**CS323-22: Project 5 (CPP)**

**Yida Tao**

**Due date: Oct. 19, 2017**

Algorithm Steps:

step 0:

- inFile 🡨 Open the input file

- K, numRow, numCol numPts 🡨 get from inFile.

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

- pointSet 🡨 Dynamically allocate the point set,size ofnumPts

- Kcentroids[K] 🡨 Dynamically allocate the K centroids struct

Step 1: call loadPointSet

Step 2: call assignLabel

Step 3: call mapPoint2Image

Step 4: call displayImage // output to output-2

step 5: 5.1: changeLabel 🡨 0

5.2: Go thru the entire pointSet struct array to compute the centroids of each of the K clusters. Store the computed centroids in each Kcentroids[i], i from 1 to K.

step 6: 6.1: 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

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

6.3: if min\_i’s Label is not the same as p's Label change p's label to min\_i’s label and increment changeLabel ++

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

Step 8: repeat step 3 to step 7 while changeLabel > 0

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

Step 10: close all files.

**Source Code**

#include <iostream>

#include <fstream>

#include <string>

#include <cmath>

using namespace std;

class Point{

public:

int X;

int Y;

int Label;

double Distance;

//constructors

Point(int x, int y){

this->X =x;

this->Y =y;

this->Label =0;

this->Distance =0;

}

//destructor

~Point(){}

void PrintPoint(){

cout<<"X:"<<X<<" Y:"<<Y<<" Label:"<<Label<<

" Distance to its own cluster centroid:"<<Distance<<endl;

}

void setPoint(int x,int y){

this->X = x;

this->Y = y;

}

//destructor

void deletePoint(){

delete this;

}

};

class Kmean{

struct xyCoord{

int K\_X;

int K\_Y;

int K\_Label;

//constructor

xyCoord(int x,int y,int l){

K\_X = x;

K\_Y = y;

K\_Label = l;

}

};

public:

int K;

xyCoord \*\*Kcentroids;

int K\_numPts;

Point \*\*pointSet;

int K\_numRow;

int K\_numCol;

int \*\*imageArray;

int changes;

int counter;

int print\_trial;

//constructor

Kmean(int k,int numPts,int numRow,int numCol){

K = k;

K\_numPts = numPts;

K\_numRow = numRow;

K\_numCol = numCol;

Kcentroids = new xyCoord\*[K];

pointSet = new Point\*[K\_numPts];

imageArray = new int\*[K\_numRow];

for(int i = 0; i < K\_numRow; i++){

imageArray[i] = new int[K\_numCol];

for(int j = 0; j < K\_numCol; j++){

imageArray[i][j] = 0;

}

}

counter = 0;

print\_trial = 0;

}

//destructor

~Kmean(){

for(int i = 0; i < K; i++){

delete Kcentroids[i];

}

delete[] Kcentroids;

delete[] imageArray;

for(int i = 0; i < K\_numPts; i++){

pointSet[i]->deletePoint();

}

delete[] pointSet;

}

void loadPointSet(int x, int y){

pointSet[counter] = new Point(x,y);

counter++;

}

void assignLable(){

// assign label from 1 to k, for every point in set

for(int i = 0; i < K\_numPts; i++){

pointSet[i]->Label = i%K+1;

}

}

void mapPoint2Image(){

int x;

int y;

int l;

for(int i = 0; i < K\_numPts; i++){

x = pointSet[i]->X;

y = pointSet[i]->Y;

l = pointSet[i]->Label;

imageArray[x][y] = l;

}

}

void kMeanClustering(){

changes = 0;

int x1=0,y1=0,count1=0;

int x2=0,y2=0,count2=0;

int x3=0,y3=0,count3=0;

int x4=0,y4=0,count4=0;

//step 5 look for centroids

for(int i = 0; i < K\_numPts; i++){

if(pointSet[i]->Label == 1){

x1 += pointSet[i]->X;

y1 += pointSet[i]->Y;

count1++;

}

else if(pointSet[i]->Label == 2){

x2 += pointSet[i]->X;

y2 += pointSet[i]->Y;

count2++;

}

else if(pointSet[i]->Label == 3){

x3 += pointSet[i]->X;

y3 += pointSet[i]->Y;

count3++;

}

else {

x4 += pointSet[i]->X;

y4 += pointSet[i]->Y;

count4++;

}

}

//x1 == x2, y1 == y2 1st time, so i modified x1 by plus 1

x1 = x1/count1 + 1;

y1 = y1/count1;

x2 = x2/count2;

y2 = y2/count2;

x3 = x3/count3;

y3 = y3/count3;

x4 = x4/count4;

y4 = y4/count4;

cout<<"centroid #1:"<<x1<<" "<<y1<<endl;

cout<<"centroid #2:"<<x2<<" "<<y2<<endl;

cout<<"centroid #3:"<<x3<<" "<<y3<<endl;

cout<<"centroid #4:"<<x4<<" "<<y4<<endl;

Kcentroids[0]= new xyCoord(x1,y1,1);

Kcentroids[1]= new xyCoord(x2,y2,2);

Kcentroids[2]= new xyCoord(x3,y3,3);

Kcentroids[3]= new xyCoord(x4,y4,4);

//step 6 ,7

reLabel();

}

void reLabel(){

//step 7 repeat step 6 until all points in pointSet are processed.

for(int i = 0; i < K\_numPts; i++){

minDistance(pointSet[i]);

}

}

//step 6 if min\_i's Label is not the same as p's Label change p's label to

void minDistance(Point \*p){

int dx;

int dy;

double min = 9999.9;

int Label\_temp = Kcentroids[0]->K\_Label;

double d;

for(int i = 0; i < K; i++){

//cout<<Kcentroids[i]->K\_X<<" "<<Kcentroids[i]->K\_Y<<" "<<Kcentroids[i]->K\_Label<<endl;

dx = pow((p->X - Kcentroids[i]->K\_X),2);

dy = pow((p->Y - Kcentroids[i]->K\_Y),2);

d = sqrt(dx + dy);

if(d < min){

min = d;

Label\_temp = Kcentroids[i]->K\_Label;

}

}

if(p->Label != Label\_temp){

p->Label = Label\_temp;

changes++;

}

p->Distance = min;

}

void printPointSet(){

cout<<endl;

cout<<"PrintPointSet:"<<endl;

for(int i = 0; i < K\_numPts; i++){

pointSet[i]->PrintPoint();

}

}

void PrettyPrint(){

cout<<endl<<"Trails: "<<print\_trial<<endl;

for(int i = 0; i < K\_numRow; i++){

for(int j = 0; j < K\_numCol; j++){

if(imageArray[i][j] )

cout<<imageArray[i][j];

else

cout<<" ";

}

cout<<endl;

}

cout<<"Trails: "<<print\_trial<<" end here!"<<endl;

cout<<endl;

print\_trial++;

}

};

int main(int argc, char \*argv[]){

Kmean \*myKmean;

int k;

int numPts;

int r;

int c;

if(argv[1]==NULL) {

cout<<"no parameter"<<endl;

return 0;

}

ifstream inFile;

inFile.open(argv[1]);

if(!inFile.is\_open()){

cout<<"cant find file"<<endl;

return 0;

}

streambuf \*console = cout.rdbuf();

ofstream out1;

out1.open(argv[2]);

ofstream out2;

out2.open(argv[3]);

//set output to outfile 1

cout.rdbuf(out1.rdbuf());

//step 1.1 read first 3 line

while(!inFile.eof()){

inFile>>k;

cout<<k<<endl;

inFile>>numPts;

cout<<numPts<<endl;

inFile>>r;

inFile>>c;

cout<<r<<" "<<c<<endl;

break;

}

//paremeters: int k,int numPts,int numRow,int numCol

myKmean = new Kmean(k,numPts,r,c);

//step 1.2 read rest file

while(inFile>>r){

if(inFile>>c){

cout<<r<<" "<<c<<endl;

myKmean->loadPointSet(r,c);

}

}

inFile.close();

//step 2 assignlable randomly

myKmean->assignLable();

cout.rdbuf(out2.rdbuf());

do{

//step 3 call mapPoint2Image

myKmean->mapPoint2Image();

//step 4 call prettyprint

myKmean->PrettyPrint();

//step 5,6,7

myKmean->kMeanClustering();

//step 8 repeat step 3 to step 7 while changes > 0

} while(myKmean->changes > 0 );

//step 9 Output the info of pointSet to Output-1 file.

cout.rdbuf(out1.rdbuf());

myKmean->printPointSet();

//step 10 close all files, delete objects

cout.rdbuf(console);

cout<<"done"<<endl;

out1.close();

out2.close();

delete myKmean;

}

**Input**

4

90

80 80

44 53

59 30

61 36

65 34

13 46

16 38

11 36

39 14

38 9

29 20

61 38

11 58

14 42

10 42

69 32

71 30

70 38

10 41

10 44

11 54

63 29

61 21

65 43

9 57

10 52

21 41

8 46

12 38

68 30

61 28

8 35

10 38

66 25

58 39

13 49

8 46

20 55

38 56

37 60

21 53

11 48

64 39

63 41

11 53

14 57

66 39

9 49

16 39

10 35

13 51

67 44

63 24

38 11

27 21

41 24

29 23

36 10

37 19

29 10

58 43

66 28

64 21

35 26

37 24

62 25

63 28

38 56

37 60

47 54

45 48

44 60

39 51

43 52

18 58

20 56

62 30

64 20

59 24

21 52

38 61

45 63

44 55

19 46

44 47

41 66

40 55

49 52

61 30

69 24

57 39

**Output 1**

4

90

80 80

44 53

59 30

61 36

65 34

13 46

16 38

11 36

39 14

38 9

29 20

61 38

11 58

14 42

10 42

69 32

71 30

70 38

10 41

10 44

11 54

63 29

61 21

65 43

9 57

10 52

21 41

8 46

12 38

68 30

61 28

8 35

10 38

66 25

58 39

13 49

8 46

20 55

38 56

37 60

21 53

11 48

64 39

63 41

11 53

14 57

66 39

9 49

16 39

10 35

13 51

67 44

63 24

38 11

27 21

41 24

29 23

36 10

37 19

29 10

58 43

66 28

64 21

35 26

37 24

62 25

63 28

38 56

37 60

47 54

45 48

44 60

39 51

43 52

18 58

20 56

62 30

64 20

59 24

21 52

38 61

45 63

44 55

19 46

44 47

41 66

40 55

49 52

61 30

69 24

57 39

PrintPointSet:

X:44 Y:53 Label:3 Distance to its own cluster centroid:3.60555

X:59 Y:30 Label:1 Distance to its own cluster centroid:5

X:61 Y:36 Label:1 Distance to its own cluster centroid:5.74456

X:65 Y:34 Label:1 Distance to its own cluster centroid:3.16228

X:13 Y:46 Label:4 Distance to its own cluster centroid:0

X:16 Y:38 Label:4 Distance to its own cluster centroid:8.544

X:11 Y:36 Label:4 Distance to its own cluster centroid:10.1489

X:39 Y:14 Label:2 Distance to its own cluster centroid:5.74456

X:38 Y:9 Label:2 Distance to its own cluster centroid:8.94427

X:29 Y:20 Label:2 Distance to its own cluster centroid:5.74456

X:61 Y:38 Label:1 Distance to its own cluster centroid:7.61577

X:11 Y:58 Label:4 Distance to its own cluster centroid:12.1655

X:14 Y:42 Label:4 Distance to its own cluster centroid:4.12311

X:10 Y:42 Label:4 Distance to its own cluster centroid:5

X:69 Y:32 Label:1 Distance to its own cluster centroid:5

X:71 Y:30 Label:1 Distance to its own cluster centroid:7.07107

X:70 Y:38 Label:1 Distance to its own cluster centroid:9.21954

X:10 Y:41 Label:4 Distance to its own cluster centroid:5.74456

X:10 Y:44 Label:4 Distance to its own cluster centroid:3.60555

X:11 Y:54 Label:4 Distance to its own cluster centroid:8.24621

X:63 Y:29 Label:1 Distance to its own cluster centroid:2.23607

X:61 Y:21 Label:1 Distance to its own cluster centroid:10.3923

X:65 Y:43 Label:1 Distance to its own cluster centroid:12.0416

X:9 Y:57 Label:4 Distance to its own cluster centroid:11.6619

X:10 Y:52 Label:4 Distance to its own cluster centroid:6.7082

X:21 Y:41 Label:4 Distance to its own cluster centroid:9.38083

X:8 Y:46 Label:4 Distance to its own cluster centroid:4.89898

X:12 Y:38 Label:4 Distance to its own cluster centroid:8.06226

X:68 Y:30 Label:1 Distance to its own cluster centroid:4.12311

X:61 Y:28 Label:1 Distance to its own cluster centroid:4.24264

X:8 Y:35 Label:4 Distance to its own cluster centroid:12

X:10 Y:38 Label:4 Distance to its own cluster centroid:8.544

X:66 Y:25 Label:1 Distance to its own cluster centroid:6.32456

X:58 Y:39 Label:1 Distance to its own cluster centroid:10

X:13 Y:49 Label:4 Distance to its own cluster centroid:3

X:8 Y:46 Label:4 Distance to its own cluster centroid:4.89898

X:20 Y:55 Label:4 Distance to its own cluster centroid:11.4018

X:38 Y:56 Label:3 Distance to its own cluster centroid:3.16228

X:37 Y:60 Label:3 Distance to its own cluster centroid:6.32456

X:21 Y:53 Label:4 Distance to its own cluster centroid:10.6301

X:11 Y:48 Label:4 Distance to its own cluster centroid:2.82843

X:64 Y:39 Label:1 Distance to its own cluster centroid:8

X:63 Y:41 Label:1 Distance to its own cluster centroid:10

X:11 Y:53 Label:4 Distance to its own cluster centroid:7.28011

X:14 Y:57 Label:4 Distance to its own cluster centroid:11

X:66 Y:39 Label:1 Distance to its own cluster centroid:8.24621

X:9 Y:49 Label:4 Distance to its own cluster centroid:5

X:16 Y:39 Label:4 Distance to its own cluster centroid:7.61577

X:10 Y:35 Label:4 Distance to its own cluster centroid:11.3578

X:13 Y:51 Label:4 Distance to its own cluster centroid:4.89898

X:67 Y:44 Label:1 Distance to its own cluster centroid:13.3041

X:63 Y:24 Label:1 Distance to its own cluster centroid:7.07107

X:38 Y:11 Label:2 Distance to its own cluster centroid:7.2111

X:27 Y:21 Label:2 Distance to its own cluster centroid:8.06226

X:41 Y:24 Label:2 Distance to its own cluster centroid:9.89949

X:29 Y:23 Label:2 Distance to its own cluster centroid:7.74597

X:36 Y:10 Label:2 Distance to its own cluster centroid:7.28011

X:37 Y:19 Label:2 Distance to its own cluster centroid:3.60555

X:29 Y:10 Label:2 Distance to its own cluster centroid:8.544

X:58 Y:43 Label:1 Distance to its own cluster centroid:13.4164

X:66 Y:28 Label:1 Distance to its own cluster centroid:3.60555

X:64 Y:21 Label:1 Distance to its own cluster centroid:9.94987

X:35 Y:26 Label:2 Distance to its own cluster centroid:9.05539

X:37 Y:24 Label:2 Distance to its own cluster centroid:7.61577

X:62 Y:25 Label:1 Distance to its own cluster centroid:6.32456

X:63 Y:28 Label:1 Distance to its own cluster centroid:3.16228

X:38 Y:56 Label:3 Distance to its own cluster centroid:3.16228

X:37 Y:60 Label:3 Distance to its own cluster centroid:6.32456

X:47 Y:54 Label:3 Distance to its own cluster centroid:6.08276

X:45 Y:48 Label:3 Distance to its own cluster centroid:8.06226

X:44 Y:60 Label:3 Distance to its own cluster centroid:5.74456

X:39 Y:51 Label:3 Distance to its own cluster centroid:4.47214

X:43 Y:52 Label:3 Distance to its own cluster centroid:3.60555

X:18 Y:58 Label:4 Distance to its own cluster centroid:12.9615

X:20 Y:56 Label:4 Distance to its own cluster centroid:12.1655

X:62 Y:30 Label:1 Distance to its own cluster centroid:2.23607

X:64 Y:20 Label:1 Distance to its own cluster centroid:10.9545

X:59 Y:24 Label:1 Distance to its own cluster centroid:8.544

X:21 Y:52 Label:4 Distance to its own cluster centroid:10

X:38 Y:61 Label:3 Distance to its own cluster centroid:6.7082

X:45 Y:63 Label:3 Distance to its own cluster centroid:8.94427

X:44 Y:55 Label:3 Distance to its own cluster centroid:3

X:19 Y:46 Label:4 Distance to its own cluster centroid:6

X:44 Y:47 Label:3 Distance to its own cluster centroid:8.544

X:41 Y:66 Label:3 Distance to its own cluster centroid:10.9545

X:40 Y:55 Label:3 Distance to its own cluster centroid:1

X:49 Y:52 Label:3 Distance to its own cluster centroid:8.544

X:61 Y:30 Label:1 Distance to its own cluster centroid:3.16228

X:69 Y:24 Label:1 Distance to its own cluster centroid:8.544

X:57 Y:39 Label:1 Distance to its own cluster centroid:10.6301

**Output 2**

Trails: 0

3 4

3 4

1 4 22 3 1

3 1 44 4

4

1 3 2

1 1

24

2

3

13

2 34

2

3 2 4

3

1

2 4 4

1 1 3 4

4 4

2

3 1

1

4 1 2 3

2 1

1

3

2

2 4

2 2

2 2 4 3 3

1 4

4 21 3

12 2

4 3

1 1 2

3

1

1 3

1

4

Trails: 0 end here!

centroid #1:42 37

centroid #2:41 37

centroid #3:35 42

centroid #4:34 41

Trails: 1

4 4

4 4

4 4 44 4 4

4 4 44 4

4

4 4 4

4 4

44

3

4

33

4 44

4

2 2 2

2

2

2 2 3

2 2 3 3

2 3

3

2 3

3

1 3 3 3

1 3

3

1

1

1 1

1 1

1 1 1 1 1

1 1

1 11 1

11 1

1 1

1 1 1

1

1

1 1

1

1

Trails: 1 end here!

centroid #1:62 33

centroid #2:35 17

centroid #3:37 57

centroid #4:13 45

Trails: 2

4 4

4 4

4 4 44 4 4

4 4 44 4

4

4 4 4

4 4

44

4

4

44

4 44

2

2 2 2

2

2

2 2 3

2 2 3 3

2 3

3

2 3

3

3 3 3 3

3 3

3

3

1

1 1

1 1

1 1 1 1 1

1 1

1 11 1

11 1

1 1

1 1 1

1

1

1 1

1

1

Trails: 2 end here!

centroid #1:64 31

centroid #2:34 17

centroid #3:41 55

centroid #4:13 46