请记得修改次数去掉打印

*# 导入随机数模块***import** random  
  
*# 定义队伍和初始积分、净胜分*teams = [**"甲"**, **"乙"**, **"丙"**, **"丁"**, **"戊"**, **"己"**]  
points = [2, 2, 2, 2, 1, 0]  
diffs = [3, 3, 2, 2, -3, -6]  
  
*# 定义剩余的比赛*matches = [(**"甲"**, **"丙"**), (**"乙"**, **"己"**), (**"丁"**, **"戊"**), (**"乙"**, **"丙"**), (**"甲"**, **"丁"**), (**"戊"**, **"己"**)]  
  
*# 定义模拟次数*n = 100  
  
*# 定义队伍N（比如己是第6个队伍）*namenum = 6  
  
*# 定义N队淘汰和不淘汰的次数*out = 0  
in\_ = 0  
  
*# 定义N队不淘汰时的排名分布*rank\_dist = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0}  
  
*# 定义抽签的次数*draw = 0  
  
*# 定义比分概率函数***def** generate\_score():  
 *# 定义每个比分的概率* p = {-3: 0.125, -2: 0.1875, -1: 0.1875, 1: 0.1875, 2: 0.1875, 3: 0.125}  
 *# 生成一个随机数* r = random.random()  
 *# 初始化累积概率* cum\_p = 0  
 *# 遍历所有可能的比分* **for** score **in** p:  
 *# 更新累积概率* cum\_p += p[score]  
 *# 如果随机数小于等于累积概率，返回该比分* **if** r <= cum\_p:  
 **return** score  
  
*# 查找某队当前排名索引的函数。定义一个函数find\_fifth，它接受两个参数：积分列表和净胜分列表***def** find\_fifth(points, diffs,x):  
 *# 创建一个空列表rank\_list，用来存储每个队伍的积分、净胜分和索引* rank\_list = []  
 *# 使用一个循环，遍历积分列表和净胜分列表，将每个队伍的信息添加到rank\_list中* **for** i **in** range(len(points)):  
 rank\_list.append((points[i], diffs[i], i))  
 *# 使用sorted函数，根据积分和净胜分对rank\_list进行降序排序* rank\_list = sorted(rank\_list, key=**lambda** x: (x[0], x[1]), reverse=**True**)  
 *# 返回rank\_list中第五个元素的最后一个值，即第五名的索引* **return** rank\_list[x-1][2]  
  
*# 开始模拟***for** i **in** range(n):  
 *# 复制当前积分和净胜分* p = points.copy()  
 d = diffs.copy()  
 *# 遍历每场比赛* **for** m **in** matches:  
 *# 随机生成bo5的比分* score = generate\_score()  
 *# 计算胜负队伍的索引,得分正数是左边赢，负数是右边赢* win\_idx = teams.index(m[score < 0])  
 lose\_idx = teams.index(m[score > 0])  
 *# 更新积分和净胜分* p[win\_idx] += 1  
 p[lose\_idx] += 0  
 d[win\_idx] += abs(score)  
 d[lose\_idx] -= abs(score)  
 print(**f"{**teams[win\_idx]**}以{**abs(score)**}分差，战胜{**teams[lose\_idx]**}"**)  
 *# 计算N队的排名,循环6次，不与自己比较* rank = sum([p[j] > p[namenum-1] **or** (p[j] == p[namenum-1] **and** d[j] > d[namenum-1]) **for** j **in** range(6) **if** j !=namenum-1]) + 1  
 *# 判断己队是否淘汰，1-3名晋级* **if** rank == 1 **or** rank == 2 **or** rank == 3:  
 *# 不淘汰次数加一* in\_ += 1  
 *# 更新排名分布* rank\_dist[rank] += 1  
 **elif** rank == 6:  
 *# 淘汰次数加一* out += 1  
 **elif** rank == 4:  
 *# 计算第五名的索引* fifth\_idx = find\_fifth(p,d,5)  
 **if** p[fifth\_idx] != p[namenum-1] **or** d[fifth\_idx] != d[namenum-1]:  
 *# 不淘汰次数加一* in\_ += 1  
 *# 更新排名分布* rank\_dist[rank] += 1  
 print(**f"己队最终积分是{**p[namenum-1]**}，净胜分是{**d[namenum-1]**}，排位是{**rank**},第五名索引是{**fifth\_idx**}"**)  
 print(**f"其他队积分是：甲积分{**p[0]**},净胜分{**d[0]**}"**)  
 print(**f"其他队积分是：乙积分{**p[1]**},净胜分{**d[1]**}"**)  
 print(**f"其他队积分是：丙积分{**p[2]**},净胜分{**d[2]**}"**)  
 print(**f"其他队积分是：丁积分{**p[3]**},净胜分{**d[3]**}"**)  
 print(**f"其他队积分是：雾积分{**p[4]**},净胜分{**d[4]**}"**)  
 **else**:  
 *# 随机生成一个0或1的整数，表示抽签结果，0表示淘汰，1表示不淘汰* result = random.randint(0,1)  
 **if** result == 0:  
 *# 淘汰次数加一* out += 1  
 **else**:  
 *# 不淘汰次数加一* in\_ += 1  
 *# 更新排名分布* rank\_dist[rank] += 1  
 *# 抽签次数加一* draw += 1  
 **elif** rank == 5:  
 *# 计算第四名的索引* fourth\_idx = fifth\_idx = find\_fifth(p,d,4)  
 **if** p[fourth\_idx] != p[namenum-1] **or** d[fourth\_idx] != d[namenum-1]:  
 *# 淘汰次数加一* out += 1  
 **else**:  
 *# 随机生成一个0或1的整数，表示抽签结果，0表示淘汰，1表示不淘汰* result = random.randint(0,1)  
 **if** result == 0:  
 *# 淘汰次数加一* out += 1  
 **else**:  
 *# 不淘汰次数加一* in\_ += 1  
 *# 更新排名分布* rank\_dist[rank] += 1  
 print(**f"己队最终积分是{**p[namenum - 1]**}，净胜分是{**d[namenum - 1]**}，排位是{**rank**},第四名索引是{**fourth\_idx**}"**)  
 print(**f"其他队积分是：甲积分{**p[0]**},净胜分{**d[0]**}"**)  
 print(**f"其他队积分是：乙积分{**p[1]**},净胜分{**d[1]**}"**)  
 print(**f"其他队积分是：丙积分{**p[2]**},净胜分{**d[2]**}"**)  
 print(**f"其他队积分是：丁积分{**p[3]**},净胜分{**d[3]**}"**)  
 print(**f"其他队积分是：雾积分{**p[4]**},净胜分{**d[4]**}"**)  
 *# 抽签次数加一* draw += 1  
  
 **elif** in\_ == 1 :  
 **break***# 计算己队淘汰和不淘汰的概率*out\_prob = out / n \* 100  
in\_prob = in\_ / n \* 100  
  
*# 计算抽签的概率*draw\_prob = draw / n \* 100  
  
*# 打印结果*print(**f"{**teams[namenum-1]**}队淘汰的概率是{**out\_prob**:.2f}%"**)  
print(**f"{**teams[namenum-1]**}队不淘汰的概率是{**in\_prob**:.2f}%"**)  
print(**f"{**teams[namenum-1]**}队抽签的概率是{**draw\_prob**:.2f}%"**)  
print(**f"{**teams[namenum-1]**}队不淘汰的情况下，最有可能的排名是{**max(rank\_dist, key=rank\_dist.get)**}，出现了{**rank\_dist[max(rank\_dist, key=rank\_dist.get)] / n \* 100**:.2f}%的次数"**)  
print(**f"{**teams[namenum-1]**}队不淘汰的情况下，最不可能的排名是{**min(rank\_dist, key=rank\_dist.get)**}，出现了{**rank\_dist[min(rank\_dist, key=rank\_dist.get)] / n \* 100**:.2f}%的次数"**)  
print(**f"{**teams[namenum-1]**}队排名为3的概率是{**rank\_dist[3]/ n \* 100**:.2f}%"**)