

2020~2021 年高中信息学多校联合训练

不知道什么比赛

模拟赛

时间：2020 年 1 月 30 日 8:00 ~ 13:00

题目名称	去南极	登山	露营
题目类型	传统型	传统型	提交答案型
目录	space	climb	camp
可执行文件名	space	climb	N/A
输入文件名	space.in	climb.in	camp1~10.in
输出文件名	space.out	climb.out	camp1~10.out
每个测试点时限	2.0 秒	2.0 秒	N/A
内存限制	256 MB	1024 MB	N/A
子任务数目	6	7	10
测试点是否等分	否	否	否

提交文件名

对于 C++ 语言	space.cpp	climb.cpp	camp1~10.out
-----------	-----------	-----------	--------------

编译选项

对于 C++ 语言	-lm -O2 -std=c++11	N/A
-----------	--------------------	-----

注意事项：

1. 测评时栈空间与内存限制相同。
2. 时间限制保证在标程的两倍以上，具体时限可随实际测评环境调整。
3. 若无特殊说明，输入文件的同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
4. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
5. 测评时在每个题目对应目录下收取答案，请对每个题目建立子文件夹。
6. 难度与题目顺序无关，请自行选择写题顺序。
7. 题目比较简单，请独立完成。请不要借助网络等工具。

去南极(space)

【题目背景】

小 H 有一个计划，那就是去南极旅游。她找到了几个志同道合的朋友，准备利用寒假的时间去南极。

结果她们还真的成功了……

当然，这项活动和日常活动不一样，在这之前需要经过一系列培训。因此，她们来到了山里进行实践培训。

【题目描述】

培训的一个项目是立旗杆来确定方向。假设她们需要确定一条长度为 n 的路线，这条路线可以被表示成 $0, 1, \dots, n, n+1$ 这 $n+2$ 个格子。首先，她们需要在 $0, n+1$ 分别立两根旗子，然后在中间的 n 格中立若干根旗子。最后她们需要保证任意两根相邻的旗子之间的距离（不算旗子所占用的格子）不超过 m 。

注意：旗子是立在某个格子的正中央，并非两格之间。

显然最优解是在 $m+1, 2m+2, \dots$ 这些位置设立旗子，但是这样太死板了，小 H 想要把旗子设立得更有趣。小 H 会采取以下算法来设立旗子：

- ① 找到一段区间 $[l, r]$ 满足 $[l, r]$ 中都没有旗子， $l-1, r+1$ 都有旗子并且 $r-l+1 > m$ 。小 H 认为这样的区间是“极大”的。
- ② 小 H 在区间中找到一个位置 x ，显然 $l \leq x \leq r$ ，然后在 x 上插上旗子。
- ③ 重复步骤一，直到找不到合法的极大区间

在设计完这个算法后，小 H 发现了个严重的问题。如果每一次 x 都是随机找的，那么这个算法有可能会让这个活动花费远多出最优解的体力!!!

因此，小 H 决定优化找 x 这个过程。具体的，小 H 会在区间 $[l, r]$ 中**等概率随机地**找出三个不同的整数 a, b, c ，然后排序使得 $a < b < c$ ，最后她令 $x = b$ 。这样一来， x 会更接近区间中点，这个算法就能更快结束。

不过，这个过程还是要花费体力的。对于一段长度为 len 的区间（即 $r-l+1 = len$ ），步骤二需要花费 s_{len} 的体力。

小 H 想知道她们在确定旗子设立点时需要花费的体力的期望值。由于现在还没开始培训，这个算法还在计划阶段，所以小 H 不知道具体的 n, m, s_i 是多少，因此她会进行一些修改和多次询问。

【输入格式】

从文件 `space.in` 中读入数据。

第一行读入两个整数 L, Q ，其中 L 表示所有询问操作中 n 可能出现的最大值， Q 表示修改操作和询问操作的次数总和。

第二行有 L 个整数，第 i 个整数表示 s_i ，这 L 个数表示初始时的序列 s 。

接下来 Q 行，每行 3 ~ 4 个整数，其中第一个数为 op 。

- 若 $op = 0$ ，则表示该操作为修改操作。接下来会输入三个整数 l, r, x ，表示将 $s_l \sim s_r$ 全部加上 x 。
- 若 $op = 1$ ，则表示该操作为询问操作。接下来会输入两个整数 n, m 。该操作为询问在给定的 n, m 和当前的 s 下期望花费的体力值。

【输出格式】

输出到文件 `space.out` 中。

对于每一个询问操作，输出体力值的期望。为了避免精度误差，你只需要输出这个期望值对 998244353 取模的结果。

【样例 1 输入输出】

space.in	space.out
6 4	798595490
1 2 3 4 5 6	9
1 6 3	6
0 4 6 1	
1 6 3	
1 5 4	

【样例 1 解释】

三问的答案分别为 $\frac{38}{5}, 9, 6$ 。

对于第三问， $n = 5, m = 4, s = \{1, 2, 3, 5, 6\}$ 。

此时有且只有一个极大区间 $[1, 5]$ ，而此时无论怎么选择，只可能在 2, 3, 4 插旗子，之后就不存在长度大于 m 的极大区间了。

那么只有一次插旗子操作，体力为 $s_5 = 6$ 。

【样例 2 输入输出】

见下发文件 `space/space2.in` 和 `space/space2.ans`。

【数据范围和提示】

本题采用捆绑测试。

对于所有数据，满足：

- $3 \leq m < n \leq L \leq 2 \times 10^5$
- $1 \leq Q \leq 2 \times 10^5$
- $\forall i \in [1, L], 0 \leq s_i < 998244353$
- $1 \leq l \leq r \leq L, 0 \leq x < 998244353, op \in \{0, 1\}$

子任务见下表：

子任务编号	L	Q	特殊性质	分值
1	≤ 100	≤ 100	无	5
2	≤ 5000	≤ 5000		10
3		$\leq 2 \times 10^5$	特殊性质 A	10
4	20			
5	特殊性质 B		5	
6			无	50

特殊性质 A：所有询问操作的 n 都相同。

特殊性质 B：所有询问操作的 m 都相同，并且所有修改操作都满足 $l = 1, r = L$ 。

登山(climb)

【题目描述】

小 H 是一名登山爱好者，她正在计划一次登山活动，并邀请她的朋友一起登山。

小 H 查找了很多山，有些山的攀登难度高，有的山攀登难度低，有点山适合挑战，有的山适合散步。针对不同的山，小 H 制定了不同的计划。

对于攀登难度高的山：小 H 观察到，这些山的登山口一般不在海拔为 0 的地方，而是在有一定高度的地方。

对于此类山，小 H 会先计划一个登山时间 n ，接下来的 n 段时间内，每一段时间都会随机向上走一单位或向下走一单位。如果记初始位置为高度 0，那么每一段时间内小 H 她们所在高度都会随机 $+1$ 或 -1 。

因为高度 0 不代表海拔为 0，所以小 H 她们所在高度可以为负数。

在 n 段时间中，会产生 $n + 1$ 个时间点，小 H 会记入下这 $n + 1$ 个时间点所在的高度，其中最高的高度就是她们本次登山所在的最高高度。

因为攀登此类山峰需要有一定目标，所以小 H 还会给定一个目标 k ，表示她们这次登山的最高高度至少为 k 。

对于攀登难度低的山：小 H 观察到，这些山一般和市区距离不会太远，所以登山口一般都在海拔为 0 的地方。

同样的，对于此类山，小 H 也会规定一个登山时间 n ，然后再接下来的 n 段时间随机的向上或向下走一单位。但是此时高度不允许为负（因为初始海拔为 0）并且终点的高度必须为 0（散步完直接回家）。

当然，因为是散步，所以小 H 对最高高度没有要求，不过她仍然会记入下最高高度。

作为小 H 的朋友之一，你注意到对于时间 n 可以产生许多合法方案，合法方案需要满足这条路可行（即在攀登难度低的山时高度不为负）以及满足要求（即在攀登难度高的山时最高高度大于等于 k 和在攀登难度低的山时终点高度为 0）。你想知道，对于所有合法方案，攀登的最高高度的期望值是多少？如果没有合法方案，则输出 0。

然而，小 H 的计划还不完善，所以可能有多个问题。同时，讨厌精度运算的你决定只求出期望值对 998244353 取模的结果。

【输入格式】

从文件 `climb.in` 中读入数据。

第一行两个整数 $T, type$ ， T 为询问数目， $type$ 为询问类型。若 $type = 0$ ，表示小 H 决定攀登难度高的山；若 $type = 1$ ，表示小 H 决定攀登难度低的山。

接下来 T 行，每行一或两个整数。若 $type = 0$ ，则有两个整数 n, k ；若 $type = 1$ ，则有一个整数 n 。

【输出格式】

输出到文件 `climb.out` 中。

对于每一个询问，输出一个整数，表示最高高度的期望值对 998244353 取模后的结果。

【样例 1 输入输出】

climb.in	climb.out
5 0	499122179
3 2	523872044
20 6	39578760
100 0	802116506
101 0	98038061
2021 130	

【样例 1 解释】

该样例为攀登难度高的山峰。

对于第一组询问，总共有 $\{+1, +1, +1\}$ 和 $\{+1, +1, -1\}$ 两种攀登方式。其中第一种攀登方式的最高高度为 3，第二种攀登方式的最高高度为 2，期望值为 $\frac{3+2}{2} = \frac{5}{2}$ ，对 998244353 取模后为 499122179。

【样例 2 输入输出】

climb.in	climb.out
5 1	0
3	499122178
4	0
127	915680881
66	765781763
1234	

【样例 2 解释】

该样例为攀登难度低的山峰。

对于第一组询问，没有一种攀登方式最后可以回到高度 0。

对于第二组询问，合法的攀登方式为 $\{+1, -1, +1, -1\}$ 和 $\{+1, +1, -1, -1\}$ ，第一种攀登方式的最高高度为 1，第二种攀登方式的最高高度为 2，期望值为 $\frac{1+2}{2} = \frac{3}{2}$ ，对 998244353 取模后为 499122178。

【数据范围和提示】

本题采用捆绑测试。
对于所有数据，满足 $1 \leq T \leq 10^5, type \in \{0,1\}, 0 \leq k \leq n \leq 2 \times 10^7, n \geq 1$ 。
子任务见下表：

子任务编号	T	n	k	$type$	分值
1	≤ 3	$\leq 10^3$	$\leq n$	$= 0$	5
2		$\leq 2 \times 10^5$			10
3	$= 0$		10		
4	$\leq n$		30		
5	≤ 2	$\leq 10^2$	$-$	$= 1$	5
6		$\leq 10^5$			10
7		$\leq 2 \times 10^7$			30

露营(camp)

【题目描述】

本题是一道提交答案题。

小 H 计划和朋友进行一次野外露营。

野外露营可以做很多事情，比如搭帐篷、野炊等。然而，最重要是事情是朋友们在一起闲聊。

不过，闲聊实在太浪费时间了，小 H 认为这段时间可以做一些有意义的事。众所周知，小 H 和她的朋友们都是数学爱好者，所以在这段时间里，她们决定讨论数学题。因此，小 H 计划每人准备一道有趣数学题，在露营时分享给大家。

终于到了露营的那天了，小 H 也拿出了她准备的数学题：

给定一张 n 个点的完全图，你现在需要使用 m 种颜色给每条边染色，但是不允许出现同色三元环，求方案。

你决定挑战这道题。

【实现细节】

在选手目录下有 12 个文件，分别为 `camp0~10.in` 和 `camp0.out`。

其中 `camp0~10.in` 为输入文件，并且 `camp0.in` 和 `camp0.out` 为样例。

在 `camp*.in` 中，总共有一行，表示两个数 n, m ，即为图的点数和你可以使用的颜色数。

你不需要提交源代码，只需要提交文件 `camp1~10.out`。

对于 `camp*.out`，记与其对应的 `camp*.in` 中的两个数为 n, m ，则在 `camp*.out` 中，一共需要输出 n 行，每行 n 个整数。第 i 行第 j 个整数 $A_{i,j}$ 表示边 (i, j) 的颜色。

你输出的矩阵 A 需要满足以下性质：

- 对于 $\forall i, j$ 且 $i \neq j$ ，都有 $1 \leq A_{i,j} \leq m$ 且 $A_{i,j}$ 为整数。
- 对于 $\forall i, j$ ，都有 $A_{i,j} = A_{j,i}$ 。
- 对于 $\forall i$ ，都有 $A_{i,i} = 0$ 。
- 对于 $\forall i, j, k, i < j < k$ ，都有 $A_{i,j}, A_{i,k}, A_{j,k}$ 不完全相同。

【检查工具】

在选手目录下，还有一个名为 `checker.cpp` 的文件，对该文件使用如下编译命令：

```
g++ -o checker checker.cpp -lm -O2 -std=c++11
```

之后你会得到一个名为 `checker` 的可执行文件。该可执行文件可以帮助你检验答案的正确性。

请将输入文件保存在 `camp.in` 中，将输出文件保存在 `camp.out` 中，运行可执行文件，它将判断你的答案。

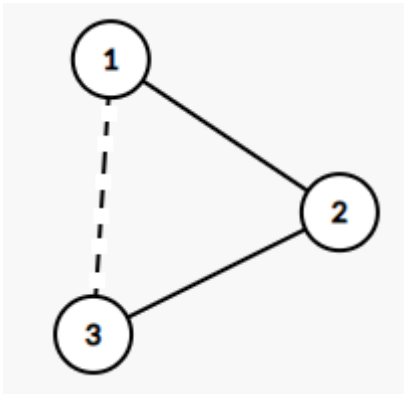
如果可执行文件返回的为 OK! ，那么代表你通过了 camp.in 中要求的构造，否则它会返回错误信息。

【样例输入输出】

camp0.in	camp0.out
3 2	0 1 2 1 0 1 2 1 0

【样例解释】

方案如下图：



【数据范围和提示】

如果你构造的方案不合法，那么该测试点得 0 分。
下表可以帮助你快速得到每个测试的信息：

测试点编号	1	2	3	4	5	6	7	8	9	10
$n =$	5	8	16	25	34	44	50	85	101	155
$m =$	2	3	3	4	4	4	4	5	5	5
分值	5	5	10	10	10	10	15	10	10	15