

第二十六届全国信息学奥林匹克竞赛

NOI 2009

第一试

竞赛时间：2009 年 7 月 27 日上午 8:00-13:00

题目名称	变换序列	诗人小 G	二叉查找树
目录	transform	poet	treapmod
可执行文件名	transform	poet	treapmod
输入文件名	transform.in	poet.in	treapmod.in
输出文件名	transform.out	poet.out	treapmod.out
每个测试点时限	1 秒	30 秒	1 秒
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	无	有	无
题目类型	传统	传统	传统

提交源程序须加后缀

对于 Pascal 语言	transform.pas	poet.pas	treapmod.pas
对于 C 语言	transform.c	poet.c	treapmod.c
对于 C++ 语言	transform.cpp	poet.cpp	treapmod.cpp

注意：最终测试时，所有编译命令均不打开任何优化开关

变换序列

【问题描述】

对于 N 个整数 $0, 1, \dots, N-1$ ，一个变换序列 T 可以将 i 变成 T_i ，其中 $T_i \in \{0, 1, \dots, N-1\}$ 且 $\bigcup_{i=0}^{N-1} \{T_i\} = \{0, 1, \dots, N-1\}$ 。
 $\forall x, y \in \{0, 1, \dots, N-1\}$ ，定义 x 和 y 之间的距离 $D(x, y) = \min\{|x - y|, N - |x - y|\}$ 。给定每个 i 和 T_i 之间的距离 $D(i, T_i)$ ，你需要求出一个满足要求的变换序列 T 。如果有多个满足条件的序列，输出其中字典序最小的一个。

说明：对于两个变换序列 S 和 T ，如果存在 $p < N$ ，满足对于 $i=0, 1, \dots, p-1$ ， $S_i = T_i$ 且 $S_p < T_p$ ，我们称 S 比 T 字典序小。

【输入文件】

输入文件 transform.in 的第一行包含一个整数 N ，表示序列的长度。接下来的一行包含 N 个整数 D_i ，其中 D_i 表示 i 和 T_i 之间的距离。

【输出文件】

输出文件为 transform.out。

如果至少存在一个满足要求的变换序列 T ，则输出文件中包含一行 N 个整数，表示你计算得到的字典序最小的 T ；否则输出 "No Answer"（不含引号）。注意：输出文件中相邻两个数之间用一个空格分开，行末不包含多余空格。

【输入样例】

```
5
1 1 2 2 1
```

【输出样例】

```
1 2 4 0 3
```

【数据规模和约定】

20%的数据中 $N \leq 50$ ；
60%的数据中 $N \leq 500$ ；
100%的数据中 $N \leq 10000$ 。

诗人小 G

【问题描述】

小 G 是一个出色的诗人，经常作诗自娱自乐。但是，他一直被一件事情所困扰，那就是诗的排版问题。

一首诗包含了若干个句子，对于一些连续的短句，可以将它们 用空格隔开 并放在一行中，注意一行中可以放的句子数目是没有限制的。小 G 给每首诗定义了一个行标准长度（行的长度为一行中符号的总个数），他希望排版后每行的长度都和行标准长度相差不远。显然排版时，不应改变原有的句子顺序，并且小 G 不允许把一个句子分在两行或者更多的行内。在满足上面两个条件的情况下，小 G 对于排版中的每行定义了一个 不协调度，为这行的实际长度与行标准长度差值绝对值的 P 次方，而一个排版的不协调度为所有行不协调度的总和。

小 G 最近又作了几首诗，现在请你对这首诗进行排版，使得排版后的诗尽量协调（即 不协调度尽量小），并把排版的结果告诉他。

【输入文件】

输入文件 poet.in 包含多组数据。

第一行包含一个整数 T ，表示诗的数量，接下来是 T 首诗，这里一首诗即为一组数据。每组数据的第一行包含三个由空格分隔的正整数 N 、 L 、 P ，其中 N 表示这首诗句子的数目， L 表示这首诗的行标准长度， P 的含义见问题描述。从第 2 行开始，每行为一个句子，句子由英文字母、数字、标点符号等符号组成（ASCII 码 33~127，但不包含 '-'）。

【输出文件】

输出文件为 poet.out。

对于每组数据，若最小的不协调度不超过 10^{18} ，则第一行一个数表示不协调度，接下来若干行，表示你排版之后的诗。注意：在同一行的相邻两个句子之间需要用空格分开。如果有多个可行解，它们的不协调度都是最小值，则输出任意一个解均可。若最小的不协调度超过 10^{18} ，则输出 "Too hard to arrange" (不包含引号)。每组数据结束后输出 "-----" (不包括引号)，共 20 个 "-"，"-" 的 ASCII 码为 45，请勿输出多余的空行或者空格。

【输入样例】

```
4
4 9 3
brysj,
hhrhl.
yqq!m,
```

```

gsycl.
4 9 2
brysj,
hhrhl.
yqq!m,
gsycl.
1 1005 6
poet
1 1004 6
poet

```

【输出样例】

```

108
brysj,
hhrhl.
yqq!m,
gsycl.
-----
32
brysj, hhrhl.
yqq!m, gsycl.
-----
Too hard to arrange
-----
1000000000000000000000
poet
-----

```

【样例说明】

前两组输入数据中每行的实际长度均为 6，后两组输入数据每行的实际长度均为 4。一个排版方案中每行相邻两个句子之间的空格也算在这行的长度中(可参见样例中第二组数据)。每行末尾没有空格。

【评分方法】

本题设有部分分，当你的程序对于该测试点内每组数据计算得出的不协调度最小值都正确时，能得到本测试点 70% 的分数。在此情况下，若每组数据的排版方案都合法并且得出的不协调度都与输出的相等，则能得到本测试点剩下 30% 的分数。注意，输出格式错误可能会导致本测试点不得分。

【数据规模和约定】

总共 10 个测试点，数据范围满足：

测试点	T	N	L	P
1	≤ 10	≤ 18	≤ 100	≤ 5
2	≤ 10	≤ 2000	≤ 60000	≤ 10
3	≤ 10	≤ 2000	≤ 60000	≤ 10
4	≤ 5	≤ 100000	≤ 200	≤ 10
5	≤ 5	≤ 100000	≤ 200	≤ 10
6	≤ 5	≤ 100000	≤ 3000000	2
7	≤ 5	≤ 100000	≤ 3000000	2
8	≤ 5	≤ 100000	≤ 3000000	≤ 10
9	≤ 5	≤ 100000	≤ 3000000	≤ 10
10	≤ 5	≤ 100000	≤ 3000000	≤ 10

所有测试点中均满足句子长度不超过 30。

二叉查找树

【问题描述】

已知一棵特殊的二叉查找树。根据定义，该二叉查找树中每个结点的**数据值**都比它左**子树**结点的**数据值**大，而比它右**子树**结点的**数据值**小。

另一方面，这棵查找树中每个结点都有一个**权值**，每个结点的**权值**都比它的儿子结点的**权值**要小。

已知树中所有结点的**数据值**各不相同；所有结点的**权值**也各不相同。这时可得出这样一个有趣的结论：如果能够确定树中每个结点的**数据值**和**权值**，那么树的形态便可以唯一确定。因为这样的一棵树可以看成是按照**权值**从小到大顺序插入结点所得到的、按照**数据值**排序的二叉查找树。

一个结点在树中的**深度**定义为它到树根的距离加 1。因此树的根结点的深度为 1。

每个结点除了**数据值**和**权值**以外，还有一个**访问频度**。我们定义一个结点在树中的**访问代价**为它的**访问频度**乘以它在树中的**深度**。**整棵树的访问代价**定义为所有结点在树中的**访问代价**之和。

现在给定每个结点的**数据值**、**权值**和**访问频度**，你可以根据需要修改某些结点的**权值**，但每次修改你会付出 K 的**额外修改代价**。你可以把结点的**权值**改为任何实数，但是修改后所有结点的**权值**必须仍保持互不相同。现在你要解决的问题是，**整棵树的访问代价**与**额外修改代价**的和最小是多少？

【输入文件】

输入文件名为 treapmod.in。

输入文件第一行包含两个正整数 N 和 K 。 N 为结点的个数， K 为每次修改所需的**额外修改代价**。

接下来一行包含 N 个非负整数，是每个结点的**数据值**。

再接下来一行包含 N 个非负整数，是每个结点的**权值**。

再接下来一行包含 N 个非负整数，是每个结点的**访问频度**。

所有的数据值、权值、访问频度均不超过 400000。每两个数之间都有一个空格分隔，且行尾没有空格。

【输出文件】

输出文件名为 treapmod.out。输出文件只有一个数字，即你所能得到的**整棵树的访问代价**与**额外修改代价**之和的最小值。

【输入样例】

4 10

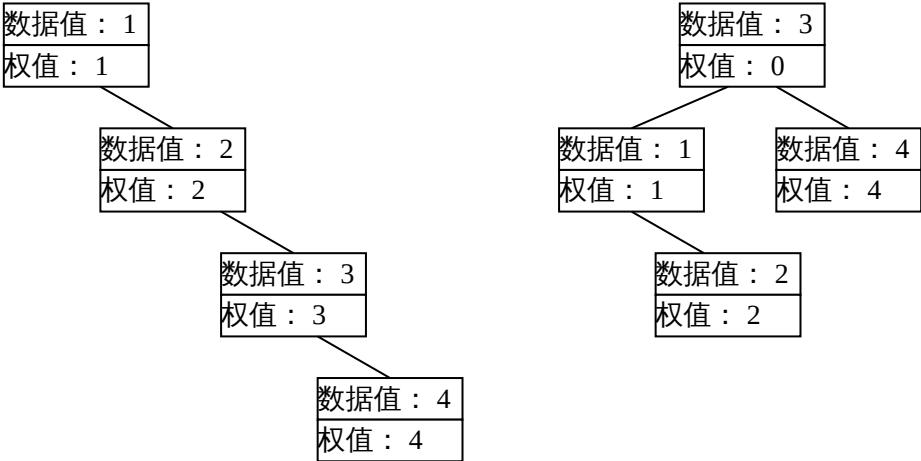
1 2 3 4
1 2 3 4
1 2 3 4

【输出样例】

29

【样例说明】

输入的原图是左图，它的访问代价是 $1\times 1+2\times 2+3\times 3+4\times 4=30$ 。最佳的修改方案是把输入中的第 3 个结点的权值改成 0，得到右图，访问代价是 $1\times 2+2\times 3+3\times 1+4\times 2=19$ ，加上额外修改代价 10，一共是 29。



【数据规模】

- 40%的数据满足 $N \leq 30$;
- 70%的数据满足 $N \leq 50$;
- 100%的数据满足 $N \leq 70, 1 \leq K \leq 30000000$ 。