

## 序列变换 (bracket)

### 【题目描述】

你手里有一个长度为  $2n$  的合法括号序列  $s$ 。 $s$  的每一个左括号有一个权值。

在你眼中，不同的括号序列带来的视觉美感不尽相同。因此，你对具有某一种结构的括号序列特别喜欢，而讨厌具有其他一些结构的括号序列。你希望对  $s$  进行一些变换，以消除掉一些自己不喜欢的结构。

具体而言，形如  $(A()B)$ （其中  $A, B$  均为合法括号序列，下同）的结构是你喜欢的，而形如  $(A)(B)$  的结构是你不喜欢的。你有两种操作来改变括号之间的位置。

这两种操作如下：

- 操作 1：交换形如  $p(A)(B)q$  的串中  $A$  和  $B$  之间的两个括号，变换为  $p(A()B)q$ （其中  $p, q$  为任意串，可以为空，但不一定分别为合法括号序列，下同），它的代价为  $x$  乘  $(A)$  中第一个左括号的权值加上  $y$  乘  $(B)$  中第一个左括号的权值，其中  $x, y \in \{0, 1\}$ ；
- 操作 2：交换形如  $pABq$  的串中的  $A$  和  $B$ ，变换为  $pBAq$ ，这个操作不需要代价。注意：交换的时候所有左括号的权值是跟着这个括号一起交换的。

你现在想知道的是，将  $s$  变换为一个不包含你不喜欢的结构的括号序列至少需要多少代价？

### 【输入格式】

从文件 `bracket.in` 中读入数据。

第一行三个整数  $n, x, y$ 。

第二行一个长度为  $2n$  的合法括号序列，表示  $s$ 。

第三行  $n$  个正整数，其中第  $i$  个表示左数第  $i$  个左括号的权值。

### 【输出格式】

输出到文件 `bracket.out` 中。

一行一个整数，表示把  $s$  变换为一个不包含你不喜欢的结构的括号序列至少需要的代价。

### 【样例 1 输入】

```
1 2 0 1
2 ()()
3 1 3
```

**【样例 1 输出】**

```
1 1
```

**【样例 1 解释】**

最优方案是先使用操作 2 交换两对括号，然后使用操作 1（此时  $A, B, p, q$  都是空串）交换中间的两个括号，代价为  $B$  左边那个括号的权值，也就是 1。最后得到括号序列  $(())$ ，不包含你不喜欢的结构。

**【样例 2 输入】**

```
1 2 1 0
2 ()()
3 1 3
```

**【样例 2 输出】**

```
1 1
```

**【样例 2 解释】**

最优方案是直接使用操作 1，因为此时计算代价的方式不同了，这次只算  $A$  左边的那个括号的权值作为代价。

**【样例 3】**

见选手目录下的 *bracket/bracket3.in* 与 *bracket/bracket3.ans*。

**【子任务】**

保证  $2 \leq n \leq 400000$ ,  $0 \leq x, y \leq 1$ 。

保证所有的权值在  $[1, 10^7]$  之内。

测试点编号	特殊限制
1 ~ 3	$n \leq 8$
4 ~ 5	所有权值均相等
6 ~ 8	$n \leq 20$
9 ~ 12	$x = 0, y = 1$
13 ~ 16	$n \leq 2000$
17 ~ 25	无特殊限制

**【提示】**

称一个字符串  $s$  为合法括号序列，当且仅当  $s$  仅由数量相等的字符 ( 和 ) 组成，且对于  $s$  的每一个前缀而言，其中 ( 的数量均不少于 ) 的数量。特别地，空串也是合法括号序列。