

CSE341 PROJECT REPORT

Name: Jennie Chepenuk

UB Person #: 50041501

Date: 12/06/2019

1. METHODS (10 POINTS)

To implement K-clustering in assembly, I hardcoded the ten input points and required user input for the initial centroids, storing them in registers. I stored the size of each cluster (initially zero) on the stack along with four points that stored the (X, Y) coordinates of the centroids.

I then looped over every point in the input and compared the Euclidean distance between the point and centroid 1 and 2. I then jumped to a function to add to one of the clusters depending if the point was closer to centroid 1 or 2. I incremented the size of the cluster on the stack at each cluster add iteration so I had a stopping condition when looping through the points in the clusters later. Afterwards I jumped back to the loop until all points had been iterated through, at which point I jumped to finding a new centroid.

I then added the each of the clusters' x and y coordinates and used integer division on each coordinate to determine the new centroid 1 and 2 (X, Y). I had four 'bne' conditions which compared each element of the new centroids with their respective former point stored on the stack. If any coordinates were unequal, I stored these new centroids on the stack, reinitialized the cluster sizes to zero, and jumped back to the loop to iterate through the points and compare the distances from the newly updated centroids.

If the stopping 'bne' conditions were not met, this meant that I had found the final centroids and the final clustering of the points. I then jumped to load the cluster arrays, print the final clustering, and exit.

2. SYSTEM ROBUSTNESS (10 POINTS)

Input points: [(1, 5), (2, 4), (4, 7), (8, 8), (6, 6), (3, 6), (4, 5), (4, 2), (1, 3), (6, 7)]

Trial one:

Centroid 1: (1, 5)

Centroid 2: (2, 4)

Output:

Cluster 1: [(1, 5), (2, 4), (1, 3)]

Cluster 2: [(4, 7), (8, 8), (6, 6), (3, 6), (4, 5), (4, 2), (6, 7)]

Trial two:

Centroid 1: (5, 4)

Centroid 2: (2, 3)

Output:

Cluster 1: [(4, 7), (8, 8), (6, 6), (3, 6), (4, 5), (4, 2), (6, 7)]

Cluster 2: [(1, 5), (2, 4), (1, 3)]

Trial three:

Centroid 1: (1, 2)

Centroid 2: (3, 4)

Output:

Cluster 1: [(1, 5), (2, 4), (1, 3)]

Cluster 2: [(4, 7), (8, 8), (6, 6), (3, 6), (4, 5), (4, 2), (6, 7)]