**MCS 253**

**HW 6**

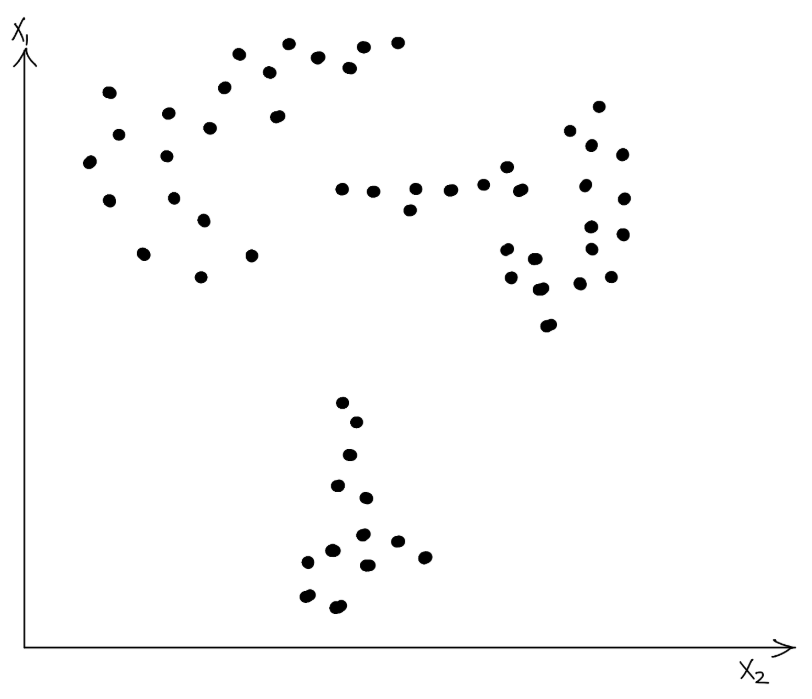
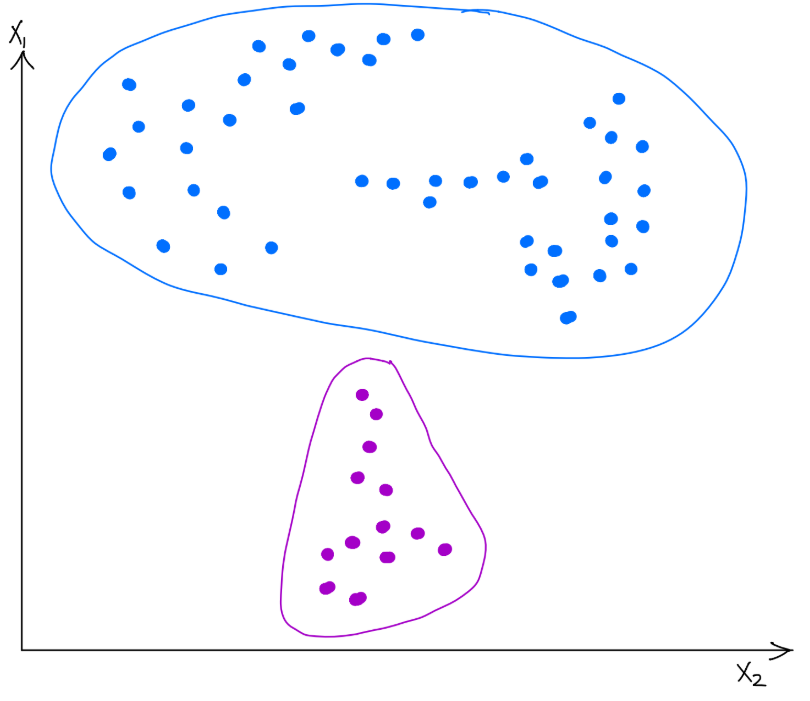
[Read this first on what to submit for homework.](https://drive.google.com/open?id=1vvTZzNrXNrxFCFVuzryEuAX6c8rfGERsEc7t4TS1vAA)

**HW 6.1 - Chained K-Nearest-Neighbor Clusters (50 points)**

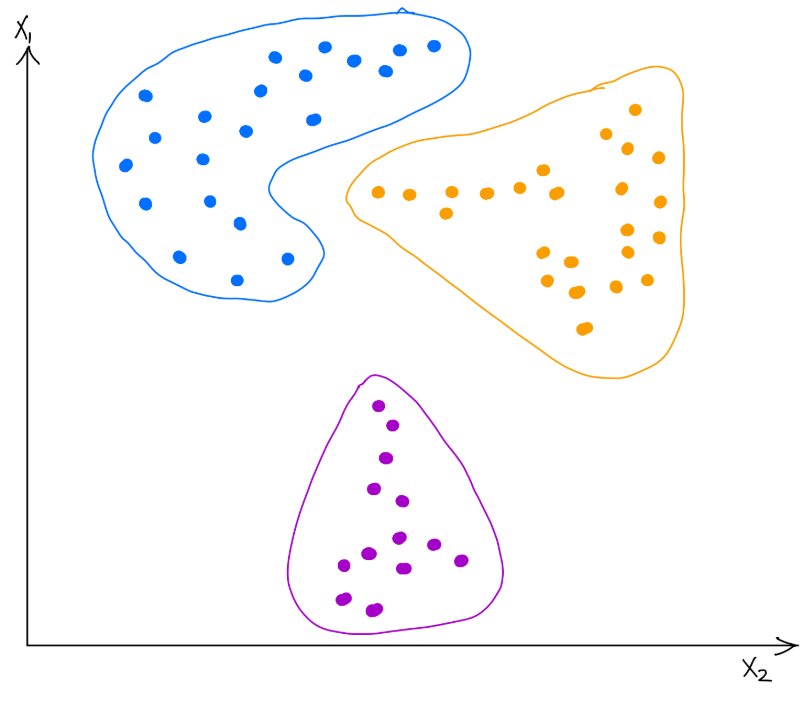
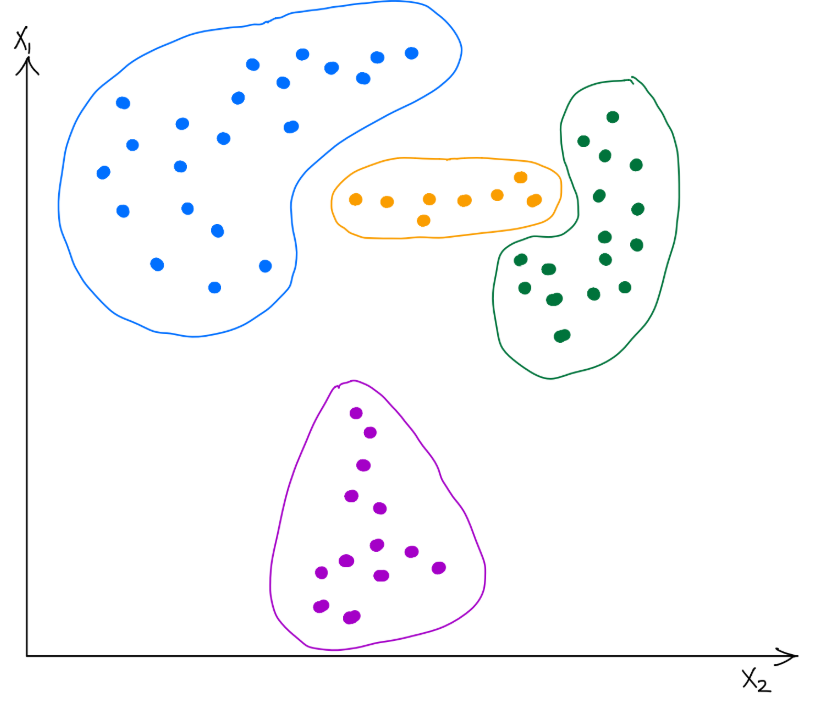
An epidemic is breaking out in a region containing M cities, and quarantine partitions are needed based on the likely spread of the disease. **You are asked to write a program that clusters the cities into K clusters (with K being a variable parameter indicating the number of quarantined partitions) such that if a city is closest to some other city, then those cities are part of the same quarantine group.**

Using Euclidean distance as a measure of closeness between points, the following cities would be clustered as follows (with varying K)…

original data: K=2 clusters:

K= clusters: K=4 clusters:

Input:

* The number of cities (ie. ***M***)
* ***M*** city names and their locations on an x1 vs. x2 axis (data provided via stdin)
* various values for ***K*** to compute clusters for (also via stdin)

Output:

* cluster assignments for each value of ***K***

Example:

input:

[hw6\_sample\_input.txt](https://drive.google.com/open?id=1DVRTlplL7vH4tP8E9y26IR7Y90WDD_-G)

* the first number is represents ***M*** (the number of data points)
* the next ***M*** non-empty lines represent the ***M*** data points [name x1 x2]
* each number that follows is a value of ***K*** to compute clusters for

output:

K=2

Cluster 1: A, B, C, D, E, F, G, H, I, J, K, L

Cluster 2: M, N, O

K=3

Cluster 1: A, B, C, D

Cluster 2: E, F, G, H, I, J, K, L

Cluster 3: M, N, O

K=4

Cluster 1: A, B, C, D

Cluster 2: E, F, G, H

Cluster 3: I, J, K, L

Cluster 4: M, N, O

\*Note: the points assigned to the same cluster must be as shown above, however the cluster numbers assigned to each cluster can be different

**HW 6.2 - Two LeetCode Problems (25 points each)**

Possible LeetCode Problems:

* 802. Find Eventual Safe States
* 913. Cat and Mouse
* 1254. Number of Closed Islands
* 1129. Shortest Path with Alternating Colors
* 1334. Find the City With the Smallest Number of Neighbors at a Threshold Distance
* 928. Minimize Malware Spread II
* 685. Redundant Connection II