Aveerup Chowdhury
2105112
CSE318

I have implemented 5 heuristics in my chain reaction game, their name and brief description is given below:

Heuristic	Description
`atom_count_difference`	-Total number of player's atoms on the board.
`place_count_difference`	-Number of cells occupied by the player.
`critical_mass_score`	-Number of cells close to explosion (for example, atoms = critical mass – 1).
`front_line_pressure`	-Measures player pressure on opponent by counting bordering cells.
`survivability_score`	-Evaluates how survivable current placements are — low risk of opponent chain reactions or backfire.

Not any of them is perfect. They all have some flaws. Some excel in some situations, some fall behind in some situations. So, we need to use them based on the situation or strategy we want our ai to tackle or apply. Their pros and cons are given below:

1. `atom_count_difference`

- Counts the total number of atoms the player has on the board.
- Measures **raw power** — more atoms usually means stronger position.
- Doesn't consider atom position or vulnerability to chain reactions.

2. place_count_difference

- Counts how many cells are occupied by the player (regardless of how many atoms each has).
- Measures **territorial control** or spread on the board.
- Doesn't reflect how strong each position is (e.g., 1 atom vs. 3 atoms treated equally).

3. critical_mass_score

- Counts how many of the player's cells are **one move away** from exploding (atoms = critical mass - 1).
- Identifies **explosive potential**, i.e., cells that could trigger chain reactions.
- **Limitation:** High value may also mean high **risk** if opponent attacks first.

4. front_line_pressure

- Measures how many of the player's cells are **adjacent to opponent cells**.
- Evaluates **aggressive pressure zones** where the player can attack soon.
- Doesn't reflect defensive strength or depth of control.

5. survivability_score

- Assesses how **safe** the player's positions are from being exploded by opponents.
- Helps estimate **long-term stability** and defense against chain losses.
- More defensive; may miss out on offensive opportunities.

We can not singly tell any one of them better. So, better approach would be to use all of them together and assign them with weights according to our playing strategy. So, the general eq would be something like -

score = a * atom_cound_diff + b * place_count_diff + c * survivability_score + d * front_line_pressure + e * critical_mass_error

here, a, b, c, d, e are weights that are arbitrary and depends on our strategy.