CSE5523 **Student Name:** _____
Fall 2020 **Student ID:** _____
Homework 6
Due date: Sunday, Nov 22, 11:59pm

Grade Table (for TA/grader use only)

Question	Points	Score
1	5	
2	5	
3	5	
4	100	
5	10	
Total:	125	

You will be given an input file and you will program to implement from scratch an bisecting clustering algorithm using the input file. The input file data.txt has n lines, and in each line there are m numbers, delimited by an empty space. Each of the n lines has a data point of m features, and each of the m numbers in a line is the value of that data point on the corresponding feature. The data in this file is represented as a matrix $A \in R^{n \times m}$.

You will implement the bisecting algorithm for clustering data points. You need to output your solution into an output file. The i-th line in the output file has exactly **one number** denoting the cluster index (C-indexing style) of the i-th data point in the input file.

To implement the algorithm, you have to first cluster all data points into two clusters using k-means algorithm (actually, 2-means). For each of the resulted clusters, you need to decide whether you need to further cluster its data points into 2 clusters or not based on some criteria (will be discussed later). If the cluster needs to be further clustered, you will again use k-means to cluster the data points in the cluster into two clusters. You recursively do the above bisecting clustering on the resulted, smaller clusters until you decide not to further cluster according to the criteria.

This bisecting clustering algorithm will generate a dendrogram (a binary tree), and each node (either internal or leave) corresponds to a cluster; the two clusters (e.g., C_1 and C_2) that are bisected from a cluster (e.g., C_0) will become the children of the cluster (i.e., a node will correspond to C_0, and it will have two child nodes corresponding to C_1 and C_2, respectively). Please always put the smaller cluster as the left child and the larger cluster as the right child. You always choose the current largest leaf node to further bisect.

We will use three criteria to decide whether further clustering will be needed. The first one is the cluster size, that is, the number of data points in the cluster – if the cluster size is smaller than the user specified threshold, you should not further cluster the data points in that cluster. The second one is the total number of clusters – if the current number of clusters in total is equal to or greater than the user specified threshold, you should not further cluster. Note that since the bisecting algorithm will generate a dendrogram, the number of clusters here refers to the number of leaves of the dendrogram. The third one is the intra-cluster distance. For a cluster c, it is defined as follows,

$$D_c = \frac{1}{n_c} \sum_{i=1}^{n_c} d(x_i, x_c)$$

where n_c is the number of data points within the cluster c, x_c is the centroid of cluster c, and $d(x_i, x_c)$ is the **euclidean** distance between data point x_i and centroid x_c, where x_i is in cluster c. If D_c is smaller than the user specified threshold, you should not further cluster the data points in c. If one of the above three criteria is met, you should stop bisecting clustering.

1. (5 points) You need to implement a simple user interface (via command line) such that users can input data file names and other parameters. Your executable should be named as BK.py. The command line should look like

   BK.py -data data.txt -k cluster_num -s cluster_size -d intra_dist -output output_file.txt

   where "-data" specifies the input file of data points that you need to cluster, "-k" specifies the total number of desired clusters, "-s" specifies the maximum number of data points

allowed within a cluster, "-d" specifies the maximum intra-cluster distance allow within in a cluster, and "-output" specifies the output file name.

2. (5 points) In addition to the output file, you will need to print to the stdout the number of data points associated with each node (both internal and leave) in the dendrogram. For example, for the root, it is going to be the total number of data points from the input file; for the left child of the root, it is going to be the number of data points from all the data points that are clustered into one cluster. Please output these numbers in a level-by-level fashion, for example, as follows:

   50

   20 30

   9 11 12 18

   4 5 5 6 5 7 8 10

   where 50 is the number of data points at root; 20 is the number of data points at the left child of the root, and 30 is the number of data points at the right child of the root; 9 is the number of data points of the left child of "20" and 11 is the number of data points of the right child of "20"; etc.

3. (5 points) You need to read in the input data from the input files. You need to store the data into proper data structures. This URL `https://www.numpy.org/devdocs/reference/generated/numpy.genfromtxt.html` may be helpful. You just need

   - import numpy as np
   - X = np.genfromtxt(file_name, delimiter=' ')

   to get the data from file_name into a matrix A, where numpy is a Python library that supports for large, multi-dimensional arrays and matrices and their operations.

4. (100 points) You need to implement the bisecting k-means algorithm to cluster data points.

   [20 points] You will implement k-means algorithm as each bisecting clustering step will use k-means. You need to have a function named "kmeans".

   [10 points] You will need a data structure for the dendrogram. The dendrogram should store all the data point information and other necessary information associated with each node in the dendrogram. You will need to build this dendrogram during the course of the recursive bisecting clustering.

   [50 points] You will need to implement a function named "bisecting", which does the bisecting clustering and generates the dendrogram as the output (i.e., function "bisecting" returns the dendrogram).

   [10] You need to implement a function named "icd", which calculates the intra-cluster distance of a given cluster.

   [10] You need to optimize the memory usage and the efficiency of your implementation (e.g., every time you use k-means, you will need the pairwise distances; you may need

to think about whether you need to calculate the distances every time you use k-means, or it is better to pre-calculate them and calculate them only once).

Note: you need to implement the algorithm from scratch (e.g., calculate the distance yourself, implement k-means on your own). You should not use any existing implementation or functions from any machine learning libraries or package (e.g., from scikit-learn)

5. (10 points) Please output your clustering solution into the specified output file. Once your bisecting algorithm is done, each data point should be at one and only one leaf node of the dendrogram. All the leaf nodes (i.e., final clusters) will be indexed from left to right (i.e., the left-most leaf is indexed as cluster 0). You need to output the cluster index of each input data point in the same order of the data points as in the input file. That is, the first line in the output file should be the cluster index of the first input data point; the second line in the output file should be the cluster index of the second input data point; etc. You need to implement a function name "print_clusters" which does the above. This function should take the dendrogram produced from the "bisecting" function as one of its inputs.

   What to submit:

   1. Your source code in a zipped file. You should name your zipped file as PA6_X_Y.tar.gz where X is your first name (capital letters) and Y is your last name (capital letters). Please following the naming schemes specified, otherwise, **20** points will be taken off. This is for the TA/graders to quickly find your results.
   2. It is mandatory that at least 1/4 of the lines (not including empty lines) in your code should be dedicated to comments. Otherwise, **20%** of the points that you will get will be taken off (non-negotiable!)