Homework – 2

Name: P. Karthikeya Sharma

Note: '*Block_puzzle_latest_optimized.py'* is the actual code behind the following implementation.

Procedure:

• The core algorithm used is Depth First Search (DFS).

• Block representation : Each block is represented by a least geometric fit rectangular array, in which 1 is assigned to the cells in which blocks are present and 0 to void cells.

• There are 9 such blocks of different shapes and 27 corresponding rotated blocks (rotated up to 4 times). It is assumed that all the rotations of a block are different, even though the blocks are symmetric. This assumption is for simplicity purpose only and can easily be overrided in the code.

• Block insertion: Blocks are inserted by checking the feasibility of no-overlap by navigating from left to right and top to bottom of the board.

• Various pruning techniques are implemented as listed below under `Pruning' to reduce the problem complexity and faster code run, based on our knowledge about the problem before solving it.

Brief explanation about the Code :

• The '__init__()' function declares the representation of all the (rotations of) blocks/pieces being considered.

• The 'run_algorithm()' function initiates the DFS (of depth 9) with each of the possible starting blocks.

• The blocks are inserted sequentially one-by-one (from top-left to bottom-right) as it searches deep into the tree.

• The 'DFS()' function is an implementation of the Depth First Search in a recursive way until the depth is equal to the no. of blocks. (Note: In the code, the depth starts at 0, hence, the depth of each leaf is 8)

• Various pruning heuristics are implemented by calling them in the 'DFS()' function before next recursive call.

• Other utility functions like 'add_piece()', 'remove_piece()' and 'add_feasible_piece()' are briefly summarized within the code.

Pruning techniques:

• Each DFS is initiated in a parallel multiprocessing manner so that all the 8 cores can be utilized.

• Backtracking is implemented so that only the current path need to be saved i.e, O(m) instead of O(bm).

• By looking at the blocks and their rotations, we know that some of them can never fetch a solution by using them as the first block or in other words, initiating the DFS with them will never give a solution because the first block itself has more than 1 hole which cannot be covered later howsoever. Such blocks are eliminated in the beginning in the 'run_algorithm()' function.

• During the tree search, the code checks if the no. of single '0's surrounded by '1's exceed one and if there are any no. of double '0's surrounded by '1's, as implemented by the 'zero_islands_check()' . Both prune the subsequent tree from the node, if any are encountered.
• To avoid multiple identical solutions, the code checks whether the order of blocks inserted by their feasibility matches that of the actual order given by the path to the node.
• From observation, it is clear that the minimum area of the blocks is 2 and there is only one such piece and for the remaining, the minimum area is 4. Hence, by deduction, after inserting 6 pieces, the first two rows should be filled with a maximum of one 0 among them.
• The code also avoids exploring the node that is already in the path and eliminates any rotation of it before searching through it.

Results:
• The code finds out all the possible **exact** no. of solutions. The multiprocessing has been utilized to run in multiple computers containing 8 cores each.
• Each of the attached 'solution_w_initial_{}.txt' text files contains the solutions, their path, locations of the blocks and the no. of solutions resulted from initiating with the {} block.
• The output of '*solutions_sort.py*' is as follows.

```
- The total no. of solutions in the puzzle with the hole being only in
the respective location are:
 [[ 676  148   28  184   58  448  922]
 [ 368   94  120  334  248  146  218]
 [  70  236  204  218  174  302  124]
 [ 266  308  252   76  310  450  210]
 [ 132  274  184  234  208  330  104]
 [ 372  142  292  486  332  174  538]
 [1010  464  114  228   72  508 1324]]
- The grand total no. of all the solutions to this problem are:
 14714 (sum of all the elements in the above board)
- Hence, every hole has solutions in the puzzle. To look at every
possible solution along with the location of pieces and the resulting
board, please navigate to the 'Detailed every solution to the
puzzle/detailed solutions_to_the_puzzle' folder.
Note: In each solution_w_initial_{}.txt, path corresponds to the order
of pieces (numbered 0 to 8) arranged in the tree. 'Locations of the
pieces arranged' have the first two coordinates indicating the
location of the piece corresponding to the piece no. in the third
coordinate.
```

The following represents the number of solutions starting with the mentioned piece and zero being present only at the corresponding location in the board:

```
The no. of solutions with zero only at corresponding locations
starting with piece 0 is as follows:
 [[ 0  4  0  4  8 20 70]
 [ 0  6  4  6 10 10  0]
 [ 0  0  0 10 14  4  4]
 [12  0  2  2  0 30  4]
 [ 0 10 30 24  6  0  4]
 [52  8 30 22  0  0 20]
 [26 18  0  6  6 20 66]]
The no. of solutions with zero only at corresponding locations
starting with piece 1 is as follows:
 [[4 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]]
The no. of solutions with zero only at corresponding locations
starting with piece 2 is as follows:
 [[ 0  4  8  0  0 38 40]
 [ 0 24  6  0 22  6 10]
 [ 0  0  6  2  8  6  0]
 [ 0 16  4  0  2  6  2]
 [ 0  0  4  4  6  0  6]
 [86  0  2 44 14 14 40]
 [86  8  0  0  4  2  8]]
The no. of solutions with zero only at corresponding locations
starting with piece 3 is as follows:
 [[84  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]]
The no. of solutions with zero only at corresponding locations
starting with piece 4 is as follows:
 [[ 0  8  0  0  0  4 64]
 [ 0  0  0  0 24  0  0]
 [ 0  0  0  8  0 28  0]
 [36 12  0 12 20 16  4]
 [ 0  0  8  8 20  0  0]
```

```
 [ 0 20  4  8  4  0  8]
 [20 12  0  0  0  0 56]]
The no. of solutions with zero only at corresponding locations
starting with piece 5 is as follows:
 [[  0   0   8   0   0 112  28]
 [  0   0   0  32   0   8  12]
 [  0  24   0  36  24   0  32]
 [  0   8   4   4   8   0   0]
 [ 36   0   4   0  12   0  12]
 [ 56   0   0  24  20   0  16]
 [ 40   8  16   8   0   0  68]]
The no. of solutions with zero only at corresponding locations
starting with piece 7 is as follows:
 [[  0   0  12   0   0   0  40]
 [  0   0   0  12   0  16  48]
 [  0   0   8  12  12  32  16]
 [ 88  32   0   4  20  32   8]
 [  4  36   8   8  12  48  20]
 [  4  24  12  12  32   4 124]
 [ 32 116  12  12   0  68 104]]
The no. of solutions with zero only at corresponding locations
starting with piece 8 is as follows:
 [[  0   0   0   0   0  20  12]
 [108  60   0  24   8   8   8]
 [ 12   0   0   8  12   0  28]
 [ 16   4   8   0  12  52   4]
 [ 16  12   0   0  28  16   0]
 [  8   0   0   0   0  12  28]
 [120   0   8  12   0   4  32]]
The no. of solutions with zero only at corresponding locations
starting with piece 9 is as follows:
 [[ 0  0  0 16  2 14 46]
 [26  4  6 14  4 28 12]
 [ 4 20  4  6 14 12  4]
 [ 6  4 14  0 12 12  0]
 [ 0  6 10  6  2 10  2]
 [16  0 20  6  4  4 16]
 [10 22  2  0 22 14 44]]
The no. of solutions with zero only at corresponding locations
starting with piece 10 is as follows:
 [[28  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
```

```
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]]
The no. of solutions with zero only at corresponding locations
starting with piece 11 is as follows:
 [[ 0  0  0  0  0  0 98]
 [62  0 12  6  0 18  0]
 [14 12  6  8 10  0  0]
 [ 0  4  0  2  0 30  8]
 [ 0 10  4  8 10 16  8]
 [22  2 16 14 10  4 10]
 [44 48  8  2  8 20 38]]
The no. of solutions with zero only at corresponding locations
starting with piece 12 is as follows:
 [[316   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]]
The no. of solutions with zero only at corresponding locations
starting with piece 13 is as follows:
 [[  0   0   0   0  16  60  80]
 [  0   0  32  40  32   0   8]
 [  0   8  24  28   0  44   0]
 [ 40  12  36  12  44  40   0]
 [ 12   0   4   8  16 108   0]
 [ 36  52  48  44   8   0  12]
 [ 84  32   8  36   8  12 120]]
The no. of solutions with zero only at corresponding locations
starting with piece 15 is as follows:
 [[ 0 24  0  0  0 36 36]
 [ 0  0  0 24 12 16  0]
 [ 0 16  8  0  0 16  0]
 [24  4  0 12 16  0 12]
 [ 0  4 16  4  8  4  8]
 [ 4  4 32 12 32 12  0]
 [56  8  0 12  8 20 60]]
The no. of solutions with zero only at corresponding locations
starting with piece 16 is as follows:
 [[  0   0   0  56   8  12 112]
```

```
 [  0   0   0  32   4   0  12]
 [ 32   0  44   8  24   0   0]
 [  0   0  36   8  60  56   0]
 [  0  24   0  24   8  44  16]
 [ 36   0  20   0  40   8  92]
 [ 60  80  12  44   0  80  92]]
The no. of solutions with zero only at corresponding locations
starting with piece 19 is as follows:
 [[  0  80   0   0   8  36  72]
 [  0   0  48   4  12  20   8]
 [  0   0  28   8   8  80   0]
 [  0  12  40  12  44  48  16]
 [ 56  16   0  36   8   8  16]
 [  8   8  28  60  28  16  40]
 [  8  20  36   4   0 112 156]]
The no. of solutions with zero only at corresponding locations
starting with piece 21 is as follows:
 [[  0   0   0  92   0  12   0]
 [152   0   0   0   0   0  20]
 [  0   0   0   8  20  12   0]
 [ 20  80  52   0  24  28  24]
 [  0  20   0  20  16  24   8]
 [  0   0   0  60  68   4   0]
 [ 60   0   0  16   8  24  52]]
The no. of solutions with zero only at corresponding locations
starting with piece 22 is as follows:
 [[ 0  0  0  0  0  0 16]
 [ 0  0  0 52  8  0  4]
 [ 0  0  0 24  4  8  0]
 [ 8  0 24  0  0 44  0]
 [ 0 16 36  4  0  8  0]
 [36 12  0  0  0  0  8]
 [48  0  0 24  0  0 44]]
The no. of solutions with zero only at corresponding locations
starting with piece 23 is as follows:
 [[ 0 24  0  0  0 16 36]
 [ 0  0  0  0  0  0  0]
 [ 8  0  0  0  8  0  0]
 [ 0 12  4  0  0 20  4]
 [ 0  0  0  0 24  0  0]
 [ 4  0  4 12  0  8 12]
 [48 20 12  4  0  0 40]]
```

```
The no. of solutions with zero only at corresponding locations
starting with piece 24 is as follows:
 [[ 0  0  0 16  0  8 12]
 [12  0 12  0  0  0 16]
 [ 0 24  4 12  0 16 12]
 [ 8  4  4  8  4  0  0]
 [ 0  0 24 16 16 12  0]
 [ 0  4 24 92 24  8 20]
 [48  0  0 20  0 36 76]]
The no. of solutions with zero only at corresponding locations
starting with piece 25 is as follows:
 [[52  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]]
The no. of solutions with zero only at corresponding locations
starting with piece 26 is as follows:
 [[36  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]]
The no. of solutions with zero only at corresponding locations
starting with piece 28 is as follows:
 [[  0   0   0   0   8  52  24]
 [  8   0   0  24   0  16  36]
 [  0  60  64   8   8  28   8]
 [  8  16   8   0  28  36  32]
 [  8  32   8  20   0  24   0]
 [  4   8  48  40  16   0  28]
 [ 60   4   0   8   8  44 124]]
The no. of solutions with zero only at corresponding locations
starting with piece 32 is as follows:
 [[120   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]
```

```
 [  0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0]]
The no. of solutions with zero only at corresponding locations
starting with piece 34 is as follows:
 [[36  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0]]
The no. of solutions with zero only at corresponding locations
starting with piece 35 is as follows:
 [[  0   4   0   0   8   8 136]
 [  0   0   0  64 112   0  24]
 [  0  72   8  32   8  16  20]
 [  0  88  16   0  16   0  92]
 [  0  88  28  44  16   8   4]
 [  0   0   4  36  32  80  64]
 [160  68   0  20   0  52 144]]
```