QUESTIONS AND EXTENSIONS

dimkadimon

March 31 Marathon Match 125

I used to play this game a lot when I was young. It has a nice balance between being greedy (getting lines) and trying to keep the space open to allow for easier movement. At one stage I got to a point where I could play "almost forever" without filling up the grid. Of course when you play manually, you get tired and eventually make a mistake, which is hard to recover from.

Now I am interested to hear your answers to the following:

1. Was it possible to make lines on every case? I figured that in the worst case (N=7 and C=9) we will get 5+ balls of at least one colour, but it may be too hard to form a line.
2. Have you ever managed to remove all the balls from a grid?
3. What is the most number of balls you have removed in one move? In theory it is actually possible to remove 33 balls in the form of a "death star", but you need a lot of

luck for the game to place the last ball in the center of the star.

4. Was it better to focus on making large lines or just try to "survive" until 1000 moves?
5. Have you managed to clear lines of different colour in one turn?
6. Was the information about the next balls useful and how do you use it?
7. This competition was unusual, because we had a very similar problem in MM5. Was the information about past approaches useful? How did this affect your solution?

View: **Threaded** | Flat

Quote ·Reply

COMMENTS

sullyper

March 31, 2021 9:49PM

1. My solution clearly doesn't, but I tried the first 20 seeds manually with n = 7 and C = 9, each time I managed to make a line (a couple of times, it was close...), so I suspect that a smart program should almost always manage to do that. I said almost always because I am pretty sure that no matter what we do, an eviel agent could pick the color and position to prevent it from happening. It seems simple, like make the balls appears in checkers pattern to quickly block the mobility, you can fill half of the board like that before a color appears 5 times, then it seems really easy to fill the remaining cells with colors which won't allow to get 5 in a row.
   So in theory no it's not possible, but manually it seems that in most cases it should be easy
2. Never checked if it happens, never tried manually, it would require luck anyway, or very risky play (let the board fill to control where the new balls arrive, and then clean everything with a succession of moves), so I am gonna assume that no, I never did that

3. my program, probably not more than 9/10, manually, seems really simple to get 20+ for N=14, C=3, I did 22 just now, but I messed up by putting some ont he wrong diagonal for a couple of moves.
4. I feel like when C=3 it's easy to survive, and therefore you want to make some longer line, at least at first
   When C >4, it's hard to survive, my program would never reach the end, so surviving is more important, because preparing longer move will almost kill you right away.
5. No idea, I bet it happend for C=3, because I expect a lot of 4 in a row to happen naturally, but never checked.
6. I tried, I failed, but manually I def use this information especially for N=7 and C=9, to see which color to target
7. See the other post, yes completely, and that was unsatisfying, in the meantime if I had not read the forum, I am not sure I would have even competed given my lack of time. 😃

Quote · Reply

nika

April 1, 2021 6:42AM

I haven't properly competed but want to leave some feedback here.

First of all, it was very hard and interesting which would make it great problem if not the bad coincidence (which was somewhat highly like due to directly reusing well know game).

What I really want to address is the tendency of having very wide constraint range on this and some previous problems.

It is already hard for some of us to have enough free time in 7 day contest (only 1 weekend) and having constraints that force you create multiple different

solutions is just not feasible. Here the cases with C=3, C=6 and C=9 require completely different approach and the central case C=6,7 is already extremely difficult to properly solve in such time frame.
Special mention for cases (and not just rare) where even getting positive score is hard - this is just very poor design for marathon match format in my opinion. MM5 was just N=9,C=7 and you can see that scores at the top weren't even close. Say, N=[8,10] C=[6,8] wouldn't make it any worse problem.

Quote ·Reply

ajdonich

April 1, 2021 3:20PM

A few more thoughts from someone who didn't manage to make a proper submission, I found this challenge deceivingly difficult programmatically, in part because manual play wasn’t particularly hard, nevertheless, I had a great time working on it and really appreciated this problem. I hope to continue with it in days to come, maybe explore how DRL handles it. I chose not to look at any MM5 info because generally I participate to practice imagining stuff up rather than optimizing existing solutions. I'm new to competitive programming (and not very competitive), but the MMs I’ve done so far intensely motivate my creative focus, so thank you! I get totally immersed and end up excitedly coding into the middle of the night, night-after-night.

The last day, I started exploring a heuristic that assigned each placed piece a wave function of influence/claim to an area of the board centered on the pieces’ square, but diminishingly extending into neighboring squares. I was trying to create a kind of force hypersurface across the board, making like-colored pieces attract and opposing-colors repel. My prototype started giving some interesting/promising play, but I felt it definitely needed more work and optimization and I just ran out of time.

I would have submitted what I had regardless but used python w/NumPy to rough out a solution knowing I couldn’t submit it that way (at least in my first MM I couldn't, forgive my laziness for not verifying that on this match, maybe it was allowed). My plan was to transcribe my solution into C++ at the end. Anyway, it leads me to inquire, must NumPy be disallowed in MMs? I understand restricting to language StdLibs as a standard, but there’s always some language-specific differences if multiple languages and allowed, and NumPy seems like such a standard extension, does it really create an unfair advantage or what?

Quote · Reply

LEAVE A COMMENT

Comment ...

lines: 1 words: 0 0:0