

2020 年牛客 NOIP 赛前集训营（第六场）

提高级

比赛地址：<https://ac.nowcoder.com/acm/contest/7615>

题目名称	袜子分配	艰难睡眠	路径难题	牛半仙的妹子序列
题目类型	传统型	传统型	传统型	传统型
每个测试点 时限	C/C++ 1 秒, 其他语言 2 秒	C/C++ 1 秒, 其他语言 2 秒	C/C++ 1 秒, 其他语言 2 秒	C/C++ 1 秒, 其他语言 2 秒
内存限制	C/C++ 256MB, 其他语言 512MB	C/C++ 256MB, 其他语言 512MB	C/C++ 256MB, 其他语言 512MB	C/C++ 256MB, 其他语言 512MB
子任务数目	20	10	20	10
测试点是否 等分	是	是	是	是

注意事项

- 所有参与 NOIP 赛前集训营的选手必须遵守约定的纪律：
 - 比赛账号不能外传。
 - 比赛中不能抄袭代码。
 - 比赛中不能恶意卡评测。
 - 报名支付账号即为比赛账号。
 - 一旦报名 NOIP 赛前集训营活动，不支持退费，请考虑清楚后报名。
 - 本活动解释权归牛客网所有，活动介绍未尽事宜以牛客网官方解释为准。
- 欢迎关注“比赛自动姬”公众号，关注更多比赛资讯~



袜子分配

【题目描述】

牛牛洗袜子时，会将 n 双袜子同时扔进洗衣机，每两双袜子的颜色均不相同。

洗完后，可以认为袜子都被随机打乱了。牛牛每次随机地从洗好的袜子堆中取出两只并晾晒。如果某次取到的袜子是同色的袜子，牛牛就会开心一次。

牛牛想知道这样取 n 次的过程中，开心次数的期望是多少。

数学期望是指所有可能结果的平均值，详见样例。

【输入格式】

一行一个整数，表示扔进洗衣机的有 n 双袜子。

对于30%的数据有 $n \leq 10$ 。

对于40%的数据有 $n \leq 20$ 。

对于80%的数据有 $n \leq 2000$ 。

对于100%的数据有 $n \leq 100000$ 。

【输出格式】

输出一行一个小数，你的答案被认为是正确的当且仅当绝对或相对误差小于 10^{-8} 。

【样例 1 输入】

2

【样例 1 输出】

0.6666666667

【样例 1 解释】

将袜子编号为 $\{1,2,3,4\}$ ，其中 $\{1,2\}$ 为一对， $\{3,4\}$ 为一对。

两次分别抽取共有以下六种可能：

$\{1,2\},\{3,4\}$ 开心次数为 2 次

$\{1,3\},\{2,4\}$ 开心次数为 0 次

$\{1,4\},\{2,3\}$ 开心次数为 0 次

$\{2,3\},\{1,4\}$ 开心次数为 0 次

$\{2,4\},\{1,3\}$ 开心次数为 0 次

$\{3,4\},\{1,2\}$ 开心次数为 2 次

共计为 4/6 次。

艰难睡眠

【题目描述】

牛牛最近睡眠很艰难。

牛牛的宿舍楼有 n 个会吵闹的人，每个人都按照固定的吵闹时间吵闹，吵闹以外的时间不会影响到牛牛睡眠。

每天有 m 分钟，从 1 到 m 标号。第 i 个人从第 a_i 分钟开始吵闹，吵闹时长为 b_i 分钟，即在 $[a_i, a_i + b_i - 1]$ 这段时间里保持吵闹。当然，可能会吵到第二天。

牛牛是个不能忍受吵闹的人，他决定花费一定的代价解决每一个人。

如果希望第 i 个人从第 j 分钟开始吵闹，牛牛需要花费代价 $c_{\{i,j\}}$ 。显然， $c_{\{i,a_i\}} = 0$ 。

牛牛每天至少需要睡眠 k 分钟，牛牛是那种睡下去，就一定要睡足 k 分钟的人，牛牛想知道自己花费的最小代价是多少。

当然，牛牛可以从一天的晚上睡到第二天的早上，但是得保证以后的每一天按同样的方式睡眠都不会被吵到。

如果想得到最后 20 分，可能需要足够快的快速读入，这里提供一个 `fread` 的范本：

```
namespace io {
const int L = (1 << 21) + 1;
char ibuf[L], *iS, *iT, obuf[L], *oS = obuf, *oT = obuf + L - 1, c, st[55];
int f, tp;
#define gc() (iS==iT?(iT=(iS=ibuf)+fread(ibuf,1,L,stdin),(iS==iT?EOF:*iS++))*iS++)
inline void gi(int& x) { //get
    for (f = 1, c = gc(); c < '0' || c > '9'; c = gc())
        if (c == '-') f = -1;
    for (x = 0; c <= '9' && c >= '0'; c = gc()) x = x * 10 + (c & 15);
    x *= f;
}
}; // namespace io
using io::gi;
```

使用方法 `gi(n)`; 读入整数 n

注意, `fread` 需要从文件读入, 不影响判题但是可能影响测试。

【输入格式】

第一行输入三个整数 n, m, k , 表示吵闹的人的数目, 一天有几分钟和牛牛至少需要的睡眠时间。

随后 n 行, 第 i 行两个整数 a_i, b_i , 即第 i 个人的开始吵闹时间和吵闹时长。

随后 n 行, 每行 m 个整数, 其中第 i 行第 j 个表示牛牛让第 i 个人从每天的第 j 分钟开始吵闹需要花费的代价 $c_{i,j}$ 。

对于20%的数据有 $n \leq 10, m \leq 10$ 。

对于40%的数据有 $n \leq 100, m \leq 100$ 。

对于80%的数据有 $n \leq 1000, m \leq 1440$ 。

对于100%的数据有 $2 \leq n \leq 5000, 1 \leq a_i, b_i, k \leq m \leq 2000, 0 \leq c_{i,j} \leq 10^4$ 。

【输出格式】

一行一个整数表示答案。

如果无论如何, 牛牛都无法睡足 k 分钟, 输出 -1。

【样例 1 输入】

3 6 3

1 1

3 2

4 3

0 1 2 7 1 2

1 2 0 3 4 5

7 9 5 0 4 6

【样例 1 输出】

4

【样例 1 解释】

让 1 去时间 5 吵闹，吵闹时段为[5,5]。

让 2 去时间 4 吵闹，吵闹时段为[4,5]。

不改变 3 的吵闹时间，吵闹时段为[4,6]。

此时，牛牛可以在[1,3]时段睡觉。

路径难题

【题目描述】

牛牛最近在为自己的出行烦恼。

牛牛所在的城市可以简化为 n 个点，编号从 1 开始。由 m 条无向边表达两两之间的路径。第 i 条边连通点 u_i 和 v_i ，距离为 d_i 。

牛牛是个懒人，他宁愿叫出租车也不愿意走路。牛牛所在城市的出租车不计算起步价，直接按照距离计费，每 r 单位距离 1 元，向上取整，即如果单次行进了 p 单位距离，将收费 $\lceil \frac{p}{r} \rceil$ 元。如果多次乘坐分开收费。

牛牛所在的城市还有 k 路不同的公交车，每一路公交车都是双向的，有 t_i 个站台，可以正着坐，也可以反着坐，但每路公交车仅会在规定的站台停车。牛牛从任意一个位置上车，可以从该路公交车的任意站台下车，仅需付一份该路的公交车车费 c_i 元。但坐到终点站之后必须下车。

牛牛有 q 个规划，其中第 i 个规划为从 a_i 位置移动到 b_i 位置，牛牛想知道这样的移动最小需要的花费是多少。花费是指出租车的花费和公交车的花费之和。

【输入格式】

第一行输入五个整数 n, m, k, r, q ，分别表示牛牛所在的城市点数，边数，公交车有多少路，出租车收费标准和规划次数。

第二行起 m 行，第 i 行三个整数 u_i, v_i, d_i ，表示对应的边连接 u_i 和 v_i ，距离为 d_i 。

随后 k 行，每行第一个整数 t_i 表示对应路公交车的站台数目。第二个整数 c_i 表示该路公交车的收费。随后 t_i 个整数，分别表示该路公交车停的站台。

随后 q 行，第 i 行两个整数 a_i, b_i 表示牛牛计划从 a_i 到 b_i ，想知道其最小花费。

【输出格式】

输出 q 行，每行一个整数 ans_i ，表示第 i 个规划的答案。

【样例 1 输入】

5 6 1 10 5

1 2 15

2 3 177

3 5 73

4 2 37

1 5 66

1 4 43

3 11 1 3 5

1 2

1 4

1 3

4 5

2 3

【样例 1 输出】

2

5

11

11

13

【样例 1 解释】

1 到 2 直接打车，花费 $15/10$ 向上取整，花费为 2。

1 到 4 直接打车，花费 $43/10$ 向上取整，花费为 5。

1 到 3 直接坐公交车，花费为 11。

4 到 5 直接打车，从 4 到 1 到 5，花费 $109/10$ 向上取整，花费为 11。

2 到 3 先打车到 1，花费 2，再坐公交车到 3，花费 11，共计花费 13。

【数据范围】

数据组数	n	m	k
1	10	30	0
2	10	30	0
3	20	1000	0
4	20	1000	0
5	100	1000	2
6	100	1000	2
7	1000	1500	3
8	1000	2000	4
9	2000	5000	≤ 500
10	2000	5000	≤ 500
11	10000	≤ 20000	≤ 100
12	10000	≤ 20000	≤ 100
13	30000	≤ 100000	≤ 1000
14	30000	≤ 100000	≤ 1000
15	60000	≤ 100000	≤ 10000
16	60000	≤ 200000	≤ 10000
17	≤ 200000	≤ 200000	≤ 10000
18	≤ 200000	≤ 200000	≤ 10000
19	≤ 200000	≤ 200000	≤ 10000
20	≤ 200000	≤ 200000	≤ 10000

对于 100% 的数据有 $2 \leq n \leq m + 1 \leq 2 \cdot 10^5, 0 \leq k \leq 10000, 1 \leq q \leq 10, 2 \leq t_i \leq$

$n, \sum t_i \leq 400000, 1 \leq a_i, b_i, u_i, v_i \leq n, 1 \leq d_i, r \leq 10^6, 1 \leq c_i \leq 10^9$

数据保证 n 个点 m 条边的图是一张连通图。但数据并不保证没有重边和自环。

牛半仙的妹子序列

【题目描述】

牛半仙有 n 个妹子，魅力值分别为 $1 \sim n$ ，排成一排。

牛半仙会在这些妹子中选若干个，但是他很贪婪，他只会选完美妹子序列。

一个妹子序列 $\{p_1, p_2, p_3 \dots p_m\}$ (p_i 指妹子的位置) 是完美的，当且仅当其是一个上升序列，且不存在一个 j ，使得 $j > p_m$ 且 $v_j > v_{p_m}$ ，且不存在一个 j 使得 $j < p_1$ 且 $v_j < v_{p_1}$ ，且不存在一个 j 使得 $p_i < j < p_{i+1}$ ， $v_{p_i} < v_j < v_{p_{i+1}}$ 。

牛半仙想知道他有多少种选出完美妹子序列的方式。

因为牛半仙忙着和妹子玩耍，所以他想你帮他解决一下问题。

方法数可能很多，牛半仙只想知道对 998244353 取模的结果。

【输入格式】

第一行一个数 n 。

接下来一行用空格隔开的 n 个数，第 i 个数表示位置在 i 的妹子的魅力值。

【输出格式】

一个数表示方法数对 998244353 取模的结果。

【样例 1 输入】

6

3 5 6 2 1 4

【样例 1 输出】

4

【数据范围】

对于 20% 的数据 $n \leq 300$ 。

对于40%的数据 $n \leq 5000$ 。

对于100% 的数据 $n \leq 200000$ 。

数据有一定梯度。