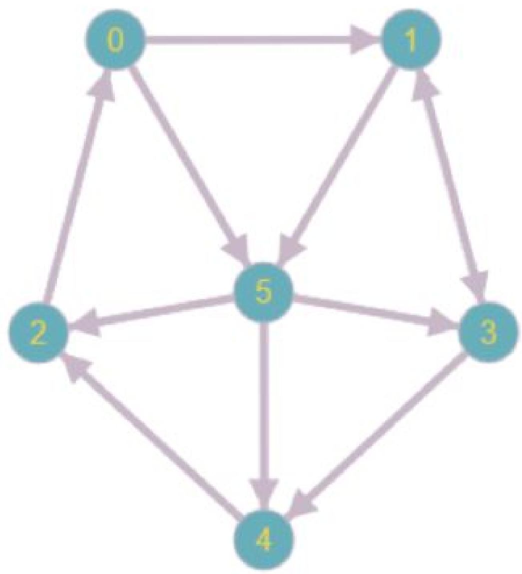
NAMES: Bingcong li, Dakota Krogmeier, David Easton
Worksheet 18: Work with the other students in your Zoom breakout room to complete this worksheet. You should only submit one paper for the group.

The reverse of a directed graph, $G = (V, E)$, is another directed graph $G^R = (V, E^R)$ on the same vertex set, but with all edges reversed.



1. If G is the graph shown above, represent both G and G^R using an **adjacency list.** (2 points)

G

Vertex	List of Adjacent Vertices
0	1, 5
1	5,3
2	0
3	1, 4
4	2
5	2, 4, 3

G^R

Vertex	List of Adjacent Vertices
0	2
1	0,3
2	4, 5
3	5, 1
4	5, 3
5	1, 0

2. Give an efficient algorithm for reversing a graph if the graph is represented using an **adjacency list**. NOTE: You must use this representation of a graph for this problem. (4 points)

KEY IDEA:

- Create a new empty adjacency list of identical size to G.
- Iterate through every vertex v in original graph G
    - Every time we detect a directed edge (v,u) for that vertex v in the adjacency list, add a new directed edge (u,v) in the new adjacency list on the same vertex.

Algorithm ReverseGraph(*G*):

Input: Adjacency list G = (V, E), representing all vertices and their directed edges in G

- V = set of all vertices in G
- E = set of all edges for each vertex in G[V]

Output: Adjacency list G^R, representing the reverse of G.

$G^R = (V, E^R)$, $E^R = (v, u) : (u, v) \in E$

PROCESS:

G^R = [|V|][|E|]   # Create new adjacency list for reversed graph, same size as G

for i ← 1 to number of vertices in G   # Iterate through every vertex in G

    for j ← 1 to number of vertices in G   # check edges of vertex i in V

        if G[i][j] != 0:   # if an edge exists between v, u

            G^R[j][i] = G[i][j]   # create an edge u,v in G^R

Use adjacency list repr

return G^R

(-1.5)