**Hooks 9 [June 2023 Jane Street Puzzle]**

Problem: <https://www.janestreet.com/puzzles/hooks-9-index/> (credit to Jane Street)

Solution by Nicholas Patel

I’m sure there are more elegant approaches than what I used to solve this problem.

Regardless, my method was as follows:

Consider the problem as adding L-shapes one by one, starting at the largest L (17 elements). Notice that at any given stage (except the last, trivial stage of adding a 1), there are 4 possible ways to insert the current L (i.e. the corner of the L can be in any corner of the grid remaining). As Ls are added, the grid transitions through a sequence of shrinking squares. Any of the numbers x between 1-9 not yet allocated could be assigned to this L (if x <= #elements in L), but the larger numbers will generally be placed in the larger Ls. This gives us a systematic way of exploring possible solutions via backtracking.

Inserting an L ‘completes’ exactly 1 row and 1 column, so the GCD requirements of those can be checked immediately. In some cases, parts of other rows/columns will also be ‘blocked’ when an L is placed; for example, for an element in the L where no number is placed, part of the column/row through that element might be complete/unchangeable, and thus for the solution to be feasible all the numbers in this blocked part must be divisible by that column/row GCD. A similar check was applied for the continuous path condition. Another check was that all 2x2 squares weren’t all filled, and that the unallocated area of the grid was large enough to physically enter all remaining numbers. These checks prevented the exploration of solutions which we knew would not be fruitful.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 5 | 1 | 6 | 1 | 8 | 1 | 22 | 7 | 8 |
| 55 | 5: {0,0} |  | | | | | | | |
| 1 |  | 9: {1,1} |  | | | | |  |  |
| 6 |  | Complete enumeration | | | | |
| 1 |
| 24 |
| 3 |
| 6 |
| 7 |  | | | | | | 7: {7,7} |
| 2 |  | | | | | | | 8: {8,8} |

I played around with the problem on paper to guess that the above L-inserts (and the corresponding number assignments) were likely correct. Note I could have tried similar L-inserts such as {0,0}, {8,8}, {1,1}, {7,7} or {0,0}, {1,1}, {8,8}, {7,7} but thankfully didn’t need to.

Other suspicions included that the 22 column had a 594 and a 22 or 44, that the 24 row lead with a 96, that there was a sequence of 9s in the second column, that the bottom left had a 5 (and that the top row had 4 5s), and that some of the rows/columns with a GCD of 6 each ended with 78. However, I wasn’t confident enough in all of these being correct (or which ones were correct) so I started by ignoring specific values. Using the above frame for the first 4 L-inserts, my (not very optimised) C++ script solved within a day, generating the solution:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 5 | 1 | 6 | 1 | 8 | 1 | 22 | 7 | 8 |
| 55 | 5 | 5 |  |  |  | 5 | 5 |  |  |
| 1 |  | 9 | 9 | 9 | 9 |  | 9 |  | 8 |
| 6 |  |  | 6 |  | 4 | 4 | 4 | 7 | 8 |
| 1 |  | 9 |  | 4 | 3 |  |  |  | 8 |
| 24 |  | 9 | 6 |  | 3 | 1 | 2 |  |  |
| 3 |  | 9 |  |  | 3 |  | 2 | 7 |  |
| 6 |  | 9 | 6 | 6 | 6 | 6 |  | 7 | 8 |
| 7 |  | 7 |  | 7 |  | 7 |  | 7 |  |
| 2 | 5 | 8 |  | 8 | 8 | 8 |  |  |  |

Answer = 8 x 3 x 3 x 2 x 6 x 6 x 3 x 1 x 1 x 1 x 1 x 1 = **15,552**

Overall thoughts:

- Good introduction to learning C++. Still lots to optimise though, like moving the search for

possible insertion indices to outside the backtracking, to prevent repeat calculations.

- A slightly brutal reminder of the time complexity of complete enumeration.

- A decent amount of ‘heuristic’/manual work needed to reduce the solution space to solve

the problem within a reasonable time, which I didn’t appreciate when I started the

problem (e.g. my first script ran for days without generating the solution).