

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
```

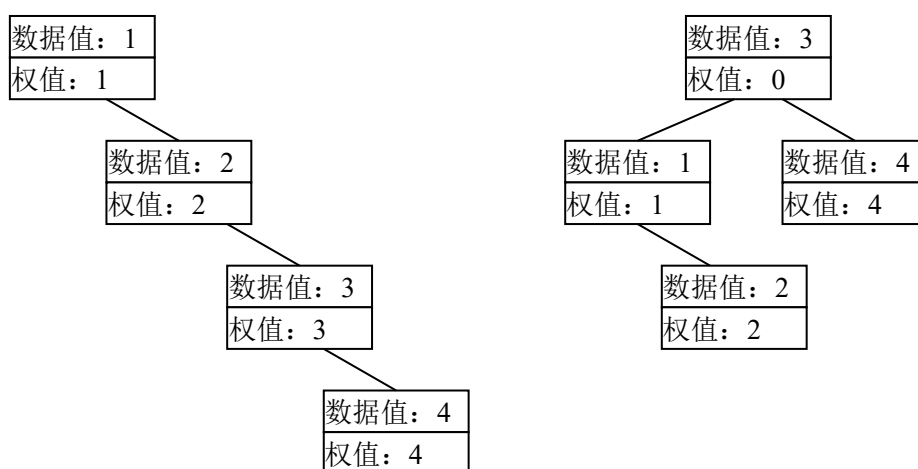
1 2 3 4
1 2 3 4

【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 $N \leq 70, 1 \leq K \leq 30000000$ 。