

{% note info %} **摘要** Title: 1801. 蹄子剪刀布 Tag: 枚举、全排列 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

[@TOC](#)

1801. 蹄子剪刀布

- 题意

你可能听说过“石头剪刀布”的游戏。这个游戏在牛当中同样流行，它们称之为“蹄子剪刀布”。游戏的规则非常简单，两头牛相互对抗，数到三之后各出一个表示蹄子，剪刀或布的手势。蹄子赢剪刀，剪刀赢布，布赢蹄子。例如，第一头牛出“蹄子”手势，第二头牛出“布”手势，则第二头牛获胜。如果两头牛出相同的手势，则算平局。农夫约翰的两头奶牛正在进行 N 轮“蹄子剪刀布”对抗，他看的十分入迷。不幸的是，虽然他可以看到奶牛正在做出三种不同类型的手势，但他却无法分辨出哪一个代表“蹄子”，哪一个代表“布”以及哪一个代表“剪刀”。不知道这三种手势的具体含义的情况下，农夫约翰给这三种手势分配了编号 1,2,3。手势 1 可能代表“蹄子”，可能代表“剪刀”，也可能代表“布”，反正他傻傻分不清楚。给出两头奶牛在 N 场比赛中所做出的具体手势对应的编号，请你判断第一头奶牛最多可能赢多少盘对抗。

- 思路

用 1, 2, 3 代表三个种类，将所有情况全排列，将此数值映射成下标，用下标来判断大小

- 代码

```
...
Author: NEFU AB-IN
Date: 2022-01-30 13:55:06
FilePath: \ACM\Acwing\1801.py
LastEditTime: 2022-01-31 10:46:22
...

from collections import defaultdict
from itertools import permutations

b = []
c = defaultdict(int)

def judge(x, y):
    x = c[x]
    y = c[y]
    if x == 0 and y == 1:
        return True
    elif x == 1 and y == 2:
```

```
        return True
    elif x == 2 and y == 0:
        return True
    else:
        return False

if __name__ == "__main__":
    n = int(input())
    a = list(permutations(range(1, 4), 3))
    for i in range(n):
        b.append(tuple(map(int, input().split())))
    res = 0
    for i in a:
        cnt = 0
        # 映射成下标 如 2->0 1->1 3->2
        for j in range(len(i)):
            c[i[j]] = j
        for j in b:
            # 如 传入 2 1
            # 那么其实传过去的是1, 0
            # 这样就可以比较了
            if judge(j[0], j[1]):
                cnt += 1
        res = max(res, cnt)
    print(res)
```