Task 1 Suppose we have 10 college football teams X1 to X10. We want to cluster them into 2 groups. For each football team, we have two features: One is # wins in Season 2016, and the other is # wins in Season 2017.

Team	# wins in Season 2016 (x-axis)	# wins in Season 2017 (y-axis)
X1	3	5
X2	3	4
X3	2	8
X4	2	3
X5	6	2
X6	6	4
X7	7	3
X8	7	4
X9	8	5
X10	7	6

(1) Initialize with two centroids, (4, 6) and (5, 4). Use Manhattan distance as the distance metric. First, perform one iteration of the K-means algorithm and report the coordinates of the resulting centroids. Second, please use K-Means to find two clusters.

(2) Initialize with two centroids, (4, 6) and (5, 4). Use Euclidean distance as the distance metric. First, perform one iteration of the K-means algorithm and report the coordinates of the resulting centroids. Second, please use K-Means to find two clusters.

(3) Initialize with two centroids, (3, 3) and (8, 3). Use Manhattan distance as the distance metric. First, perform one iteration of the K-means algorithm and report the coordinates of the resulting centroids. Second, please use K-Means to find two clusters.

(4) Initialize with two centroids, (3, 2) and (4, 8). Use Manhattan distance as the distance metric. First, perform one iteration of the K-means algorithm and report the coordinates of the resulting centroids. Second, please use K-Means to find two clusters.

Task 2 K-Means Clustering with Real World Dataset

First, download the Iris data set from: https://archive.ics.uci.edu/ml/datasets/Iris. Then, implement the K-means algorithm. K-means algorithm computes the distance of a given data point pair. Replace the distance computation function with Euclidean distance, 1- Cosine similarity, and 1 – the **Generalized** Jarcard similarity (https://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/jaccard.htm).

Q1: Run K-means clustering with Euclidean, Cosine and Jarcard similarity. Specify K= the number of categorical values of y (the variable of label). Compare the SSEs of Euclidean-K-means Cosine-K-means, Jarcard-K-means. Which method is better?

Q2: Compare the accuracies of Euclidean-K-means Cosine-K-means, Jarcard-K-means. First, label each cluster with the label of the highest votes. Later, compute the accuracy of the K-means with respect to the three similarity metrics. Which metric is better?

Q3: Which of Euclidean-K-means, Cosine-K-means, Jarcard-K-means requires more iterations and times?

Q4: Compare the SSEs of Euclidean-K-means Cosine-K-means, Jarcard-K-means with respect to the following three terminating conditions:

- when there is no change in centroid position
- when the SSE value increases in the next iteration

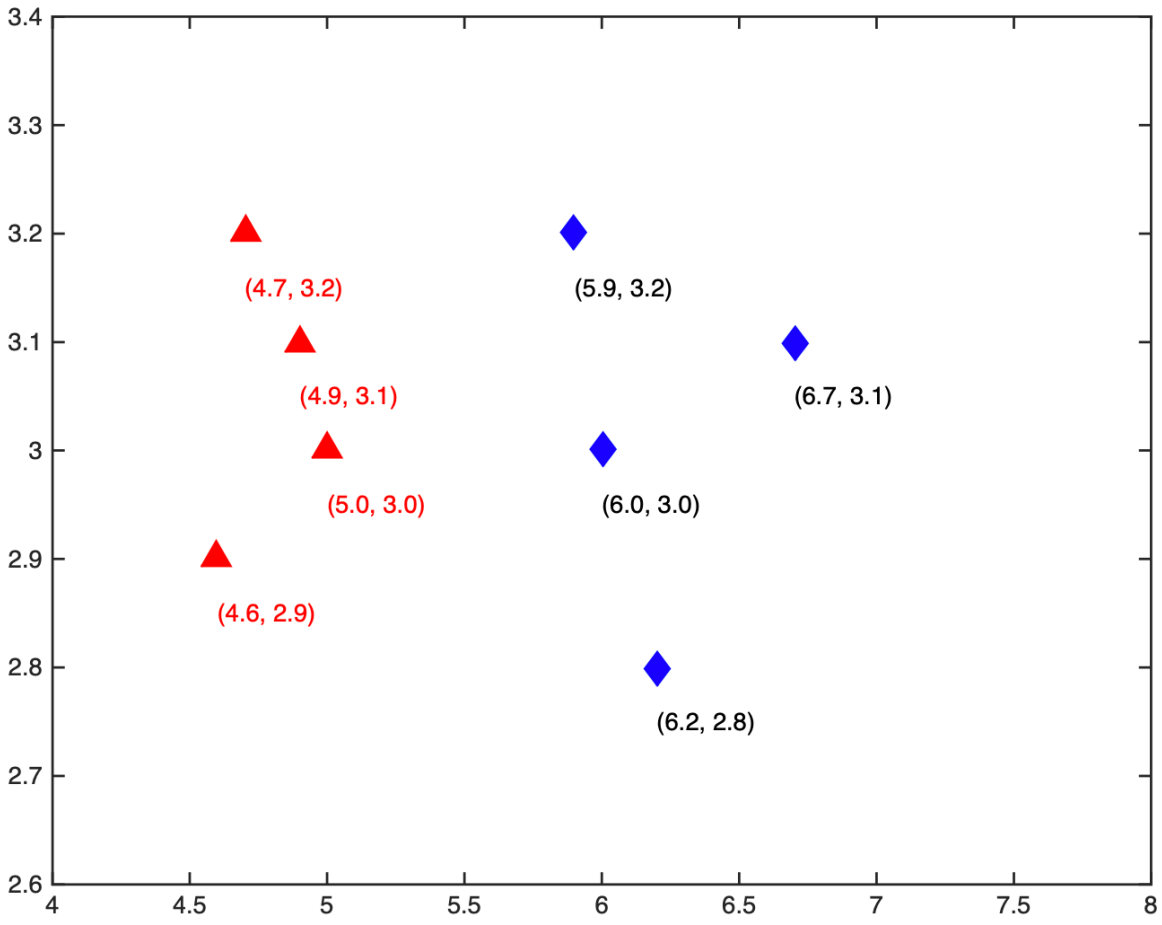
- when the maximum preset value (100) of iteration is complete

Which method requires more time or more iterations?

3, Understanding K-Means:

- Please give a scenario in which K-means cluster may not work very well?
- The classic K-means algorithm randomly initializes K centers. Is there any better strategy for selecting K initial centers?

4, There are two clusters A (red) and B (blue), each has four members and plotted in Figure. The coordinates of each member are labeled in the figure. Compute the distance between two clusters using Euclidean distance.



A. What is the distance between the two farthest members? (round to four decimal places here, and next 2 problems);

B. What is the distance between the two closest members?

C. What is the average distance between all pairs?

D, Discuss which distance (A, B, C) is more robust to noises in this case?

Please submit a PDF report. In your report, please answer each question with your explanations, plots, results in brief. DO NOT paste your code or snapshot into the PDF. At the end of your PDF, please include a website address (e.g., Github, Dropbox, OneDrive, GoogleDrive) that can allow the TA to read your code.