

# 囚徒困境博弈对战系统

## 1. 背景知识：囚徒困境描述

警方逮捕甲、乙两名嫌疑犯，但没有足够证据指控二人有罪。于是警方分开囚禁嫌疑犯，分别和二人见面，并向双方提供以下相同的选择：

- 若一人认罪并作证，指控对方（相关术语称“背叛”对方），而对方保持沉默，此人将即时获释，沉默者将判监 10 年。
- 若二人都保持沉默（相关术语称互相“合作”），则二人同样判监半年。
- 若二人都互相检举（互相“背叛”），则二人同样判监 5 年。

## 2. 需求描述

用程序模拟囚徒做出“合作”还是“背叛”的决策，如果一人选择背叛，对方选择合作，背叛者积 5 分，合作者积 0 分；如果双方都选择合作，则双方都积 3 分；如果双方都背叛，双方都积 1 分。

用表格描述如下：

|     | 乙合作         | 乙背叛         |
|-----|-------------|-------------|
| 甲合作 | 甲 3 分、乙 3 分 | 甲 0 分、乙 5 分 |
| 甲背叛 | 甲 5 分，乙 0 分 | 甲 1 分、乙 1 分 |

写 6 个不同的模拟囚徒决策函数，同时写一个调度函数，让 6 个囚徒函数两两对战，当两个囚徒函数完成一次对战之后，将有一定概率会进入下一轮对战。

当所有的对战结束后，用 `console.log` 输出各囚徒函数每次对战的平均分（即：总积分/总对战次数），以及囚徒函数两两之间对战的总积分、每次对战的策略（放在一行里一起输出，如 1. 合作 vs 背叛 2.合作 vs 合作 3.背叛 vs 背叛 4.背叛 vs 合作）。

囚徒决策函数

入参：res1List,res2List，双方对战的历史决策数据数组。决策函数基于入参数据做出决策，不能使用入参以外的数据

返回值："合作" or "背叛"

调度函数

入参：囚徒函数数组（数组 item 包含两个字段，囚徒函数和名称）、2 个囚徒函数之间完成一轮对战后继续对战的概率

为每个囚徒函数起名字，并传入调度函数，继续对战概率传入 0.9，示例：

调度函数([f1, 'f1'], [f2, 'f2'], [f3, 'f3'], [f4, 'f4'], [f5, 'f5'], [f6, 'f6']), 0.9)

用 js 实现代码，可以在 chrome 控制台运行并输出对战结果