

Implement Page Rank Algorithm. (Use python or beautiful soup for implementation).

```
In [2]: import numpy as np

# Step 1: Define the directed graph as an adjacency List
graph = {
    'A': ['B', 'C'],
    'B': ['C'],
    'C': ['A'],
    'D': ['C'],
    'E': ['A', 'D']
}

# Step 2: Calculate PageRank
def calculate_pagerank(graph, d=0.85, num_iterations=100):
    # Initialize PageRank
    num_nodes = len(graph)
    pagerank = {node: 1 / num_nodes for node in graph}

    for _ in range(num_iterations):
        new_pagerank = {}
        for node in graph:
            # Base rank for the node
            new_rank = (1 - d) / num_nodes
            # Add rank from each page linking to this node
            for other_node in graph:
                if node in graph[other_node]:
                    new_rank += d * (pagerank[other_node] / len(graph[other_node]))
            new_pagerank[node] = new_rank
        pagerank = new_pagerank

    return pagerank

# Step 3: Run the PageRank algorithm and print the results
if __name__ == "__main__":
    pagerank_result = calculate_pagerank(graph)
    for node, rank in pagerank_result.items():
        print(f"Page: {node}, Rank: {rank:.4f}")
```

Page: A, Rank: 0.3642

Page: B, Rank: 0.1848

Page: C, Rank: 0.3782

Page: D, Rank: 0.0428

Page: E, Rank: 0.0300

In [ ]: