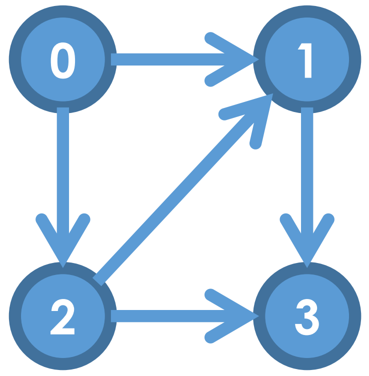
Finding Two-Hop Neighbors

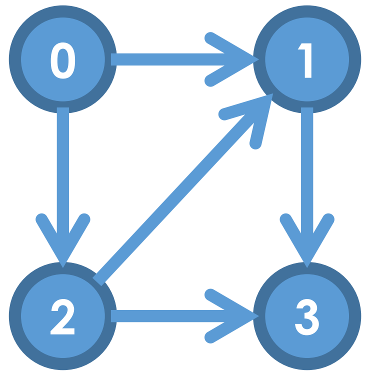
By the end of this video you will be able to…

- Implement the algorithms for finding two-hop neighbors in both an adjacency list and an adjacency matrix representation
- Explain how matrix multiplication can be used to find the two-hop neighbors in a graph represented as an adjacency matrix



$V = \{0, 1, 2, 3\}$

Assignment: Find all two-hop neighbors from given vertex



V = {0, 1, 2, 3}

0 → {1,2}

1 → {3}

2 → {1,3}

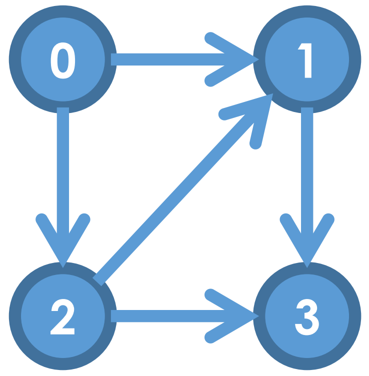
3 → null

Assignment: Find all two-hop neighbors from given vertex (Adjacency List)

The one-hop neighbors are easy to get!

```
public List<Integer> getNeighbors(int v) {
    return new ArrayList<Integer>(adjListsMap.get(v));
}
```

Assignment: Find all two-hop neighbors from given vertex (Adjacency List)



V = {0, 1, 2, 3}

0 → {1,2}

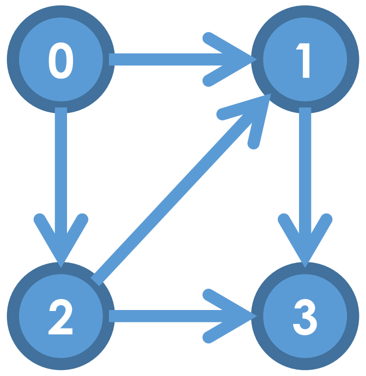
1 → {3}

2 → {1,3}

3 → null

```
public List<Integer> getDistance2 (int v) {
    List<Integer> twoHop = new ArrayList<Integer>();
    List<Integer> oneHop = adjListsMap.get(v);
    // Loop through oneHop and get the neighbors of each...
}
```

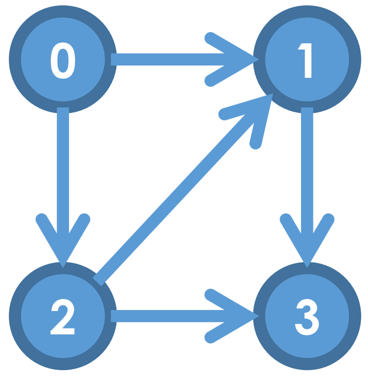
Assignment: Find all two-hop neighbors from given vertex (Adjacency Matrix)



V = {0, 1, 2, 3}

0	1	1	0
0	0	0	1
0	1	0	1
0	0	0	0

```
public List<Integer> getNeighbors(int v) {
    List<Integer> neighbors = new ArrayList<Integer>();
    for (int i = 0; i < getNumVertices(); i ++) {
        if (adjMatrix[v][i] > 0) {
            neighbors.add(i);
        }
    }
    return neighbors;
}
```



V = {0, 1, 2, 3}

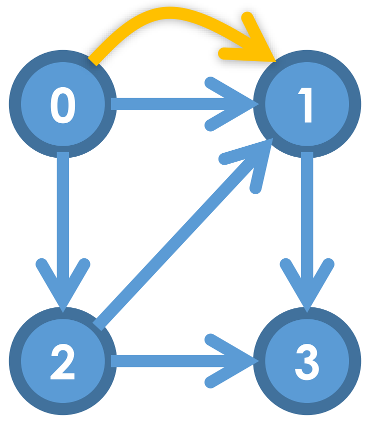
0	1	1	0
0	0	0	1
0	1	0	1
0	0	0	0

Assignment: Find all two-hop neighbors from given vertex (Adjacency Matrix)

```
public List<Integer> getNeighbors(int v) {
    List<Integer> neighbors = new ArrayList<Integer>();
    for (int i = 0; i < getNumVertices(); i ++) {
        for (int j=0; j< adjMatrix[v][i]; j ++) {
            neighbors.add(i);
        }
    }
    return neighbors;
}
```

What does this change do?

Assignment: Find all two-hop neighbors from given vertex (Adjacency Matrix)



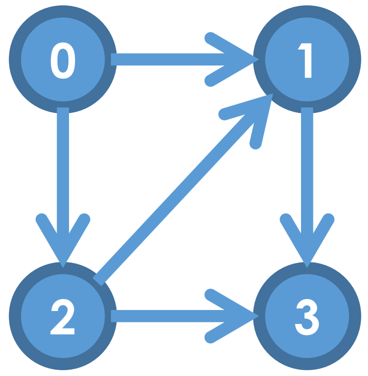
V = {0, 1, 2, 3}

0	2	1	0
0	0	0	1
0	1	0	1
0	0	0	0

```
public List<Integer> getNeighbors(int v) {
    List<Integer> neighbors = new ArrayList<Integer>();
    for (int i = 0; i < getNumVertices(); i ++) {
        for (int j=0; j< adjMatrix[v][i]; j ++) {
            neighbors.add(i);
        }
    }
    return neighbors;
}
```

What does this change do?

Assignment: Find all two-hop neighbors from given vertex (Adjacency Matrix)



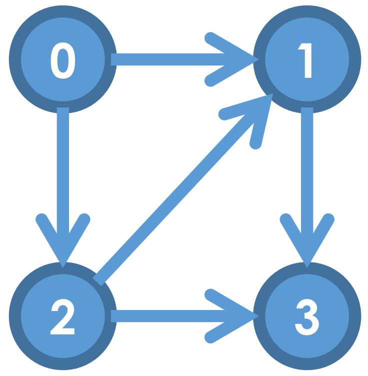
$$\begin{tikzcd} 0 \arrow[r] \arrow[d] & 1 \arrow[d] \\ 2 \arrow[ur] \arrow[r] & 3 \end{tikzcd}$$

V = {0, 1, 2, 3}

0	1	1	0
0	0	0	1
0	1	0	1
0	0	0	0

```
public List<Integer> getNeighbors(int v) {
    List<Integer> neighbors = new ArrayList<Integer>();
    for (int i = 0; i < getNumVertices(); i ++) {
        for (int j=0; j< adjMatrix[v][i]; j ++) {
            neighbors.add(i);
        }
    }
    return neighbors;
}
```

Assignment: Find all two-hop neighbors from given vertex (Adjacency Matrix)



V = {0, 1, 2, 3}

0	1	1	0
0	0	0	1
0	1	0	1
0	0	0	0

```java
public List<Integer> getDistance2(int v) {
    List<Integer> twoHop = new ArrayList<Integer>();
    for (int i = 0; i < getNumVertices(); i ++) {
        for (int j=0; j< adjMatrix[v][i]; j ++) {
            // Instead of adding i directly, add the
            // neighbors of i
        }
    }
    return neighbors;
}
```