6.851: Advanced Data Structures, Spring 2021

Prof. Erik Demaine, Josh Brunner, Dylan Hendrickson, Yevhenii Diomidov

Problem Set 5

Due: Thursday, March 25, 2021

Problem 5.1 [Walking the Matrix]. Design a **cache-oblivious** data structure that stores a static $N \times N$ matrix and maintains a finger at some cell (i, j) of the matrix, subject to the following operations:

- **teleport**(i', j'): Move the finger (i, j) to the cell at row i' and column j'.
- **move-left**(): Move the finger one step by decrementing the current column j.
- **move-right**(): Move the finger one step by incrementing the current column j.
- **move-up**(): Move the finger one step by decrementing the current row i.
- **move-down**(): Move the finger one step by incrementing the current row i.
- **get**(): Return the value at the current finger.

The number of memory transfers should be $O(1)$ amortized per teleport, $O(1/\sqrt{B})$ amortized per move, and 0 per get.

You can assume that the finger never gets moved out of bounds (invariant: $1 \le i, j \le N$), and that $M \geq c\,B$ for any desired constant c.