

Problem: Implement a graph.

Constraints

- Is the graph directed?
 - Implement both
- Do the edges have weights?
 - Yes
- Can the graph have cycles?
 - Yes
- If we try to add a node that already exists, do we just do nothing?
 - Yes
- If we try to delete a node that doesn't exist, do we just do nothing?
 - Yes
- Can we assume this is a connected graph?
 - Yes
- Can we assume the inputs are valid?
 - Yes
- Can we assume this fits memory?
 - Yes

Test Cases

Input:

- `add_edge(source, destination, weight)`
- `graph.add_edge(0, 1, 5)`
- `graph.add_edge(0, 5, 2)`
- `graph.add_edge(1, 2, 3)`
- `graph.add_edge(2, 3, 4)`
- `graph.add_edge(3, 4, 5)`
- `graph.add_edge(3, 5, 6)`
- `graph.add_edge(4, 0, 7)`
- `graph.add_edge(5, 4, 8)`

`graph.add_edge(5, 2, 9)`

Result:

- `source` and `destination` nodes within `graph` are connected with specified `weight`.

Note:

- The Graph class will be used as a building block for more complex graph challenges.

Algorithm

Node

Node will keep track of its:

- id
- visit state
- incoming edge count (useful for algorithms such as topological sort)
- adjacent nodes and edge weights

add_neighbor

- If the neighbor doesn't already exist as an adjacent node
 - Update the adjacent nodes and edge weights
 - Increment the neighbor's incoming edge count

Complexity:

- Time: $O(1)$
- Space: $O(1)$

remove_neighbor

- If the neighbor exists as an adjacent node
 - Decrement the neighbor's incoming edge count
 - Remove the neighbor as an adjacent node

Complexity:

- Time: $O(1)$
- Space: $O(1)$

Graph

Graph will keep track of its:

- nodes

add_node

- If node already exists, return it
- Create a node with the given id
- Add the newly created node to the collection of nodes

Complexity:

- Time: $O(1)$
- Space: $O(1)$

add_edge

- If the source node is not in the collection of nodes, add it
- If the dest node is not in the collection of nodes, add it
- Add a connection from the source node to the dest node with the given edge weight

add_undirected_edge

- Call add_edge
- Also add a connection from the dest node to the source node with the given edge weight

Complexity:

- Time: $O(1)$
- Space: $O(1)$
-