

Sudoku Solver, brute force with back tracking

Algorithm 1 Serial backtracking of Sudoku board

```
function VALIDATEBOARD(board, pivot_x, pivot_y)                                ▷ Check validity around pivot.
  if Duplicate numbers in (row of pivot_y in board) then
    return FALSE
  end if
  if Duplicate numbers in (column of pivot_x in board) then
    return FALSE
  end if
  if Duplicate numbers in (subgrid pf pivot_x and pivot_y in board) then
    return FALSE
  end if
  return TRUE
end function

function SOLVEBOARD(board, unassigned_indices, unassigned_n)
  if unassigned_n == 0 then                                                    ▷ Valid solution is found.
    return TRUE
  end if
  index = unassigned_indices[unassigned_n]
  for val in [1, board_side_length] do
    board[index] = val                                                         ▷ Set guess in board.
    if validateBoard(board, index.x, index.y) then                             ▷ Valid guess?
      return solveBoard(board, unassigned_indices, unassigned_n - 1)         ▷ Recursively traverse.
    end if
    board[index] = 0                                                         ▷ Reset guess in board.
  end for
  return FALSE
end function
```
