



排列鞋子

Adnan 拥有巴库最大的鞋店。现在有一个装着 n 双鞋的箱子刚运到他的鞋店。每双鞋是大小相同的两只：一只左脚，一只右脚。Adnan 把这 $2n$ 只鞋排成一行，该行总共有 $2n$ 个位置，由左到右编号为从 0 到 $2n - 1$ 。

Adnan 想把这些鞋子重新排成合法的排列。一个排列是合法的，当且仅当对于所有 i ($0 \leq i \leq n - 1$)，以下条件都成立：

- 在位置 $2i$ 和 $2i + 1$ 上的鞋子大小相同。
- 在位置 $2i$ 上的鞋子是一只左脚鞋。
- 在位置 $2i + 1$ 上的鞋子是一只右脚鞋。

为实现上述目标，Adnan 可以做一系列的对调。在每次对调中，他选择当前相邻的两只鞋子进行对调（也就是把它们拿起来，然后将每只鞋子放回到另一只鞋子原来的位置上）。两只鞋子是相邻的，如果其位置编号的差为 1 。

请找出 Adnan 最少需要做多少次对调，才能得到一个合法的排列。

实现细节

你需要实现下述函数：

```
int64 count_swaps(int[] S)
```

- S ：一个包括 $2n$ 个整数的数组。对于每个 i ($0 \leq i \leq 2n - 1$)， $|S[i]|$ 是一个非零的值，等于最初在位置 i 上的鞋子的大小。这里 $|x|$ 表示 x 的绝对值，在 $x > 0$ 时等于 x ，而在 $x < 0$ 时等于 $-x$ 。如果 $S[i] < 0$ ，位置 i 上的鞋子是一只左脚鞋，否则是一只右脚鞋。
- 该函数应当返回为得到合法的排列而最少要做的（相邻鞋子）对调的次数。

例子

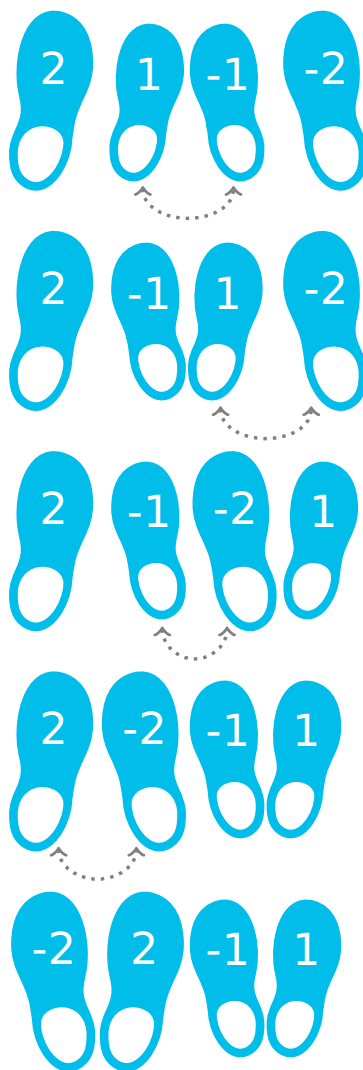
例 1

考虑下述调用：

```
count_swaps([2, 1, -1, -2])
```

Adnan 可以通过 4 次对调而得到一个合法的排列。

例如，他可以先对调鞋子 1 和 -1，再对调 1 和 -2，再对调 -1 和 -2，最后对调 2 和 -2。随后他就可以得到合法的排列 $[-2, 2, -1, 1]$ 。无法用少于 4 次对调就得到合法的排列。因此，该函数应当返回 4。



例 2

在下面的例子中，所有鞋子的大小相同：

```
count_swaps([-2, 2, 2, -2, -2, 2])
```

Adnan可以对调在位置 2 和 3 上的鞋子来得到合法的排列 $[-2, 2, -2, 2, -2, 2]$ ，因此该函数应当返回 1。

限制条件

- $1 \leq n \leq 100\,000$
- 对于所有 i ($0 \leq i \leq 2n - 1$)，有 $1 \leq |S[i]| \leq n$ 。
- 总有某个合法的排列可以经由一系列对调而得到。

子任务

1. (10 分) $n = 1$
2. (20 分) $n \leq 8$
3. (20 分) 所有鞋子大小都是相同的。
4. (15 分) 所有在位置 $0, \dots, n - 1$ 上的鞋子都是左脚鞋，而所有在位置 $n, \dots, 2n - 1$ 上的鞋都是右脚鞋。而且对于所有 i ($0 \leq i \leq n - 1$)，在位置 i 和 $i + n$ 上的鞋子大小是相同的。
5. (20 分) $n \leq 1000$
6. (15 分) 没有任何附加限制。

评测程序示例

评测程序示例读取如下格式的输入：

- 第 1 行： n
- 第 2 行： $S[0] \ S[1] \ S[2] \ \dots \ S[2n - 1]$

评测程序示例输出单独的一行，其中包含 `count_swaps` 的返回值。