

Life (alive.cpp) 2s 1024MB

你可以像疯狗一样对周围的一切愤愤不平，你可以诅咒命运，但是等到最后一刻，你还是要平静地放手而去。

——《本杰明·巴顿奇事》

Background

这道题是一道NOI难度的题。

本杰明是一个返老还童人，大体来说就是他从99岁活到了1岁，从而在他最苍老的年华邂逅了最珍视的人。虽然已经垂垂老矣，但有着一张十分青春的面孔的他，回顾自己的人生，总是一个一个人从他的生命中出现，又消失——生命的江河仿佛永远这样浩荡地流过每个人的人生，让人扼腕叹息的同时，又让人轻快地唱出告别的挽歌。长歌当哭，却又止讷于言……

Description

本杰明决定用生命的最后一点清醒时刻，整理自己的日记。他发现自己一共有 n 篇不同的日记，并希望知道以下两个问题的答案：

- 本杰明在回忆过去的时候，喜欢将一些日记摆成圆圈欣赏。他想知道将日记们排成 k 个非空圆圈的方案数（如果圆圈 A 通过旋转可以变成圆圈 B ，那么就是相同的方案，且圈与圈之间没有顺序可言）
- 他还想给日记们分个类并装订成日记本，所以他想知道，将日记们划分成 k 个非空日记本的方案数。

注意，这两个问题本质是不同的。

本杰明自己可不傻！他也是会计算的，不过仅限于 n 和 k 很小的时候：比如 $n = 6, k = 4$ 时，第一种整理日记的、本质不同的方法数量为85，第二种整理日记的、本质不同的方法数量为65。但是好像当 n 和 k 比较大的时候，他就无能为力了。

但由于本杰明现在时常神志不清，所以他会向你询问 q 次，每一次会给你一个问题 $mark$ （ A 表示问题1， B 表示问题2），一个 N 表示不同的日记篇数，一个 K 即每个问题中的 k 。

为了保证这是个在线可做题，题目一开始会给出一个 Max 。若记上一次的答案为 ans_{i-1} ，当 $i > 1$ 时，当前输入的数据应该经过以下处理：

$$N_i = N_i^{ans_{i-1}+1} \bmod Max + 1$$

$$K_i = K_i \bmod N_i + 1,$$

其中输入的 Max 意义是 $1 \leq K_i \leq N_i \leq Max$

最终所有的方案数对993244853取模。

Sample In

```
7 5
A 4 4
B 6 6
A 3 2
B 6 5
A 7 5
```

第一行两个数， Max 和 q

第2 ~ $q + 1$ 行，每行一个字母，两个数字，即 $mark_i$ 未经处理的 N_i 、 K_i ， $mark$ 的作用详见题目描述，并保证对于所有的 i ，有 $1 \leq K_i \leq N_i \leq Max$ 。

Sample Out

```
3
1
1
1
1
```

共 q 行，每行一个数表示 ans 。

Hint

对于5% 的数据，有 $Max = 1$

对于20%的数据，有 $1 \leq Max \leq 10$

对于40%的数据，有 $1 \leq Max \leq 1000, 1 \leq q \leq 1000$

对于另外10%的数据，有 $1 \leq Max \leq 5000, 1 \leq q \leq 5$

对于全部100%的数据，有 $1 \leq Max \leq 7000, 1 \leq q \leq 3 \times 10^5$

Ps0 :如果结果太大，请对993244853取模

Ps1 :如果你写的是正解，请注意空间哦~

Love_(love.cpp) 2s 128MB

天空是连着的，如果我们也能各自发光的话，无论距离有多远，都能看到彼此努力的身影。

——《龙与虎》

Background

这是一道普及难度的题。

高须龙儿，一个看似很凶恶（遗传了父亲的凶恶面相）、实际很细心很温柔的人，遇到了因为家庭原因变的乖戾暴躁的逢坂大河，一“龙”一“虎（*Taiga*）”之间产生了微妙的化学反应，并最终收获了爱情。他们之间，虽然是历经磨难，但是天空永远是连着的啊。平凡的人与平凡的人相识相知，共同走到平凡的生活里，纵然不能时时刻刻相伴相携，但仍会心有灵犀，仍会看到彼此在蔚蓝的天空上发着平凡却又真挚的光。

他们并没有什么“泰坦尼克”式的轰轰烈烈，也没有由三流作家写成、被包装成一流电视剧的三流故事里面“前世今生”式的缠绵悱恻。就只是平凡的生活，平凡地爱着彼此，知足上进。所谓“看遍春花秋月”者，惟是尔尔。

Description

虽然高须龙儿和逢坂大河已经在一起了，但是大河要去找她的家人，休学了，连龙儿也不知道她去哪了。但是他还是希望能与她互相收发短信，也算是看到对方正在“发光”的一种途径吧。

龙儿似乎发现日本岛似乎不存在有些地方没有信号这种GG的情况。整个日本岛可以看做由一堆 N_i 个节点的不相交的树的集合，对于每一棵树，我们称为一个**联邦**（*A union*）。很显然，每个节点都只有**唯一的一条路径**通往该联邦的其他**所有节点**，且每个联邦 i 都由 $N_i - 1$ 条**道路**（*edge*）和 N_i 座**城市**（*node*）组成。

同时，日本的通讯公司通过信号塔为不同的联邦提供了两种不同的信息交流服务。一种是对点辐射服务：每座 A 信号塔有一个相同的参数 R ，表示**可以向周围与自己相距不超过 R 条道路的城市辐射信号**。第二种是对路辐射服务：每座 B 信号塔有一个相同的参数 R ，表示**可以向周围与自己相距小于 R 座城市的道路辐射信号**。

但每个联邦出于实际需要，已经选好了两种服务中的其中一种（也就是说对于同一个联邦，只存在一种信号塔）。现在，龙儿想知道，对于不同的联邦，为了实现该城市选择的全覆盖方式，最少需要多少座信号塔呢？注意，无论何时，信号塔都必须建在城市里而不是道路上。

Sample In

Example 1

```
1
B 7 1
1 2
1 3
3 4
3 5
4 7
4 6
```

Example 2

```
1
A 6 2
2 3
4 5
5 6
1 2
3 4
```

输入数据第一行一个 T 表示联邦个数，之后输入分为 T 个部分，对于每部分：

第一行一个 $Mark$ ， $Mark = A$ 说明这个联邦使用的服务类型为对点辐射服务， $Mark = B$ 则说明是对路辐射服务；紧跟着的一个 N 表示该联邦的城市个数，一个 R 表示辐射半径。

接下来 $N - 1$ 行，每行一组 u, v 表示城市 u, v 之间有一条道路相连，所有城市编号 $\in [1, N]$ 。

Sample Out

Example 1

```
3
```

Example 2

```
2
```

一个数 Ans ，表示最少需要的信号塔个数。

Hint

对于样例一，我们只需要在1 4 5三座城市建立辐射半径为1的对路辐射信号塔即可。

对于样例二，我们只需要在1 4两座城市建立辐射半径为2的对点辐射信号站即可。

对于20%的数据，满足 $1 \leq N \leq 50, 1 \leq R \leq 3$

存在20%的数据，只有 B 类联邦或 A 类联邦；

同时存在另外10%的数据，保证树是一条链；

对于50%的数据，满足 $1 \leq N \leq 5 \times 10^4, 1 \leq R \leq 5$

对于100%的数据，满足 $1 \leq N \leq 10^6, 1 \leq R \leq 10, 1 \leq T \leq 5$

Legendary

Legendary.cpp 16MB–256MB 0.5s–2s

追逐影子的人，自己就是影子。——《荷马史诗》

General Background

这次NOIp的分数其实并不那么重要。重要的是你是否可以从中学到什么东西，重要的是你是否记得住这段为了拼搏而拼搏的岁月，重要的是你是否有决心去缔造一段传奇。

是啊，这只会是漫长人生微不足道的一粒尘埃。但是我希望有人可以记住——在庸碌的成人都市生活中，闲暇之余可以回想起这段岁月。纵然它看起来好像不是那么的完美，但你仍可以温柔的目光投向自己的曾经。

只要不辜负自己就好。

请相信，LCEZ 55机组一定可以缔造一个传奇。

General Description

本题一共4个 *subtask*，满分200分。

确切来讲，本题只需要1个 *cpp* 文件，在输入文件的开头会有一个大写字母 *A – D* 表示第1 – 4个 *subtask*。

保证数据不会出现同一个测试点有 ≥ 2 种的不同的 *subtask*，即测试点之间互不影响

以下样例均省略该大写字母。

捆绑测试。

其实四合一的原因，在于每道题都挺简单，大概和**未来程序**那道提交答案题差不多，给出不同的选择，总可以让各位选手得到分。

Background1

听说 *pks* 发明了一种叫做 *stack tree* 的东西，好像很厉害，大概 *stack tree* 支持以下操作：

- 合并森林、动态维护森林方差&平均值。
- 动态维护添加 *k* 条边可以组成的 *DAG* 的个数。
- 动态维护前 *k* 小权值和的树。
- 有根树转无根树，同时将其合并到另一棵树里面。

道理我读懂，但是 *subtask1* 跟 *stack tree* 并没有什么关系。

Description1

本 *subtask* 共60pts

给定一个带权序列 $base_{1,n}$ ，我们每次可以调用 *stack tree* 函数，实现删除一个区间的操作——彻底删除的那种

每次删除的代价定义为 $|base_i - base_j|$ ，其中 *i, j* 为当前删除的区间的左、右端点。保证所有 $base \in \mathbb{Z}$

求删除整个序列的最小代价。

Example

样例#1 : (*in*)

```
5
1 2 3 4 5
```

样例#1 : (*out*)

```
3
```

样例#2 : (*in*)

```
6
3 4 5 2 3 9
```

样例#2 : (*out*)

```
5
```

Hint

对于33.3333%的数据， $n \leq 666$

对于66.6666%的数据， $n \leq 2333$

对于100.0000%的数据， $n \leq 500,000$

其中时限分别为：

0 – 33.3333%的数据：1s

33.3333% – 66.6666%的数据：0.5s

66.6666% – 100.0000%的数据：1.2s

保证所有数据的时间限制是`std`运行的两倍（以上）

Background2

某王姓同学经常会fa出一些奇妙的声音，比如什么社区送温暖，友情伸手，我常常因为我不是**而感到和你们格格不入，经常大谈特谈某种政治必修四涉及的社会科学问题。当然，这很正常。

因为某位王姓同学的号召力太大了，导致大家都跟着他一起fa出声音，但是大多数人由于AK UOI之后精疲力尽，所以并不可以模仿的很像。所以现在需要评判谁模仿的更像。

Description2

本 *subtask* 共 40pts

准确来说，每次会有 m 个基佬同学和王姓同学发出差不多的声音，每个人的声音片段是一个字符串。王姓同学本身发出的声音也是一个字符串。相似程度的比较完全依靠其**最长公共子序列的长度**，也就是说，如果某渣渣辉发出了 *AAA* 的愉悦声音，某渣渣淳发出了 *BBB* 的愉悦声音，而某王姓同学发出了 *BFUCK* 的愉悦声音，那么我们就可以说渣渣淳和这位王姓同学更加像一对基佬渣渣淳与王*的发音相似程度最高。

现在给定 m 个同学的发音片段，和王*的发音片段，求与王*发音相似程度最高的基佬同学的最长最长相似发音片段长度，并输出这段公共片段。

Watery Example

样例#1 : (*in*)

```
Fuckqwqwqwq
3
qwqwq
FuckFuckFuck
pksishandsome
```

样例#1 : (*out*)

```
5
qwqwq
```

样例#2 : (*in*)

```
wxisanartist
3
wxFuck
isanFuck
artistFuck
```

样例#2 : (*out*)

```
6
artist
```

Hint

输入：

第一行一个串，表示王*发出的声音。

接下来一行一个 m ，表示有 m 个基佬同学和他一起共同发声；

接下来 m 行，共 m 个串。

输出

共两行，第一行为长度（数值），第二行是用空格隔开的片段（路径）

限制：

对于50%的数据，空间为128MB

对于100%的数据，空间为16MB

什么？你问时间？ Θ (能过)左右。

没准你来gay死出题人他就告诉你数据范围子

Background3

从前有一只yjk和一只lwy，他俩喜欢赌蛤，谁堵输了就要喊对方dai。但是因为本身他们都能AK IMO数竞里面的IOI，所以他们赌的东西都很新奇——赌到底可以有几只蛤飞跨江而不是被水淹没。

蛤们可以选择直接跳到对岸，也可以选择在江中辗转

事实上，江里面有一堆奇特的生物，它们有的喜欢看别人赌蛤，有的不喜欢，我们不妨称这种生物为qwq。蛤在过江时，遇到喜欢赌蛤的qwq，这只qwq会允许蛤在它身上按某种奇怪的方式叠罗汉，而如果某只qwq不喜欢赌蛤，它就只会允许一只蛤在它身上停留。他们想知道在qwq数量一定的情况下，最终可以有多少只蛤过江。

当然，更神奇的是，江的两岸都是喜欢看别人赌蛤的qwq。也就是说蛤们最终跳到的地方是qwq而不是陆地。

对于yjk和lwy这两只AK小能手，这个问题自然不在话下。现在这两个人打算来考考你。

Description3

本subtask共40pts

其中题意在背景里。对于叠罗汉法则：

- 当一个喜欢看人赌蛤的qwq上有多只蛤时，则上面的蛤只能跳到比它大1号的蛤上面。

Example

样例#1：(in)

```
6 3
```

样例#1：(out)

```
256
```

样例#2：(in)

```
666 233
```

样例#2：(out)

494326200

Hint

输入：

一行， n, m

输出

一个数 Ans

限制：

对于50%的数据， $n, m \leq 5000$

对于100%的数据， $n, m \leq 500,0000$

本题所有时限均为0.5s

在该时限下std不一定能跑过去.....但也差不多.....各位加油

谁赌对子yjk喊谁dai哦

Background4

李老铁是LCEZ信竞组里面最神秘的人，他时而和渣渣淳一起肝FGO，时而学一些毒瘤数据结构。并且他有自己独成一体的否滚句式结构，更是让广大Oler望而生畏.....

现在，宇宙第一的李老铁打算研究NP - Hard问题。

Description4

本 *subtask* 共60pts

有一天，李老铁突然翻到了旅行商问题：平面上 n 个点，确定一条连接各点的最短闭合旅程。

显然这个解的一般形式为NPC的（在多项式时间内不可以求出）。但是LLT由于过硬的个人实力，半秒想出了一个 $O(\text{多项式})$ 的解法，让人叹为观止。自此，LCEZ 55级机组已经有凭着贪心证明走上负担的qcr，凭着 $O(1)$ 弗洛伊德走上姚班的zzc，凭着负权图 *dijkstra* 前去中建十八局打铁的pks和凭个人实力让火星大学（MU）哭着求着请进来的LLT

qwq但是李老铁觉得让你们相处NPC问题的多项式时间解法实在不可能，于是他决定简化问题：

知名科学家李老铁，建议通过只考虑双调旅程(*bitonic tour*)来简化问题。这种旅程即为从平面中最左边的点开始，严格地从左到右直至最右点，然后严格地从右到左直至出发点。在这种情况下，多项式的算法是可能的。

——选自qcr编纂《LLT —— 一个疯狂改变OI又疯狂改回去并不知疲倦的人——的学术报告集锦——的梗概》

Example

样例：(*in*)

```
7
0 6
1 0
2 3
5 4
6 1
7 5
8 2
```

样例：(*out*)

```
25.58
```

Hint

输入：

一行， n, m

输出

一个数 Ans ，保留两位小数。

限制：

对于50%的数据， $n \leq 800$

对于100%的数据， $n \leq 500, 0, x_i, y_i \leq 10^5$

本题所有时限均为2s

据说某种复杂度的算法在50%的数据里面可以稳过，不信去问qcr