Semester	T.E. Semester V – Computer Engineering
Subject	Data Warehousing and Mining
Subject Professor In-charge	Prof. Kavita Shirsat
Assisting Teachers	Prof. Kavita Shirsat
Laboratory	M-313A

Student Name	Vibodh Bhosure	
Roll Number	20102A0032	
Grade and Subject Teacher’s Signature		

Experiment Number	06	
Experiment Title	To implement Agglomerative Hierarchical based clustering for a given set of data points and drawing its dendrogram	
Resources / Apparatus Required	Hardware: Computer system	Software: Python

Description

In data mining and statistics, hierarchical clustering analysis is a method of cluster analysis that seeks to build a hierarchy of clusters i.e. tree-type structure based on the hierarchy. Also known as bottom-up approach or hierarchical agglomerative clustering (HAC). A structure that is more informative than the unstructured set of clusters returned by flat clustering. This clustering algorithm does not require us to prespecify the number of clusters. Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.

Algorithm:

```
given a dataset (d1, d2, d3, ....dN) of size N
# compute the distance matrix
for i=1 to N:
  # as the distance matrix is symmetric about
  # the primary diagonal so we compute only lower
  # part of the primary diagonal
  for j=1 to i:
    dis_mat[i][j] = distance[di, dj]

each data point is a singleton cluster

repeat

  merge the two cluster having minimum distance

  update the distance matrix

until only a single cluster remains
```

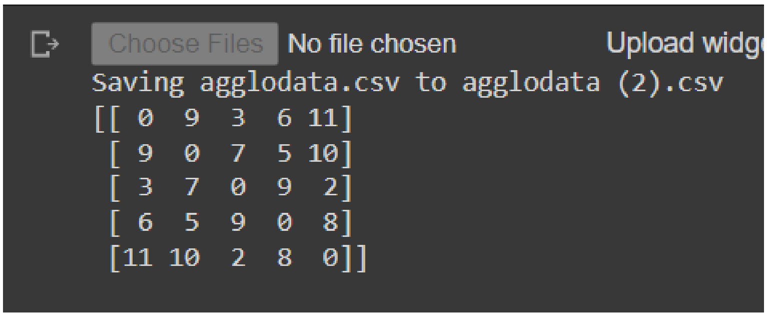
Program

```
# -*- coding: utf-8 -*-
"""Agglomerative.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1XOAMbRJuXMwgCJqKwBwwfCi8GoxrJNCS
"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage

from google.colab import files
uploaded = files.upload()
df = pd.read_csv('agglodata.csv', sep=',', header=None)
print(df.values)
arr = np.array(df.values)

#X = np.array([[0,9,3,6,11],[9,0,7,5,10],[3,7,0,9,2],[6,5,9,0,8],[11,10,2,8,0],])
X = np.array(df.values)

import matplotlib.pyplot as plt

labels = range(1, 6)
plt.figure(figsize=(10, 3))
plt.subplots_adjust(bottom=0.1)
plt.scatter(X[:,0],X[:,1], label='True Position')

for label, x, y in zip(labels, X[:, 0], X[:, 1]):
    plt.annotate(
        label,
        xy=(x, y), xytext=(-3, 3),
        textcoords='offset points', ha='right', va='bottom')
plt.show()
```

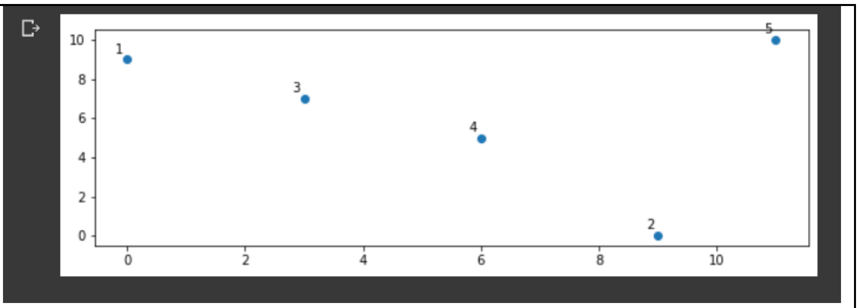
```
linked = linkage(X, 'single')

labelList = range(1, 6)

plt.figure(figsize=(10, 3))
dendrogram(linked,
            orientation='top',
            labels=labelList,
            distance_sort='descending',
            show_leaf_counts=True)
plt.show()

linked = linkage(X, 'complete')

labelList = range(1, 6)

plt.figure(figsize=(10, 3))
dendrogram(linked,
            orientation='top',
            labels=labelList,
            distance_sort='descending',
            show_leaf_counts=True)
plt.show()

linked = linkage(X, 'average')

labelList = range(1, 6)

plt.figure(figsize=(10, 3))
dendrogram(linked,
            orientation='top',
            labels=labelList,
            distance_sort='descending',
            show_leaf_counts=True)
plt.show()
```

Output

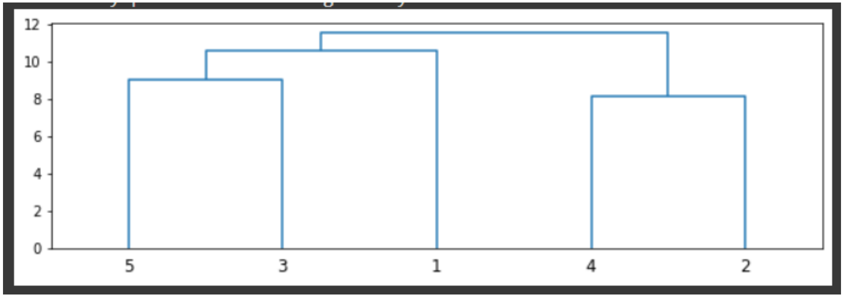
Data Points –



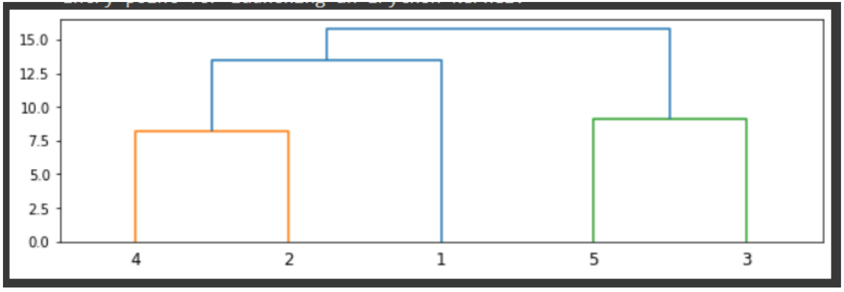
```
Saving agglodata.csv to agglodata (2).csv
[[ 0  9  3  6 11]
 [ 9  0  7  5 10]
 [ 3  7  0  9  2]
 [ 6  5  9  0  8]
 [11 10  2  8  0]]
```



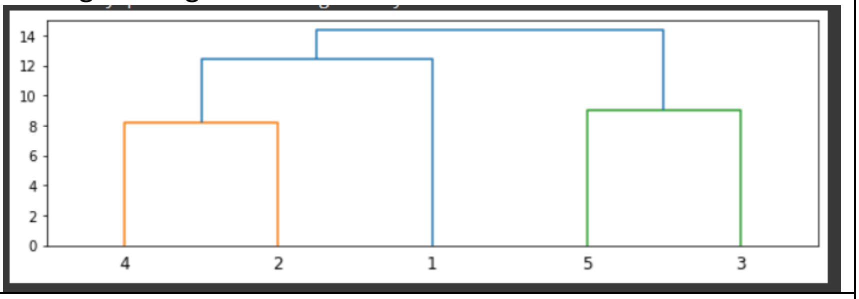
Single Linkage –



Complete Linkage –



Average Linkage –



Conclusion:

Hence, an input of distance between data points was taken from a csv file. Single, complete, and average linkage was found for the data points.