1. Write the function \textit{count_edges}(G) that receives a graph \(G\) represented as an adjacency list and returns the number of edges in \(G\). Make sure your function works for directed and undirected graphs.

2. Write the function \textit{highest_cost_edge}(G) that receives a weighted directed graph \(G\) represented as an adjacency list and returns a list of length 3 representing the highest-cost edge in \(G\) using the format \([\text{source}, \text{destination}, \text{weight}]\).

3. The out-degree of a vertex \(v\) in a directed graph \(G=(V,E)\) is the number of edges going out from \(v\) in \(G\). Write the function \textit{out_degrees}(G) that receives a graph \(G\) represented as an adjacency list and returns a list of length \(|V|\) containing the out-degrees of the vertices in \(V\).

4. The in-degree if a vertex \(v\) in a directed graph \(G=(V,E)\) is the number of edges going into \(v\). Write the function \textit{in_degrees}(G) that receives a graph \(G\) represented as an adjacency list and returns a list of length \(|V|\) containing the in-degrees of the vertices in \(V\).

5. Write the function \textit{delete_edge}(source,dest) to be included in the graph\_AL class that removes the edge going from \textit{source} to \textit{dest} in the graph. If there is no edge going from \textit{source} to \textit{dest} your function should display an error message. Make sure your function works for directed and undirected graphs.