|  |
| --- |
| var cols = new int[9];  var rows = new int[9];  var sqs = new int[9]; |
| if (board[i][j] != '.') {  int parsed = board[i][j] - '0';  int offset = 1 << parsed;  rows[i] |= offset;  cols[j] |= offset;  sqs[**getSq**(i, j)] |= offset;  } |
|  |
| int minOptionsRow = -1;  int minOptionsCol = -1;  int opts = 0;  int optsSize = 0;  boolean foundOneOpt = false; |
| if (board[i][j] != '.') {  continue;  } |
| int sq = **getSq**(i, j);  var localOpts = ~(cols[j] | rows[i] | sqs[sq]) & 0x3fe;  var localOptsSize = Integer.**bitCount**(localOpts);  if (localOptsSize == 0) {  return false;  } |
| if (minOptionsRow == -1 || optsSize > localOptsSize) {  minOptionsRow = i;  minOptionsCol = j;  opts = localOpts;  optsSize = localOptsSize;  } |
| if (localOptsSize == 1) {  foundOneOpt = true;  break;  } |
| if (foundOneOpt) {  break;  } |
| if (opts == 0) {  return true;  } |
| int i = 1; |
| while (opts > 0) |
| opts = opts >> 1;  if ((opts & 1) == 0) {  ++i;  continue;  } |
| int mask = 1 << i;  board[minOptionsRow][minOptionsCol] = (char) ('0' + i);  rows[minOptionsRow] |= mask;  cols[minOptionsCol] |= mask;  sqs[**getSq**(minOptionsRow, minOptionsCol)] |= mask; |
| if (**solveSudokuImpl**(board, cols, rows, sqs)) {  return true;  } |
| board[minOptionsRow][minOptionsCol] = '.';  rows[minOptionsRow] &= ~mask;  cols[minOptionsCol] &= ~mask;  sqs[**getSq**(minOptionsRow, minOptionsCol)] &= ~mask;  ++i; |
| return false; |
|  |
| return (row / 3) \* 3 + col / 3; |