My Solution

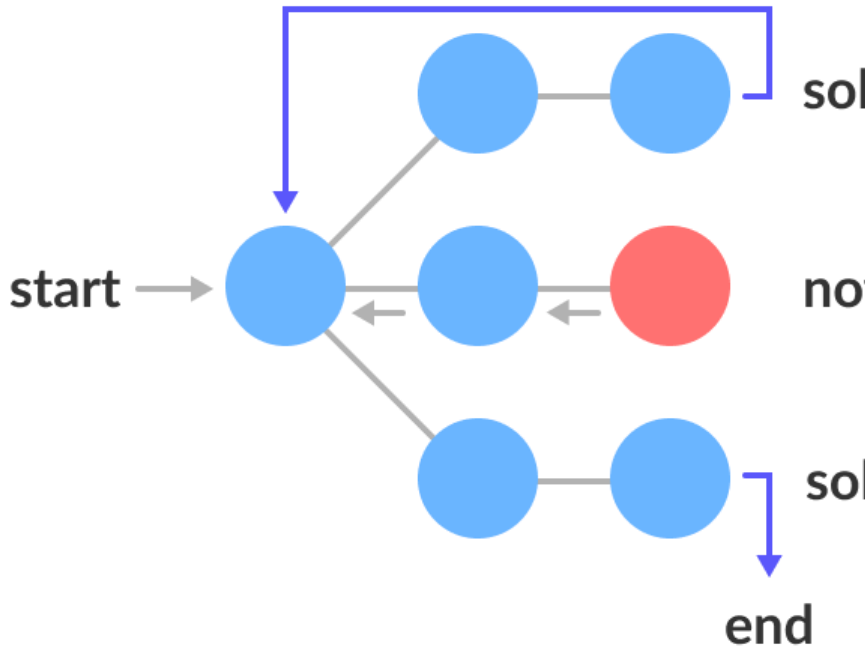
Álvaro Vega Romero

Done for the technical challenge in Aircury

Way proceeded

1. Read patiently the exercise. Make notes
2. Disconnect from the computer & Brainstorm
3. “Implement” a solution on paper
4. Choose Programming Language
5. Manage input and output
6. Implement an algorithm for solving the Puzzle
    1. Iterate over it to get a better solution (New ideas Fixes, Testing, Refactor…)

Algorithm: Backtracking

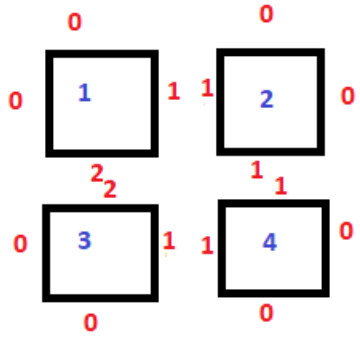


Implementation

1. Create Board & Pieces
2. Try to place every Piece with all the orientation
3. If the piece fits, make a recursive call with the remaining pieces in a different position.
4. If you can’t continue, go back
5. If you have all the pieces placed → Possible Solution
    1. If the solution is not a rotation of the current solution, save it

Easy Example in 2x2 Puzzle

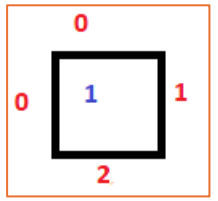
Pieces and Sides



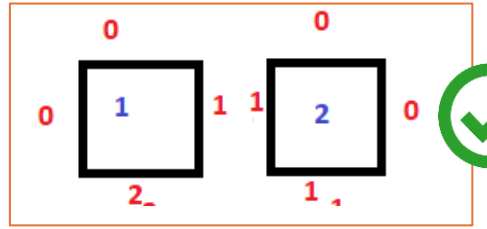
Representation

```
2 2
0 1 2 0
0 0 1 1
2 1 0 0
1 0 0 1
```

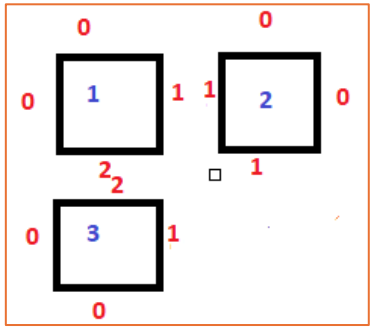
1. Try piece 1 in (0,0)



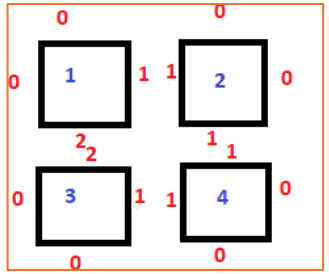
2. Try piece 2 in (0,1)



3. Try piece 3 in (1,0)



4. Try piece 4 in (1,1)



5. ALL PIECES PLACED → Possible Solution

6. Solution is not a rotation of another. SAVE SOLUTION

Improvements

- **More efficiency. Reduce Real Complexity (O(n!))**
- Heuristic on how to solve. Maybe start with corners, then borders…
- Pruning (if we are not able to go into a acceptable solution)
- Avoid the creation of remaining pieces in every recursive call
- Use SET (Data Structure) for the Pieces. (O(1) in default operations)
- …
- Process of determining the solution was good

Time Spent in Task

- Around 6h (Coding) + 1h (Presentantion) Part-time

(but not very efficiently because of the Heat and family)