App Academy

Adjacency Lists

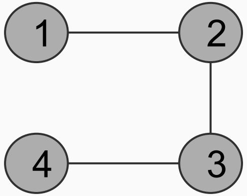
Adjacency Lists

- Working with adjacency lists is actually exactly like matrices, and in some ways they're even easier.
- We do not need to worry about going out of bounds anymore.
- We still need to keep track of visited nodes, especially if we are working with an undirected graph.
- Demo
    - All Paths From Source to Target
      Shortest Path

What if I need to construct the adjacency list myself?

- Sometimes we aren't given an adjacency list, and instead we are given a list of edges.
    - Example:
        - edges = [[1,2], [2,3], [3,4]]
        - adjList ={1: [2],
          2: [1,3]
          3: [2,4]
          4: [3]}

```
def constructGraph(edges):
  graph = defaultdict(list)
  for edge in edges:
        src = edge[0]
        dst = edge[1]
        graph[src].append(dst)
        graph[dst].append(src)
  return graph;
```



Questions?

Let’s practice!

- Review
    - Shortest Path with Alternating Colors
    - Is Graph Bipartite?
- Bonus
    - Course Schedule
    - Find Eventual Safe States
    - Course Schedule II
    - Shortest Bridge