A single IntGrid2D object has several fields that are always stored in the same way. These can be discounted when thinking about space complexity because they occupy a constant and relatively small amount of space compared to the “grid” field. Since IntGrid2D’s constructor takes parameters that don’t directly correspond to the size of grid, it’s important to relate the two before thinking about the amount of space in memory grid takes up. IntGrid2D’s parameters are as follows:

- X-coordinate of the upper left corner (ulX).
- Y-coordinate of the upper left corner (ulY).
- X-coordinate of the lower right corner (lrX).
- Y-coordinate of the lower right corner (lrY).

These parameters can be converted to the width and height of grid like so:

- Width = lrX - ulX + 1
- Height = ulY - lrY + 1

And, likewise, the total amount of memory that must be allocated because of the grid field is:

Width * Height * 2 (bytes per char) + Height * 32 (for length values of arrays in grid) + 32 (for length value of grid)

For this reason, I believe that the space complexity of an IntGrid2D object varies linearly in terms of Width * Height. This means that the function that determines the size of an IntGrid2D object is an O(n) function.