Pawnscape

Aryan Katiyar

May 2024

1 Introduction

In this game, you're provided with an NxN board where each player has N pawns positioned on both ends in the first and last rows. Victory is achieved by getting one of your pawns to the opposite end. If no feasible moves remain, the game results in a draw. Pawns are restricted to moving one step forward, even for their initial move.

2 Helper code

To understand more about the game, run the the helper.cpp provided to you.For now choose white. You need to move for both the set of players as no strategy has been implemented in this file.

2.1 How to move

Suppose we want to move the pawn at (4,0) to (3,0). Enter your move as"3040" i.e (row final,column final,row initial,column initial).

```
Current Board is
  0 1 2 3 4
0 B B B B B
1 * * * * *
2 * * * * *
3 * * * * *
4 W W W W W

Enter your move as white
```

```
Enter your move as white
3040

Current Board is
  0 1 2 3 4
0 B B B B B
1 * * * * *
2 * * * * *
3 W * * * *
4 * W W W W
```

Figure 1: board before and after the move

Play around till you get the hang of it.

3 Tasks

1. For n=4 find if there is a strategy that ensures that white never loses irrespective of what black does.
2. For n=4 find if there is a strategy that ensures that white always wins irrespective of what black does. If the answer of the first part is NO then you can ignore this part.
3. (optional) Code you strategies for parts 1 and 2(if they exist) in the "To do 1" section provided int the helper.cpp . Replace the existing code with your strategy in this section.
4. Now play this game when you move second i.e. you have chosen the black pieces. For n=4 find if there is a strategy that ensures that black always wins irrespective of what white does. Ignore this if the answer of part 1 is YES.
5. For n=4 find if there is a strategy that ensures that black never loses irrespective of what white does. Ignore this if the answer of part 2 is YES.
6. (optional) Code your strategies for part 3 and 4(if they exist) in the "To do 2" section.
7. (challenge/optional) Try part 1-6 for n=5
8. (For those who are comfortable in Python/optional) Implement this game on your own in python. You can implement it in any way you like. Also code your strategies as well.

Try it for n=3 to get some idea. You will realise that the game always ends up in a draw if both the players play optimally. The task for this week is to try part 1,2,4 and 5.
NOTE: If you can prove that the answer for part 1 is NO and the game doesn't end in a draw if both players play optimally,then you are done for the week as part 2,3 and 4 end up being false and there is no strategy to code :) .