

Analysis Quiz

ANSWER KEY

Do a runtime analysis of the constructor for the `TestRandomArray` class. Label each line with its Big- O run time. Give the total for any loops (and their bodies) on the closing brace of the loop. Give the total for the constructor on the closing brace of the constructor. If make the assumption that other functions run in $O(1)$ time. You should use the rules we discussed in class to do the analysis, but show **all** work.

Hint: Don't neglect the header of a for loop. How much work is being done there?

```
class TestRandomArray {

    private boolean [][] mBoard; // Good for a two-dimensional world
    private int mDimension;      // the size of the board

    public TestRandomArray(int dimension, double probLife) {

        mDimension = dimension;
        // O(1)

        mBoard = new boolean[mDimension][mDimension];
        // O(1)

        for (int i = 0; i < mBoard.length; i++) {
            // n repetitions,
            // O(1) init, length lookup, < operation, and update
            // O(1)+O(1)+O(1)+O(1) = O(4) = O(1)

            for (int j = 0; j < mBoard[i].length; j++) {
                // n repetitions,
                // O(1) init, array lookup, length lookup, < operation, and update
                // O(1)+O(1)+O(1)+O(1)+O(1) = O(5) = O(1)

                mBoard[i][j] = (Math.random() < probLife);
                // assume that random() is O(1)
                // O(1) assign, random(), < operation, and array lookup
                // O(1)+O(1)+O(1)+O(1) = O(4) = O(1)

            }
            // body: O(1)+O(1) = O(2) = O(1)
            // loop: n*O(1) = O(n)

        }
        // body: O(1)+O(n) = O(n+1) = O(n)
        // loop: n*O(n) = O(n*n) = O(n^2)

    }
    // constructor: O(1)+O(1)+O(n^2) = O(n^2+2) = O(n^2)
}

```