

银币。每匹马只能用一次。齐王的马好,同等级的马,齐王的总是比田忌的要好一点。于是每次和齐王赛马,田忌总会输 600 银币。

田忌很沮丧,直到他遇到了著名的军师——孙臆。田忌采用了孙臆的计策之后,三场比赛下来,轻松而优雅地赢了齐王 200 银币。

这实在是个很简单的计策。由于齐王总是先出最好的马,再出次好的,所以田忌用常规马对齐王的超级马,用自己的超级马对齐王的上级马,用自己的上级马对齐王的常规马,以两胜一负的战绩赢得 200 银币。实在很简单。

如果不止三匹马怎么办? 这个问题很显然可以转化成二分图最佳匹配的问题。把田忌的马放左边,把齐王的马放右边。田忌的马 A 和齐王的马 B 之间,如果田忌的马胜,则连一条权为 200 的边;如果平局,则连一条权为 0 的边;如果输,则连一条权为 -200 的边……如果你不会求最佳匹配,用最小费用最大流也可以啊。

然而,赛马问题是一种特殊的二分图最佳匹配的问题,上面的算法过于先进了,简直是杀鸡用牛刀。现在,就请你设计一个简单的算法解决这个问题。

【输入格式】

第一行一个整数 N ,表示他们各有 N 匹马(两人拥有的马的数目相同)。第二行 N 个整数,每个整数都代表田忌的某匹马的速度值($0 \leq \text{速度值} \leq 100$)。第三行 N 个整数,描述齐王的马的速度值。两马相遇,根据速度值的大小就可以知道哪匹马会胜出。如果速度值相同,则和局,谁也不拿钱。

【输出格式】

仅一行,一个整数,表示田忌最大能得到多少银币。

【样例输入】

```
3
92 83 71
95 87 74
```

【样例输出】

```
200
```

【数据范围】

对于 20% 的数据, $1 \leq N \leq 65$;

对于 40% 的数据, $1 \leq N \leq 250$;

对于 100% 的数据, $1 \leq N \leq 2000$ 。