Generalized Specker Game: Rules

Overview

A multiplayer game with `p` players and `n` heaps of coins. Players take turns cyclically, removing and redistributing coins under specific rules. The last player to make a valid move wins.

Rules

Players and Heaps

- **Players**: `p` players (numbered `0` to `p-1`).
- **Heaps**: `n` heaps (numbered `0` to `n-1`) with identical coins.

Turn Order

- Players take turns in **cyclic order**: `Player 0` → `Player 1` → ... → `Player p-1` → `Player 0` (repeats).

Move Mechanics

On their turn, a player must:

1. **Select two heaps**:
   - **Source heap `i`**: Must have **≥ `k` coins** (`k > 0`).
   - **Target heap `j`**: Arbitrary if no coins are added (`m = 0`).
2. **Actions**:
   - Remove `k` coins from heap `i`.
   - Add `m` coins to heap `j`, where `0 ≤ m < k`.

Special Case

- If `m = 0` (no coins added to `j`), the choice of `j` is irrelevant.

Winning Condition

- The game ends when **all heaps are empty**.
- The **last player to make a valid move** wins.

Example

- **Players**: `3` (`0`, `1`, `2`).
- **Heaps**: `2` heaps with `[5, 5]` coins initially.

Turn Sequence:

1. **Player 0** removes `3` coins from heap `0`, adds `2` to heap `1` → `[2, 7]`.
2. **Player 1** removes `5` from heap `1`, adds `0` → `[2, 2]`.
3. **Player 2** removes `2` from heap `0`, adds `1` to heap `1` → `[0, 3]`.
4. Game continues until all heaps are empty.

Constraints

1. **Valid Moves**:

   - `k > 0` (must remove at least 1 coin).
   - `m < k` (added coins < removed coins).
   - Source heap `i` must have `≥ k` coins.

2. **Termination**: Game ends **only** when all heaps are empty.

Implementation Notes

- Use classes like `Move`, `State`, and `Player` to model the game.
- Throw `logic_error` for invalid moves.