

Adjacency List

Graphs can be represented in different ways. Here we describe one way, which is called an *adjacency list*. An adjacency list is essentially a bulleted list where the left side is the node and the right side lists all the other nodes it's connected to. Below is a representation of an adjacency list.

```
Node1: Node2, Node3
Node2: Node1
Node3: Node1
```





Above is an undirected graph because Node1 is connected to Node2 and Node3, and that information is consistent with the connections Node2 and Node3 show. An adjacency list for a directed graph would mean each row of the list shows direction. If the above was directed, then Node2: Node1 would mean there the directed edge is pointing from Node2 towards Node1. We can represent the undirected graph above as an adjacency list by putting it within a JavaScript object.

```
var undirectedG = {
  Node1: ["Node2", "Node3"],
  Node2: ["Node1"],
  Node3: ["Node1"]
};
```

This can also be more simply represented as an array where the nodes just have numbers rather than string labels.

```
var undirectedGArr = [
  [1, 2], // Node1
  [0],    // Node2
  [0]     // Node3
];
```

Create a social network as an undirected graph with 4 nodes/people named James, Jill, Jenny, and Jeff. There are edges/relationships between James and Jeff, Jill and Jenny, and Jeff and Jenny.

	<code>undirectedAdjListeap</code> should only contain four nodes.
	There should be an edge between Jeff and James.
	There should be an edge between Jill and Jenny.
	There should be an edge between Jeff and Jenny.