

Lucas Manker
Cosc 3020
Lab5
10/9/17

My implementation of the function parses over every matrix element to find edges. I accomplished this using a two for loops, with the outer loop parsing through rows, while the inner parses through columns. This is the most "brute-force" method, and it could probably be accomplished in a more efficient way Javascript's `elementOf` function.

A nested for loop usually has a time complexity of $T(n) = \theta(n^2)$

My implementation follows this assumption as it must parse through n rows and n columns where n is the number of vertices. This is the case because each row of a matrix represents the presence of an edge between two vertices. So my time complexity is:

$T(n) = \theta(v^2)$ where v = number of vertices

This implementation was made with the assumption that the graph was directional, but if it were to be changed to parse through undirectional graphs the time complexity would remain the same. This is because while we are still parsing one half the entries (only above or below the diagonal of the matrix), asymptotic notation requires the elimination of all constants. So the time complexity would be as follows:

$T(n) = \theta\frac{1}{2}(v^2) = \theta(v^2)$ where v = the number of vertices.