Name: Kushal Kishor Shankhapal

Group A, Assignment 2: Single Pass Algorithm

Problem Statement:

Implement Single-pass Algorithm for clustering of files.

SPC.cpp

```
#include <iostream>
    #include <vector>
    #include <set>
    #include <string>
    #include <cmath>

    using namespace std;

    // Compute Dice Coefficient between two sets
    double diceCoefficient(const set<string> &A, const set<string> &B) {
        int intersection_size = 0;
        for (const auto &elem : A) {
            if (B.find(elem) != B.end()) {
                intersection_size++;
            }
        }
        return (2.0 * intersection_size) / (A.size() + B.size());
    }

    // Single Pass Clustering
    void singlePassClustering(const vector<set<string>> &documents, double
    threshold, vector<vector<int>> &clusters) {
        for (int i = 0; i < (int)documents.size(); i++) {
            bool assigned = false;
            for (int c = 0; c < (int)clusters.size(); c++) {
                // Compare with cluster representative (first doc in cluster)
                int repIndex = clusters[c][0];
                double sim = diceCoefficient(documents[i], documents[repIndex]);
                if (sim >= threshold) {
                    cout << i + 1 <<": Added into existing Cluster: Cluster "<< c <<
    ", repIndex " << repIndex << ", Threshold: " << threshold << "\n";
                    clusters[c].push_back(i);
                    assigned = true;
                    break;
                }
            }
            if (!assigned) {
                cout << i + 1 <<": Created New Cluster, Threshold: " << threshold <<
    "\n";
                clusters.push_back({i});  // Create new cluster
            }
        }
    }

    int main() {
        // Document representatives (each document as a set of tokens)
        vector<set<string>> documents = {
            {"data", "mining", "clustering", "algorithm"},    // Doc 0
            {"data", "clustering", "analysis", "method"},     // Doc 1
            {"machine", "learning", "algorithm", "model"},    // Doc 2
            {"mining", "data", "model", "analysis"},          // Doc 3
            {"graph", "network", "clustering", "algorithm"}   // Doc 4
        };

        double threshold = 0.50;
        vector<vector<int>> clusters;
```

```
    singlePassClustering(documents, threshold, clusters);

    // Output clusters
    cout << "Clusters formed with threshold = " << threshold << ":\n";
    for (int i = 0; i < (int)clusters.size(); i++) {
        cout << "Cluster " << i + 1 << ": ";
        for (int docID : clusters[i]) {
            cout << docID + 1 << " ";  // Document numbering from 1
        }
        cout << "\n";
    }

    return 0;
}
```

Output:

```
1: Created New Cluster, Threshold: 0.5
2: Added into existing Cluster: Cluster 0, repIndex 0, Threshold: 0.5
3: Created New Cluster, Threshold: 0.5
4: Added into existing Cluster: Cluster 0, repIndex 0, Threshold: 0.5
5: Added into existing Cluster: Cluster 0, repIndex 0, Threshold: 0.5
Clusters formed with threshold = 0.5:
Cluster 1: 1 2 4 5
Cluster 2: 3
```