}$ 的非空子集有： $\{2, 3\}$ ， $\{2\}$ ， $\{3\}$ 。

$\{2, 3\}$ 的非空子集的权值之和为： $2 \times 3 + 2 + 3 = 11$ 。

$\{2\}$ 的非空子集的权值之和为：2。

$\{3\}$ 的非空子集的权值之和为：3。

所以答案为 16。

2.5 数据规模与约定

本题采用子任务测试的形式。仅当你通过了子任务中所有的测试点，你才能得到该子任务对应的分数，否则不得分。

对于 100% 的数据，有 $1 \leq s_i < 998244353$ 。

子任务 1 (12 分)： $n \leq 8$ 。

子任务 2 (20 分)： $n \leq 15$ 。

子任务 3 (27 分)： $n \leq 5 \times 10^3$ ， $\forall i, s_i = 1$ 。

子任务 4 (41 分)： $n \leq 5 \times 10^3$ 。

子任务 5 (100 分)： $n \leq 10^5$ 。

3 农场 farm.cpp / farm.in / farm.out

3.1 题目描述

lyc 在 CCF 田野上有一个农场，农场里有 n 块土地，从左到右呈一字形排列，依次编号为 $1 \sim n$ 。为了防止 DB 来偷菜，他要在某些地下安放监控器。每块地里最多只能建立一个监控器。设土地 i 左边离它最近的第一个监控器的位置在 d_i ，如果在 i 安放监控器，那么它要监控的土地为 $d_i + 1 \sim i$ 。特别地，第一个监控器要监控的土地为 $1 \sim i$ 。由于离监控器越远的土地需要的技术含量越高，lyc 对每一块土地的花费等于它到监控它的监控器之间的土地数目（不包括自身，但包括监控器所在的土地）乘上该土地的种植量。

已知在第 i 块土地安放监控器的花费是 a_i ，第 i 块土地的种植量是 b_i 。请你帮 lyc 求出最小总花费。

3.2 输入

输入包含 3 行。

第一行输入一个整数 n ，表示土地数目。

第二行输入 n 个整数，第 i 个整数表示 a_i 。

第三行输入 n 个整数，第 i 个整数表示 b_i 。

3.3 输出

输出包含一行一个整数，表示答案。

3.4 样例

3.4.1 输入

```
4
2 4 2 4
3 1 4 2
```

3.4.2 输出

```
9
```

3.4.3 解释

选取第 1, 3, 4 号土地安放监控器，此时的费用最小，为 $2 + (2 + 1 \times 1) + 4 = 9$ 。

3.5 数据规模与约定

对于 10% 的数据， $n \leq 10$ 。

对于 40% 的数据， $n \leq 10^3$ 。

对于 100% 的数据， $n \leq 10^6$ ， $1 \leq a_i, b_i \leq 10^4$ 。