Matrix to List

10-16-19

In order to convert an adjacency matrix to an adjacency list, you need to implement a function that uses a double for loop structure. The outermost for loop iterates for a number of times equal to the number of vertices. The innermost for loop iterates for a number of times also equal to number vertices. This is the case because in an adjacency matrix, each vertice is represented by an array with a length equal to number of vertices. Thus, the runtime complexity is:

$$T(V, E) = |V|^2 + |E| + 1$$

$$T(V) \in O(|V|^2)$$

It depends on the number of vertices because an adjacency matrix is implimented with a 2D array with a length equal to number of vertices, where each element is also an array with length of number of vertices.

List to Matrix

10-16-19

In order to convert a list to a matrix, you would need to implement a function that looks at each key, being one of the vertices, and for each key looks at an array of the vertices it's connected to. This is equivelant to the number of edges times vertices. Thus, the runtime complexity is:

$$T(V, E) = |V||E|$$

$$T(V, E) \in O(|V||E|)$$

It depends on number of edges and vertices because the outer for loop in our function iterate over the number of vertices. For each vertex we look at, we then look at the edges it has. Thus, we end up with a function, $T(V, E)$, equal to the cardinality of V times the cardinality of E. This function would then be in the big Oh of: $|V||E|$