

### Problem: ROCK-PAPER-SCISSORS

In a game of rock-paper-scissors (RPS), each player chooses to play Rock (R), Paper (P), or Scissors (S).

The rules are: R beats S; S beats P; and P beats R. We will encode a rock-paper-scissors game as a list, where the elements are themselves 2-element lists that encode a player's name and a player's selected move, as shown below:

```
[["Armando", "P"], ["Dave", "S"]] # Dave would win since S > P
```

Part A: Write a method `rps_game_winner` that takes a two-element list and behaves as follows:

- If the number of players is not equal to 2, raise `WrongNumberOfPlayersError`.
- If either player's strategy is something other than "R", "P" or "S" (case-insensitive), raise `NoSuchStrategyError`.
- Otherwise, return the name and move of the winning player. If both players play the same move, the first player is the winner.

Part B: We will define a rock-paper-scissors tournament to be an array of games in which each player always plays the same move.

A rock-paper-scissors tournament is encoded as a bracketed array of games:

```
[
  [
    ["Armando", "P"], ["Dave", "S"] ],
    ["Richard", "R"], ["Michael", "S"] ],
  ],
  [
    ["Allen", "S"], ["Omer", "P"] ],
    ["David E.", "R"], ["Richard X.", "P"] ]
  ]
]
```

In the tournament above Armando will always play P and Dave will always play S. This tournament plays out as follows:

- Dave would beat Armando ( $S > P$ ),
- Richard would beat Michael ( $R > S$ ), and then
- Dave and Richard would play (Richard wins since  $R > S$ ).

Similarly,

- Allen would beat Omer,
- Richard X would beat David E., and
- Allen and Richard X. would play (Allen wins since  $S > P$ ).

Finally,

- Richard would beat Allen since  $R > S$ .

**Comments:**

- Note that the tournament continues until there is only a single winner.
- Tournaments can be nested arbitrarily deep, i.e., it may require multiple rounds to get to a single winner.
- You can assume that the initial tournament is well-formed (that is, there are  $2^n$  players, and each one participates in exactly one match per round).
- Write a method `rps_tournament_winner` that takes a tournament encoded as a bracketed array and returns the winner (for the above example, it should return `["Richard", "R"]`).