Ascent Software Challenge

<http://www.ascentsoftware.eu/>

Playing with 7-Snakes

This problem is a typical search problem much like tic-tac-toe or chess. You must search through a (large) space for possible solutions. The cleverness (in our view) is in the way you enumerate the search space.

Let’s start with some definitions:

Consider the Grid A below. A 7-Snake is a sequence of cells *c1, c2, …,c7* in a grid such that each cell is *adjacent* to the one before it. More formally, for *1 ≤ i < 7*, *ci+1* is adjacent to *ci*. Two cells *a* and *b* are *adjacent* if *b* is to the top, bottom, left, or right of *a*. Given an arbitrary ordering of the cells in a 7-Snake, each cell *ci*, can only be adjacent to *ci-1* or *ci+1*. Note that this exclude cycles.

In Grid A below, the yellow and blue 7-Snakes are valid but the green one is not. This is because cell 7 is not to the top, bottom, left, or right of cell 6. ‘Diagonal’ adjacency is not allowed.

**Grid A**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **1** |  |  |  |  |  |  |  |  |  |  |
| **2** |  | 1 |  |  |  |  |  |  |  | 1 |
| **3** |  | 2 |  |  |  | 1 |  |  |  | 2 |
| **4** |  | 3 | 4 |  |  | 2 | 3 |  | 4 | 3 |
| **5** |  |  | 5 |  |  |  | 4 |  | 5 |  |
| **6** |  |  | 6 |  |  | 6 | 5 |  | 6 |  |
| **7** |  |  | 7 |  |  | 7 |  |  |  | 7 |
| **8** |  |  |  |  |  |  |  |  |  |  |
| **9** |  |  |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |  |  |

**Problem Definition**

The problem is very simple to describe. Given a grid of integers such as Grid B below, 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*.

**Grid B**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **1** | 227 | 191 | 234 | 67 | 43 | 13 | 48 | 211 | 253 | 243 |
| **2** | 36 | 95 | 229 | 209 | 49 | 230 | 46 | 16 | 190 | 49 |
| **3** | 206 | 130 | 85 | 67 | 104 | 93 | 128 | 243 | 38 | 173 |
| **4** | 234 | 82 | 191 | 153 | 170 | 99 | 124 | 60 | 12 | 31 |
| **5** | 192 | 9 | 24 | 127 | 183 | 241 | 139 | 21 | 244 | 66 |
| **6** | 93 | 200 | 66 | 16 | 189 | 42 | 209 | 113 | 215 | 4 |
| **7** | 182 | 141 | 153 | 64 | 229 | 55 | 115 | 139 | 12 | 187 |
| **8** | 133 | 241 | 35 | 255 | 126 | 39 | 110 | 147 | 24 | 241 |
| **9** | 2 | 202 | 191 | 159 | 223 | 128 | 154 | 109 | 6 | 200 |
| **10** | 173 | 44 | 163 | 196 | 159 | 232 | 135 | 159 | 117 | 175 |

Notes:

1. The two 7-Snakes must be distinct. They cannot share cells.
2. In general there may be more than one pair of 7-Snakes with the required property. Your program need only find one pair.
3. If no such pair exists the program should output ‘FAIL’. Otherwise it should output the first pair it finds that has the above property.
4. The solution depends on your ability to *enumerate* the set of all pairs of distinct 7-Snakes in the given grid.
5. In general, the input grid can be any (square) size. Grid B above is just an example of a 10 X 10 grid. The grid should be stored in CSV format on disk and loaded by your solution. This will allow us to test your solution on various test examples. The integers in each cell must range from 1 to 256.