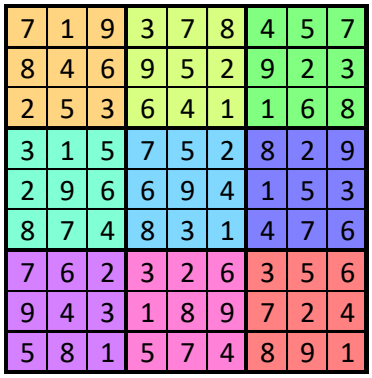
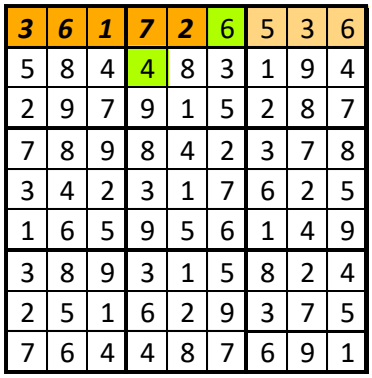
SUDOKU SORTING ALGORITHM



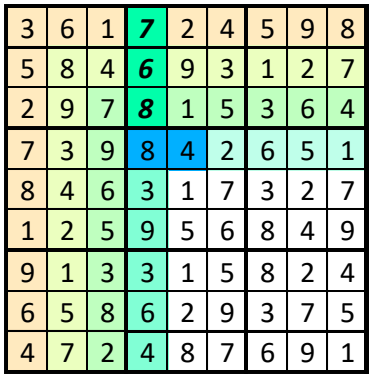
7	1	9	3	7	8	4	5	7
8	4	6	9	5	2	9	2	3
2	5	3	6	4	1	1	6	8
3	1	5	7	5	2	8	2	9
2	9	6	6	9	4	1	5	3
8	7	4	8	3	1	4	7	6
7	6	2	3	2	6	3	5	6
9	4	3	1	8	9	7	2	4
5	8	1	5	7	4	8	9	1

I begin by creating an array of length 81. I create a list with Integer values 1 through 9, and I use the Collections.shuffle() method in the java.util package to randomly assign each Integer's location within each of the 3×3 boxes. So now, all boxes are pre-sorted, but each row and column still contains duplicate values, which is illegal in a Sudoku game. In order to generate a valid Sudoku grid, I approach the problem with a nested loop, attempting 1 of 3 different sorting strategies, starting from row 1, progressing from left to right, to column 1, going top to bottom, then continuing to row 2, to column 2, to row 3, and so on. I named this approach the shrinking square.



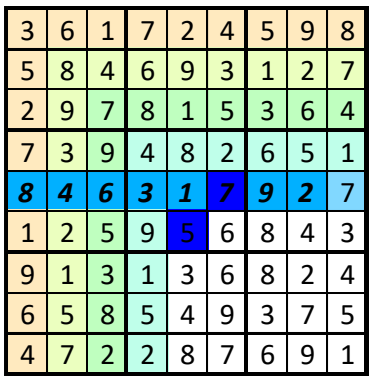
3	***6***	***1***	***7***	***2***	6	5	3	6
5	8	4	4	8	3	1	9	4
2	9	7	9	1	5	2	8	7
7	8	9	8	4	2	3	7	8
3	4	2	3	1	7	6	2	5
1	6	5	9	5	6	1	4	9
3	8	9	3	1	5	8	2	4
2	5	1	6	2	9	3	7	5
7	6	4	4	8	7	6	9	1

As we can see in the first row, our program runs into a duplicate 6. The first strategy attempted is called Box and Adjacent-cell Swap, or **BAS** for short. It works backwards, searching each cell inside the boxes that contains the duplicate values' cells. The program will swap the duplicate number with the number in the cell it finds as long as 1.) the cell is not part of the current row or column being sorted, 2.) the number in that cell has not been registered, and 3a.) the cell being swapped in OR out is not part of a row or column that has been previously sorted unless 3b.) the cell is in the row or column adjacent to the duplicate value.



3	6	1	***7***	2	4	5	9	8
5	8	4	***6***	9	3	1	2	7
2	9	7	***8***	1	5	3	6	4
7	3	9	8	4	2	6	5	1
8	4	6	3	1	7	3	2	7
1	2	5	9	5	6	8	4	9
9	1	3	3	1	5	8	2	4
6	5	8	6	2	9	3	7	5
4	7	2	4	8	7	6	9	1

In the fourth column of the image to the left, we find a duplicate 8. The unsorted cells in this box are 1, 7, 5, and 6; however, the 8 was previously sorted in its row. If the program swapped the 8 for the 1, we would unwittingly unsort the fourth row. To avoid this, the program searches the cells progressively adjacent (down for row searches, and right for column searches). In this example, the adjacent cells contain a 4 and 2. Either would work, so the algorithm takes the first found value, 4. This algorithm does not work retrogressively (up or left), so the 9 cannot be used.



3	6	1	7	2	4	5	9	8
5	8	4	6	9	3	1	2	7
2	9	7	8	1	5	3	6	4
7	3	9	4	8	2	6	5	1
8	***4***	***6***	***3***	***1***	***7***	***9***	***2***	7
1	2	5	9	5	6	8	4	3
9	1	3	1	3	6	8	2	4
6	5	8	5	4	9	3	7	5
4	7	2	2	8	7	6	9	1

Here's another example of the BAS method at work. The algorithm finds a duplicate 7 in the fifth row, eighth column. 8, 4, and 3 of that box were already registered for that row, so the algorithm finds the other instance of the 7 and attempts to swap that another cell. Between the 5 and 6 available, only the 5 works.

Legend:

In the figures found on this document, sorted rows/columns are lightly colored; the row/column actively being sorted is more deeply colored; the dark shaded cells are the numbers that have registered so far; and the odd-colored cells represent the two numbers that are being swapped.

3	6	1	7	2	4	***5***	9	8
5	8	4	6	9	3	***1***	2	7
2	9	7	8	1	5	***3***	6	4
7	3	9	4	8	2	***6***	5	1
8	4	6	3	5	1	***9***	2	7
1	2	5	9	7	6	***8***	4	3
9	1	3	5	4	8	***7***	2	6
6	5	8	1	3	9	3	8	5
4	7	2	2	6	7	4	9	1

Here we see BAS working in the seventh column for the number 3. The algorithm couldn't swap with the 8, 5, 9, or 1, so it keeps working backwards and finds the next instance of 3. The only numbers that can be swapped are adjacent ones because these cells are part of a row that's been previously sorted. The algorithm searches adjacent cells and finds 4, an unregistered number, and that's swapped with the cell containing the 3.

3	6	1	7	2	4	***5***	9	8
5	8	4	6	9	3	***1***	2	7
2	9	7	8	1	5	***4***	6	3
7	3	9	4	8	2	***6***	5	1
8	4	6	3	5	1	***9***	2	7
1	2	5	9	7	6	***8***	4	3
9	1	3	5	4	8	***7***	2	6
6	5	8	1	3	9	***3***	8	5
4	7	2	2	6	7	4	9	1

The next strategy, Preferred Adjacent-cell Swap—or **PAS** for short—swaps with an adjacent cell with a preference for unregistered numbers, but swaps regardless if an unregistered number is there or not when BAS has failed. The algorithm goes to the first occurrence of the duplicated number 4, and swaps with the next available cell, 6. It is already a registered number, so the algorithm starts searching for the other occurrence of the 6 in the column and then swaps with a 5, and the other 5 swaps with 9, etc. The algorithm repeats this swapping process until it finds an unregistered number.

3	6	1	7	2	4	9	5	8
5	8	4	6	9	3	1	7	2
2	9	7	8	1	5	6	4	3
7	3	9	4	8	2	5	6	1
8	4	6	3	5	1	2	9	7
1	2	5	9	7	6	8	3	4
9	1	3	5	4	8	7	2	6
6	7	2	1	3	9	4	8	5
4	5	8	2	6	7	3	1	9

Once in every few thousand grid generations, there is the possibility that the algorithm cannot swap any more unique cells without falling into an infinite loop. After 18 iterations of this inner-loop, there is a fail-safe strategy called Advance and Backtrack Sort (**ABS**) that allows the computer to continue on to sort the next row and column (effectively pulling out the required number) before returning to the current row/column to retry the BAS and PAS strategies.

The above 3 strategies (BAS, PAS, and ABS) are the only ones required to completely sort a Sudoku grid which started with an essentially random placement of Integer values. These steps can produce a perfect Sudoku grid in under seven millionths of a second (7 µs).

This algorithm was developed by Mark F. Graves, Jr. to initially generate a working Sudoku grid for the purpose of creating an entire Sudoku puzzle game generator.

Legend:

In the figures found on this document, sorted rows/columns are lightly colored; the row/column actively being sorted is more deeply colored; the dark shaded cells are the numbers that have registered so far; and the odd-colored cells represent the two numbers that are being swapped.