7-Snake Solution Analysis

25th November 2017

Author: Maylon Pedroso
Email: maylonpedroso@gmail.com

OVERVIEW

7-Snake problem break down and complexity analysis of the solution.

PROBLEM SPECIFICATIONS

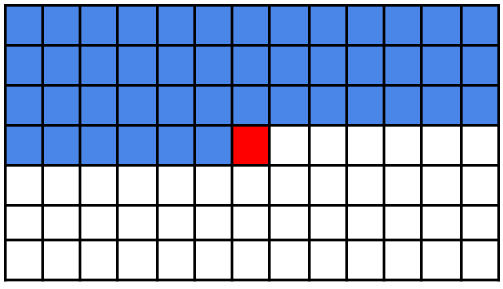
Given a grid of integers in range from 1 to 256, find a pair (two) of 7-Snakes A and B that has the property that the sum of the integers in 7-Snake A is exactly the same as the sum of integers in 7-Snake B.

SOLUTION AND COMPLEXITY ANALYSIS

Snakes generation

For each cell the algorithm will need to find all the snakes that can be created starting from this point. The maximum number of snakes you can create starting from a cell depends exponentially on the snake length (**L**). Without considering the snake restrictions (each cell ci of the snake, can only be adjacent to ci-1 or ci+1), for **L > 1** the number of snakes starting from a cell will never be greater than **4*3^(L-2)**.

For snakes of length 7 the limit is 4*3^5 = 972, and using restrictions the real maximum is 396 (empirically calculated). Half of them would had been created from previous analyzed cells. In the grid below all the snakes created starting at the red cell and ending in the blue region. Will be created previously from the blue cells. Then the number of new snakes created from each cell is 198.



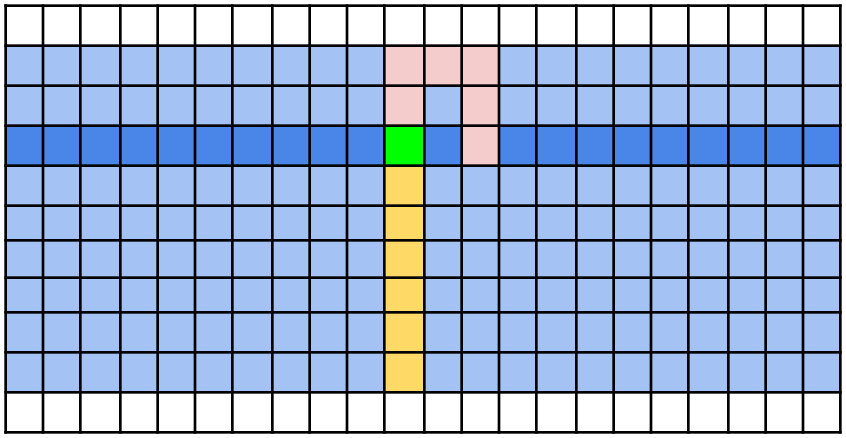
In a grid of **N** cells the maximum number of snakes generated will be fewer than **198*N**. Then the time complexity of snakes generation is **O(198*N)** and dropping the constant **O(N)**.

Same sum pair search

The cells value are in the range from 1 to 256, then the snakes sums will range from **7** to **1786** for a total of **1780** different possible sums. Having an array of this size and storing in each index the list of snakes with that sum, the search of all snakes with a given sum is **O(1)**.

Grid reading and snakes search

The grid have no size restrictions by the specifications, therefore loading the whole grid in memory is not recommended. In order to generate the new snakes starting in a single row are needed only the next and previous 6 rows. Indeed only the previous 2 rows are needed to generate snakes that end at least in the same row they started. In the grid below dark blue is the current row and green is the current cell, light blue are the needed rows, pink and yellow are the longer snakes can go up and down respectively.



As the application will need a maximum of 9 consecutive rows in memory, then the rows can be read in demand. Also the number of possible sums is low and constant (1780) and the number of snakes generated in a row are 198 times the length of the row. Then for longer grids a solution will be found earlier in the processing.

PERFORMANCE

The algorithm was tested with grids up to 100000x100000 (CSV size: 22Gb) and a pair of same sum snakes is always found in less than one second.