

CSP-S 模拟赛

2019 年 10 月 14 日 08:00 - 11:30

题目概况

题目名称	CSM	名额限制	模拟赛摸鱼
源程序文件名	csm.c/cpp/pas	limit.c/cpp/pas	game.c/cpp/pas
每个测试点时限	1 秒	1.5 秒	2 秒
测试点数目	捆绑测试	25	捆绑测试
每个测试点分值		4	
运行内存上限	512 MB	1024 MB	512 MB
题目类型	传统题	传统题	传统题
结果比较方式	全文比较（过滤行末空格及文末回车）		

※ 除特殊说明外，输入、输出文件的数字之间均以恰一个空格分隔，行末字符为 Line Feed（'\n'，ASCII 10），且该字符也存在于输入文件的最后一行末尾。

编译命令（以第一题为例）

C	<code>gcc -o csm csm.c -O2 -lm</code>
C++	<code>g++ -o csm csm.cpp -std=c++11 -O2 -lm</code>
Pascal	<code>fpc csm.pas -O2</code>

Problem A. CSM

输入文件: `csm.in`
输出文件: `csm.out`
时间限制: 1 秒
内存限制: 512 MB

在另一个平行宇宙里面, 作为一个 OMer&IOer 的小 U, 成功在 Dress OJ (裙子 OJ) 的比赛中获得前 10 名, 拿到了一款美丽的女装。小 U 由于穿上女装去考试的缘故, 通过了 CSM-S 的初赛。

下面是小 U 顺利解出的一道组合计数题:

将 6 个数 2, 0, 1, 9, 20, 19 按任意顺序排成一行, 拼成一个八位数, 则产生的不同 8 位数的个数为多少个?

由于某种原因, 这道题目穿越到了这次联测里面。不过你不满足于解决这一道题目, 因此你需要解决下列问题:

将 n 个由 0-9 组成的数字串排成一列, 能够组成多少个不同的数字 (注意这里的数字要求不能有前导 0)。

输入格式

第一行包含一个正整数 n , 表示数字串的个数。

接下来 n 行, 每行一个由 0-9 组成的数字串, 第 i 行代表第 i 个数字串。

输出格式

输出一行一个整数, 表示能够组成的不同数字的个数。

样例 1

<code>csm.in</code>	<code>csm.out</code>
2 1 0	1

样例解释 1

只能组成 10 这一种数字。

样例 2

<code>csm.in</code>	<code>csm.out</code>
6 2 0 1 9 20 19	498

样例 3

见选手文件夹中的 `csn3.in/ans`。该样例满足子任务 3 中的限制。

数据范围

记 S 为所有字符串的长度之和。对于所有测试数据, 满足 $n \leq 7, 1 \leq S \leq 30$ 。

子任务 1 (分值: 10)

$$n \leq 2.$$

子任务 2 (分值: 20)

$$n \leq 3.$$

子任务 3 (分值: 70)

无特殊限制。

Problem B. 名额限制

输入文件: `limit.in`
输出文件: `limit.out`
时间限制: 1.5 秒
内存限制: 1024 MB

在另一个平行宇宙里面, SH 始终是中国 IO 竞赛的强省, 有 m 所 IO 强校。而小 U 所在的学校 ZFE, 作为传统的 IO 强校, 每年省选时, 都会被 FCC 的学校名额限制卡。

“你们这十个人都是集训队水平的, 然而很遗憾, 由于 FCC 的名额限制, 你们只能进两个, 所以你们都要做好退役的准备”教练在省选前一周对小组里面的每一个人说。

值得一提的是, 在这个平行宇宙里面, FCC 对学校的名额限制是“ k 分之一”名额限制, 而 SH 省队一共 a 个人。决定省队名单的方法是: 先将选手按成绩从高到低排序, 依次决定每一位选手是否进省队。如果加上这一位选手过后, 名单中的人数不超过 a 人, 并且每个学校在名单中的人数不超过 $\lfloor \frac{a}{k} \rfloor$ 人, 就让这位选手进省队, 否则就不让。

不论是惧怕还是期待, 决定命运的那一天终究要来临。省选一試和二試考完了, 每个人的成绩和最终省队名单公布在大屏幕上。

“全校的 Rk3, 怎么会这样!”小 U 看到自己省选二試的分数, 又叹息道: “要是我这题没写挂, 我就 Rk1 了啊。要是我...”

泪水无法改变小 U 被“ k 分之一”名额限制所卡的事实。在 OM 上没有发挥失误的小 U, 最终在 IO 最关键的一场比赛—省选, 因为一个小失误而无缘省队。

他偷偷穿越到你现在身处的这个宇宙中, 看到这个平行宇宙的自己数竞无缘省队。他又搜寻了无数的平行宇宙, 却始终无法找到一个他所认为的没有缺憾的世界。

至于那个平行宇宙中, 小 U 之后的人生轨迹, 你已经是无从知晓了。但是你发现你这个世界里, 多了一份那次省选的名单, 那是小 U 穿越时无意间留下的痕迹。你发现这个名单上写着 n 名参加省选的人的名字, 并且按照成绩从高到低排序, 没有同分。你能看到哪些人进入的省队, 而哪些人没有进, 但是你不能看到每个人所在的学校。

给定 n, m, k , 以及每个人是否进入了省队。你要求出来, 省队名额总数与进入省队的人数相等的前提下, 每个人所在学校的可能状态数。如果两种状态中, 存在一个人在第一种状态和第二种状态的所在学校不同, 就认为两种状态不同。由于答案很大, 请输出答案模 998244353 意义下的最小非负剩余。当然, 你无法保证名单的正确性, 所以答案可能为 0。

输入格式

第一行包含三个正整数 n, m, k , 分别表示参加省选的人数, 学校的个数, 名额限制的比例。

第二行包含一个长度为 n 的字符串, 保证字符串中只有 0 或 1。其中第 $i(1 \leq i \leq n)$ 位为 0 表示排在第 i 名的人没有进入省队, 否则表示进入了省队。

输出格式

仅一行, 表示每个人所在学校的可能状态数模 998244353 的最小非负剩余。

样例 1

<code>limit.in</code>	<code>limit.out</code>
5 5 4 10111	120

样例解释 1

把学校标号为 1,2,3,4,5, 那么第 1 名和第 2 名必定在一个学校, 而第 1,3,4,5 名选手所在学校必定互不相同, 否则就会有人被”4 分之一” 名额限制所卡。

样例 2

<code>limit.in</code>	<code>limit.out</code>
7 3 2 1111111	1050

样例 3

见选手文件夹中的 `limit3.in/ans`。

样例 4

见选手文件夹中的 `limit4.in/ans`。

样例 5

见选手文件夹中的 `limit5.in/ans`。

数据范围

设 a 表示进入省队的人数个数。对于所有测试数据, 满足 $1 \leq n, m \leq 10^5, 1 \leq k \leq 10^2, k \leq a \leq n$ 。

测试点编号	n	m	k	特殊性质 1	特殊性质 2
1	≤ 15	≤ 3	≤ 15	×	×
2	≤ 7	≤ 7	≤ 7	×	×
3	≤ 100	≤ 2	≤ 100	×	×
4	≤ 100	≤ 2	≤ 100	×	×
5	≤ 100	≤ 3	≤ 100	×	×
6	≤ 1000	≤ 2	≤ 100	×	✓
7	≤ 3000	≤ 2	≤ 100	×	×
8	≤ 100	≤ 100	≤ 100	×	✓
9	≤ 100	≤ 100	≤ 100	×	✓
10	≤ 100	≤ 1000	≤ 100	×	×
11	≤ 100	≤ 1000	≤ 10	×	✓
12	≤ 1000	≤ 5000	≤ 10	×	✓
13	≤ 1000	≤ 5000	≤ 10	×	×
14	≤ 1000	≤ 5000	≤ 100	×	✓
15	≤ 1000	≤ 5000	≤ 100	×	✓
16	≤ 1000	≤ 5000	≤ 100	×	×
17	≤ 1000	≤ 5000	≤ 100	×	×
18	≤ 100000	≤ 100000	≤ 100	✓	×
19	≤ 100000	≤ 100000	≤ 100	✓	×
20	≤ 10000	≤ 10	≤ 20	×	✓
21	≤ 10000	≤ 10	≤ 10	×	×
22	≤ 100000	≤ 100	≤ 10	×	✓
23	≤ 100000	≤ 100000	≤ 10	×	×
24	≤ 100000	≤ 100000	≤ 100	×	✓
25	≤ 100000	≤ 100000	≤ 100	×	×

特殊性质 1: $k = a$ 。

特殊性质 2: 保证选出的人是前 a 名。

Problem C. 模拟赛摸鱼

输入文件: `game.in`
输出文件: `game.out`
时间限制: 2 秒
内存限制: 512 MB

小 Z 是一个很菜的 IOer，全国高中 IO 联赛的难度对他来说太难了。在某次模拟赛中，他确认自己一题都不会后开始了摸鱼。

小 Z 在电脑上发现了一个神奇的小游戏，这个游戏的规则如下：

二维平面上有 n 个点，第 i 个点的坐标为 (x_i, y_i) 。在游戏过程中，你可以选择一个未被破坏的点，把这个点破坏并获得 1 点收益，然后，所有满足 $x_j - x_i > u_i$ 或 $y_j - y_i > v_i$ 的点 j 都会自动被破坏。你可以多次重复以上操作，直到所有点都已经被破坏为止。 u_i 和 v_i 是游戏中给定的常数，保证 $u_i, v_i > 0$ 。

小 Z 想知道，如果采取最优策略，最多能够获得多少点收益。小 Z 的 IO 水平很菜，所以他不会计算这个游戏的最大收益。于是他找到了你，希望你能帮他求出这个游戏可能得到的最大收益。

输入格式

第一行包含一个正整数 n ，表示平面上的点数。

接下来 n 行，每行 4 个正整数 x_i, y_i, u_i, v_i ，意义如题目描述中所述。

输出格式

输出一行一个整数，表示最多能够获得多少点收益。

样例 1

game.in	game.out
10 1 7 4 7 20 4 9 3 13 19 8 3 4 5 7 5 10 19 3 3 17 4 10 6 5 16 5 6 7 6 1 9 7 18 4 4 18 11 5 2	7

样例解释 1

一种最优方案是依次破坏点 3, 5, 7, 9, 4, 8, 1。

样例 2

game.in	game.out
20	13
21 34 32 29	
55 47 47 18	
68 82 10 19	
52 17 28 47	
12 12 42 20	
91 92 9 9	
12 18 34 42	
21 13 45 30	
28 76 3 12	
55 49 32 15	
13 89 2 33	
69 24 19 24	
37 14 48 28	
49 54 12 50	
17 68 1 39	
26 77 36 50	
86 5 27 38	
98 40 45 16	
22 59 32 35	
15 54 10 13	

样例解释 2

一种最优方案是依次破坏点 6, 3, 14, 16, 2, 4, 10, 1, 7, 19, 5, 8, 13。

样例 3

见选手文件夹中的 game3.in/ans。该样例满足子任务 3 中的限制。

样例 4

见选手文件夹中的 game4.in/ans。该样例满足子任务 4 中的限制。

数据范围

对于所有测试数据, 满足 $n \leq 5 \cdot 10^5, 1 \leq x_i, y_i, u_i, v_i \leq 10^9$ 。

子任务 1 (分值: 10)

$n \leq 11$.

子任务 2 (分值: 20)

$n \leq 80$.

子任务 3 (分值: 20)

$$n \leq 400.$$

子任务 4 (分值: 20)

$$n \leq 1000.$$

子任务 5 (分值: 30)

$$n \leq 5 \cdot 10^5.$$

提示

本题读入量较大。建议使用较快的读入方式。