1. Generate cellToSquares
2. Generate PrevKnownGrids - ( popcount, grid )
3. Set the grid to all 1s.
4. Init recursion: (currentPopcount, currentGrid, currentCell)
   1. If (depth > currBest) – depth is really popcount
      1. Return;
5. If currGrid == one of the PrevKnownGrids -> return with answer from PrevKnownGrids.
6. Check if it is a valid solution.
   1. Yes
      1. Return ( popcount, grid )
   2. No
      1. From HeatMap – get cells with highest squareOverlapFactor and remove the ones that are symmetrically the same.
      2. Once you have that list of cells. Branch to each.
7. Compare results from the branches
8. If all branching is complete: return with the best result.



