Kick Start 2022 - Coding Practice with Kick Start Session #1

Analysis: Hex

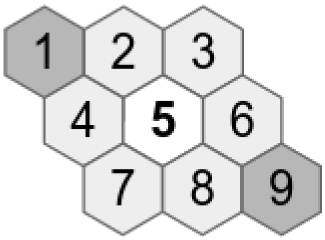
View problem and solution walkthrough video

Given an $\mathbf{N} \times \mathbf{N}$ grid representing a rhombus-shaped board with hexagonal cells, we have to output one of four possible board states: `Impossible`, `Red wins`, `Blue wins`, or `Nobody wins`.

The square input grid does not fully indicate what it means for cells to be connected. For example, consider the cells labeled `1` through `9` in this 3×3 grid:

```
123
456
789
```

Compare the equivalent cells in the game board:

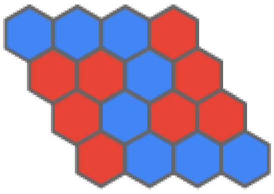


The central cell `5` is not connected to cell `1` or cell `9`. It is connected to cells `2`, `3`, `4`, `6`, `7`, and `8`. This adjacency rule applies throughout the solution.

Another important observation is that the game board is not symmetrical for the Red and Blue players. This is not obvious from the square grid input. For example, consider this 4×4 grid:

```
BBBR
RRBR
RBRR
RBBB
```

The text representation looks symmetrical for Red and Blue. However, the equivalent portion of the game board looks like this:



Notice that the blue stones in the center are connected, but the red stones are not.

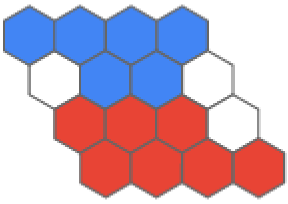
Test set 1

In test set 1, we are given cases with $1 \le \mathbf{N} \le 10$. We can use flood fill to find connected sets of cells. We start by considering the blue cells at the west side of the board. From a given blue cell, we visit each connected cell to see if it is also blue. We keep track of which cells have been visited so that we do not examine the same cell more than once.

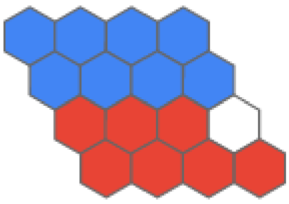
If the flood fill reaches a blue cell at the east side of the board, we know that there is a winning path for the Blue player. If this is not the case, we flood-fill red cells starting from the north side of the board and check whether we reach the south side. This tells us whether the Red player has a winning path. (It is not physically possible for both Blue and Red to have a winning connection, so we do not consider that case.)

Because we traverse the entire $\mathbf{N} \times \mathbf{N}$ board, the time complexity of the flood fill is $O(\mathbf{N}^2)$. However, the solution is not complete yet. We have to think about what would make a board state impossible.

The following state is impossible according to the game rules. By counting the stones, we can see that Red must have placed a stone after Blue won:



Another kind of impossible state is when a player has a winning path, but there is no single stone that could have been placed in the final move to change the board from a non-winning state to a winning state. Consider the following board:



Blue has a winning path. Now consider what state the board was in before Blue's final move. No matter which blue stone we remove from the board, Blue still has a winning path. This tells us that the game did not end according to the rules, so we say that the board is in an `Impossible` state.

Bringing everything together, we can solve test set 1 as follows.

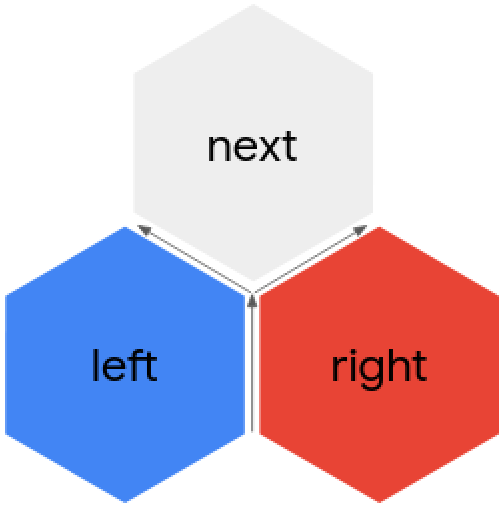
1. Count the stones placed by each player and make sure they took no more moves than allowed, or else the state is `Impossible`.
2. Use flood fill to look for a path of blue cells from the west side of the board to the east side.
3. If there is a blue path, check whether the board state is possible. For each blue stone on the board, temporarily remove the stone and use flood fill again. If there is still a blue path in the temporary state for every single blue stone, the board state is `Impossible`. Otherwise, the state is `Blue wins`.
4. Perform steps 2 and 3 for the Red player, looking for a path of red cells from the north side of the board to the south side.
5. If there is no blue path or red path, the state is `Nobody wins`.

In step 3, for each of $O(\mathbf{N}^2)$ stones, we run flood fill in $O(\mathbf{N}^2)$ time, making a total time complexity of $O(\mathbf{N}^4)$.

Test set 2

In test set 2, we are given cases with $1 \le \mathbf{N} \le 100$. An $O(\mathbf{N}^4)$ solution may be too slow for $\mathbf{N} = 100$.

A more efficient approach involves tracing paths *between* cells. Imagine that we are walking between two cells with a blue stone on our left and a red stone or empty cell on our right:

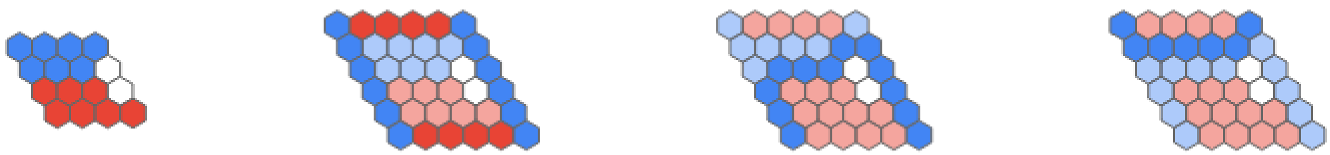


There is a cell ahead of us. If the next cell has a blue stone on it, we go right. Otherwise, we go left. We continue to traverse the grid, maintaining the last moved direction and always turning left if the next cell is not blue. We treat the sides of the board as colored stones for this purpose.

Using this traversal method, we start from the southwest corner and trace a path between cells that always keeps blue stones on the left. We continue until we reach the east side or the north side. No other outcome is possible. If we reach the east side, there are blue stones connecting the west and east sides. Otherwise, there are not.

If there is a connection from west to east, we now trace a path starting from the northwest corner, this time keeping blue stones to the right. The result will be two paths that enclose a connected set of blue stones from west to east.

The following diagrams illustrate this approach. We start by padding the board with colored stones to simplify the path-tracing procedure. You can convince yourself that this does not affect the outcome.



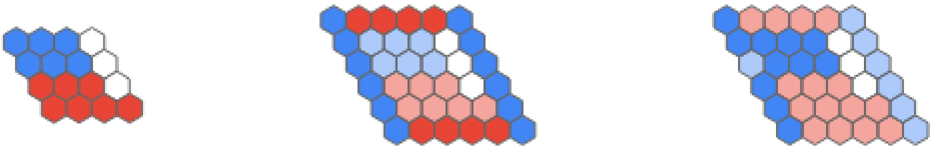
Now we must decide whether Blue has won or the board state is `Impossible`. To do this, we look for a blue stone that Blue could have placed to win the game. If the two paths that we traced from west to east share a stone, we can break the west-to-east connection by removing that stone. This means that Blue could have placed that stone and won the game, so the state is `Blue wins`. If the two paths do not share a stone, there is no single stone that Blue could have placed to win, so the state is `Impossible`.

If there is no blue path, we repeat the above procedure for red paths from north to south. If there are no red paths either, the game state is `Nobody wins`.

In the example above, notice that the two paths share a stone in the top right cell of the original board. By removing this stone, it is clear that both paths are broken, so this must have been the last stone that Blue placed. The state is `Blue wins`.

In the example below, notice that the paths do not share any stones. There is no stone that we can remove to break the connection. As discussed above, this state is `Impossible`.

In the final example, there is no path for Blue or Red. The state is `Nobody wins`:



It takes $O(\mathbf{N}^2)$ time to trace one path and we trace fewer than four paths, so the total time complexity of the path-tracing approach is $O(\mathbf{N}^2)$.