* Graph Representations:
  + Adjacency Matrix:
    - Two-dimensional array.
      * # of rows = # of vertices (nodes)
      * # of columns = # of vertices (nodes)
    - From row number to column number, indicates an edge. In other words, if node x has an edge that points to node y, then row x will have a 1 in its column y.
    - Binary values inside of 2d array. 1 means the edge exists, o means the edge doesn’t exist.
    - Space required: O(|V|2), where |V| is the number of vertices (nodes).
    - Efficient for dense graphs
  + Adjacency List:
    - One dimensional array, where each array entry contains a linked list. The linked list will contain a list of the nodes that the current node (the node that corresponds with the array location) can point to.
    - Space required: O(|V| + |E|), where |V| is the number of vertices and |E| is the number of edges.
      * Note: This IS addition, not multiplication.
    - Efficient for sparse graphs
    - Algorithms visiting each neighbor of each vertex is more efficient on Adjacency List, especially for sparse graphs.