CSC 323-32: Project 4 <K Mean Clustering> (C++)

Preston Peck

Due date: Nov. 1, 2016

\*\*\*\* Algorithm steps for K Mean Clustering:

Step 0: Open the input file

0.1: K, numRow, numCol numPts- get from input file.

0.2: imageArray- Dynamically allocate a 2-D arrays, size numRows X numCols.

0.3: pointSet- Dynamically allocate the point set, size of numPts

0.4: Kcentroids[K]- Dynamically allocate the K centroids struct.

Step 1: call loadPointSet

Step 2: call assignLabel

Step 3: call mapPoint2Image

Step 4: call displayImage

Step 5: Go thru the entire pointSet struct array to compute the centroids of the K clusters. Store

the centroids in each Kcentroids[i], i from 1 to K.

Step 6: for each point, p, in the pointSet array compute the distance, dist(p,ci), from p to the

centroids of each Kcentroids[i], i = 1 to K

6.1: min\_i <-- determine which dist(p,ci) is minimum

6.2: if min\_i is not the same as p's old label change p's label to min\_i and set changeLabel to

true

Step 7: repeat step 6 until all points in pointSet are process.

Step 8: repeat step 3 to step 7 until no point changes its label.

Step 9: Output the info of pointSet to Output-1 file.

Input

4

90

80 80

18 58

21 52

19 46

13 46

16 38

11 36

11 58

14 42

10 42

10 41

10 44

11 54

9 57

10 52

21 41

8 46

12 38

8 35

10 38

13 49

8 46

20 55

21 53

11 48

11 53

14 57

9 49

16 39

10 35

13 51

39 14

38 9

29 20

38 11

27 21

41 24

29 23

36 10

37 19

29 10

35 26

36 20

35 23

34 18

37 24

38 56

37 60

47 54

45 48

44 60

39 51

43 52

Output

18 58 1

21 52 1

19 46 1

13 46 1

16 38 1

11 36 1

11 58 1

14 42 1

10 42 1

10 41 1

10 44 1

11 54 1

9 57 1

10 52 1

21 41 1

8 46 1

12 38 1

8 35 1

10 38 1

13 49 1

8 46 1

20 55 1

21 53 1

11 48 1

11 53 1

14 57 1

9 49 1

16 39 1

10 35 1

13 51 1

39 14 4

38 9 4

29 20 4

38 11 4

27 21 4

41 24 4

29 23 4

36 10 4

37 19 4

29 10 4

35 26 4

36 20 4

35 23 4

34 18 4

37 24 4

38 56 2

37 60 2

47 54 2

45 48 2

44 60 2

39 51 2

43 52 2

38 61

45 63

44 55

44 47

41 66

40 55

49 52

44 53

59 30

61 36

65 34

61 38

66 25

58 39

66 39

68 30

61 28

62 25

62 30

64 20

59 24

63 28

61 30

69 24

69 32

71 30

70 38

57 39

58 43

66 28

64 21

64 39

63 41

67 44

63 29

61 21

65 43

63 24

38 61 2

45 63 2

44 55 2

44 47 2

41 66 2

40 55 2

49 52 2

44 53 2

59 30 3

61 36 3

65 34 3

61 38 3

66 25 3

58 39 3

66 39 3

68 30 3

61 28 3

62 25 3

62 30 3

64 20 3

59 24 3

63 28 3

61 30 3

69 24 3

69 32 3

71 30 3

70 38 3

57 39 3

58 43 3

66 28 3

64 21 3

64 39 3

63 41 3

67 44 3

63 29 3

61 21 3

65 43 3

63 24 3

2 1

3 1

1 3 21 3 2

2 4 14 3

1

4 4 2

4 2

14

1

3

2

3 23

3

4 1 1

4

3 1

2 2

3 1 3

4 2 2 1

3 3

2

4 1

4

4 4 3 2

1 2

4

3

4

2 1

1 1

4 1 3 2 4

2 3

2 23 1

43 4

3 1

1 2 3

2

4

4 1

3

2

1 1

1 1

1 1 11 1 1

1 1 11 1

1

1 1 1

1 1

11

1

1

1

1 11

4

4 4 4

4

4 4

4 4

4 4 1

4 4 2 2

4 2

2

4 2

2

2 2 2 2

2 2

2

2

3

3 3

4 3

4 3 3 3 3

4 3

4 33 3

44 3

3 3

3 3 3

3

3

3 3

3

3

1 1

1 1

1 1 11 1 1

1 1 11 1

1

1 1 1

1 1

11

1

1

1

1 11

4

4 4 4

4

4 4

4 4

4 4 2

4 4 2 2

4 2

2

4 2

2

2 2 2 2

2 2

2

2

3

3 3

3 3

3 3 3 3 3

3 3

3 33 3

33 3

3 3

3 3 3

3

3

3 3

3

3