TÜRKİYE CUMHURİYETİ

YILDIZ TEKNİK ÜNİVERSİTESİ

BİILGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

K MEANS CLUSTERING / K-ORTALAMA KÜMELEME YÖNTEMİ

Öğrenci No: 20011610

Öğrenci Adı Soyadı: Fatih ALTINCI

Öğrenci Okul e-posta: l1120610@std.yildiz.edu.tr

Öğrenci Kişisel e-posta: fatihaltinci@hotmail.com

Ders/Grup: YAPISAL PROGRAMLAMAYA GİRİŞ / Gr-1

FİNAL PROJE RAPORU

Ders Yürütücüsü

Doç. Dr. Mehmet Fatih Amasyalı

Haziran, 2021

İçindekiler Tablosu

ALGORİTMA

TANIM

- Şimdi öncelikle kümeleme nedir onunla başlayalım.
- Kümeleme veri seti içerisindeki benzer ilişkileri ve yapıları bulur ve benzer özellikleri taşıyan verileri gruplar. Gruplama yapılırken etiketler bilinmediği için, kümeleme analizi denetimsiz ya da danışmasız öğrenme çeşididir.
- Kümeleme: Benzer veri noktalarının aynı gruplarda yer alacak şekilde ayrıştırılmasıdır.
- Kümeleme algoritmaları verinin özelliklerinden öğrenerek en iyi bölünmeyi bulmaya çalışır.
- K-Means algoritması verideki belirli bölgeleri temsil eden küme merkezlerini bulmaya çalışır. Bu küme merkezleri kümeye ait bütün noktaların aritmetik ortalamasıdır. Verideki her nokta, kendi küme merkezine diğer kümelerin merkezinden daha yakındır. Bu iki şartı sağlamak için, algoritmanın 2 adımı tekrar eder.

İŞLEYİŞİ

- 1- Ayrılacak küme sayısı kadar nokta belirlenir ve bu noktalar merkez kabul edilir. Bu merkeze en yakın noktaların ortalaması bulunur. Bulunan ortalamalar merkez kabul edilerek küme merkezleri güncellenir.
- 2- Tekrar bu merkezlere yakın noktalar bulunur ve bunların ortalaması alınır. Bu ortalamalar merkez olur. Tekrar küme merkezleri güncellenir ve bu adımlar küme merkezi değişmeyene kadar devam eder.

UYGULAMA ALANLARI

- Peki bu algoritmayı nerelerde kullanabiliriz? Aşağıda birkaç örnek kullanım alanı verdim.
- Belgeleri Sınıflandırma
- Suç Yerlerinin Belirlenmesi
- Müşteri Segmentasyonu
- Oyuncu Analizi
- Dolandırıcılık Tespiti
- Çağrı Kaydı Detay Analizi
- BT Uyarılarının Kümelenmesi

RAKİP ALGORİTMALAR

- Aynı amaç için kullanılan, rakip veya alternatif algoritmalar nelerdir? Benzer sayılabilecek fakat farklı teknikleri kullanan algoritmalardan bazıları bunlardır:
- Single-Linkage Clustering
- Fuzzy C-Means
- Flame Clustering
- SUBCLU
- Ward's Method
- WACA Clustering

AVANTAJLAR/DEZAVANTAJLAR

Daha iyi özelliklere sahip diğer kümeleme algoritmaları daha pahalıya gelir. Pahalıdan kasıt algoritmanın çalışma hızı ve bellek yoğunluğudur. Bu durumda, k-means, alanı diğer kümeleme algoritmalarının uygulanabileceği ayrık daha küçük alt alanlara indirgeyerek ön kümeleme için harika bir çözüm haline gelir.

AVANTAJLAR

- Uygulaması nispeten daha basittir.
- Büyük veri kümelerini ölçekler.
- Yakınsamayı garanti eder.
- Merkezlerin konumlarını yarı otomatik başlatma ile yapabilir.
- Eliptik kümeler gibi farklı şekil ve büyüklükteki kümelerin belirli özelliklerini genel yönleri ile vurgular.

DEZANAVANTAJLAR

- K'yi elle seçmek
- Başlangıç değerline bağımlı olmak
- Değişen boyut ve yoğunlukta kümeleme verileri
- Kümelenmeye aykırı değerler
- Boyut sayısı ile ölçekleme

ANALİZ

KARMAŞIKLIK

Algoritmanın, çözülecek probleme göre nasıl kullanıldığına göre değişir. K Means dediğimiz k değişkenini belirleyerek yapılan kümeleme algoritmalarının birden çok varyantı var. Genel olarak Lloyd'un algoritmasının ve çoğu varyantının karmaşıklığı.

O (n * K * I * d)

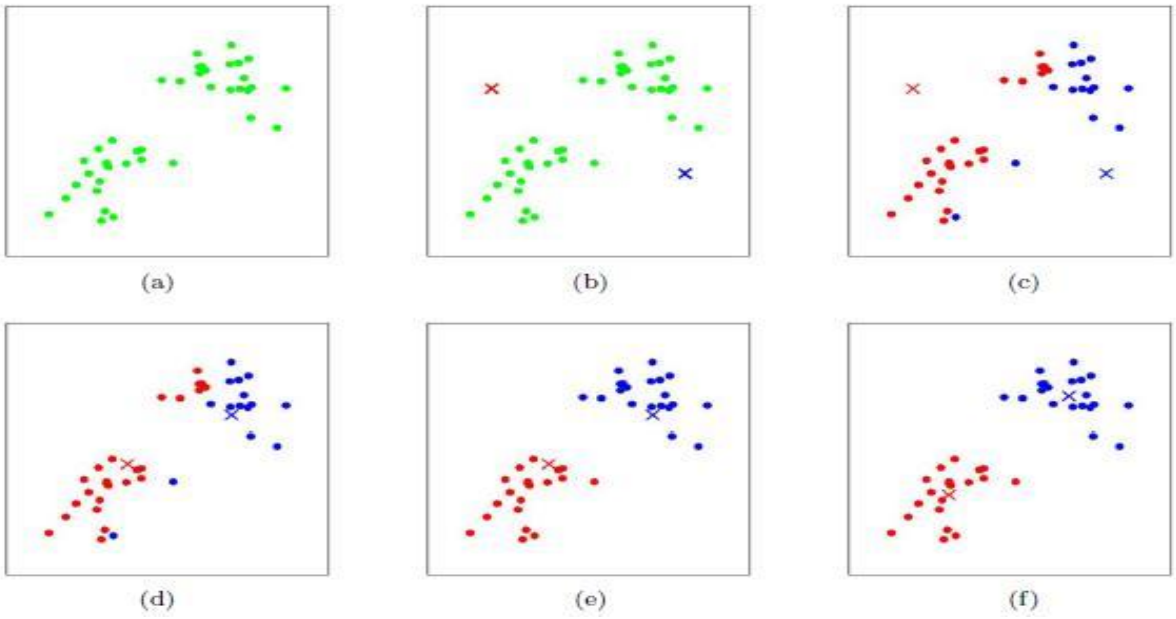
n : noktaların sayısı

K : kümelerin sayısı

I : iterasyon sayısı

d : nitelik sayısı

UYGULAMA



- Şekilde a, orijinal nesne noktasını gösteriyor. Şekil b'de önce k'nın 2 olduğunu varsayıyoruz, yani örnek uzay alanında rastgele iki koordinat noktası seçiyoruz ve ardından her noktayı hesaplıyoruz. İkisi arasındaki mesafe, şekil c'de gösterildiği gibi iki kategoriye ayrılıyor ve ardından merkezi koordinat noktaları her nesne noktası kümesinde sayılır, böylece d grafiğinin iki yeni koordinat noktasını elde ederiz. Böyle yinelemeli bir süreçte, nihai sonuç istikrarlı olma eğilimindedir ve sonunda şekil f'de gösterildiği gibi iki kategori elde ederiz.

- İlk olarak, 6*6 boyutlarında iki boyutlu bir dizinini oluşturulduğunu varsayıyoruz. Rastgele bir sayı üreterek, rastgele değerlere sahip iki boyutlu bir dizi oluşturulur. Bu değerler, 06 gibi, bunun ilk nesne noktası olduğunu gösterir. Altıncı noktanın nesne noktası olduğunu, ve eğer 00'sa, bu koordinatın nesne noktası olmadığını belirtir.
- Ardından, 2 kategoriye ayırarak d1 ve d2 dizisi, her nesne noktası ve 2 koordinat noktası arasındaki mesafeyi hesaplayarak ilk sınıflandırma için iki başlangıç koordinat noktası belirleriz.
- Ardından, yeni iki koordinat noktasının elde edilebilmesi için sırasıyla d1 dizisindeki ve d2 dizisindeki nesne noktaları kümesinin merkez değer koordinatlarını buluruz.
- Son olarak, yinelemeli algoritma aracalığıyla, artık değiştirilmeyen yani nihai koordinat noktalarını bulunur ve sınıflandırma tamamlanır.

C PROGRAMLAMA KODU

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h> // Zaman fonksiyonları için kullandığımız kütüphane

#include <math.h> // Matemiksel fonksiyonlar için kullandığımız kütüphane


int main(){

   clock_t t;

   t = clock();

   int i,j; // İterasyon değişkenlerimizi tanımladık.

   int a[6][6]; // 2 boyutlu dizimizi tanımladık.

   srand(time(0)); // zamanı seed olarak kullandık ve her seferinde farklı sayılar ürettik.

   for(i=0;i<6;i++)

   {

      for(j=0;j<6;j++)

// rastgele oluşturulmuş sayı

         a[i][j]=0+rand()%2; //rand() fonk ile üretilen sayıların aralığı [a,b)'de şu şekilde yazılır: a[i]=a+
Rand()%(ba);

   }


   for(i=0;i<6;i++)
```

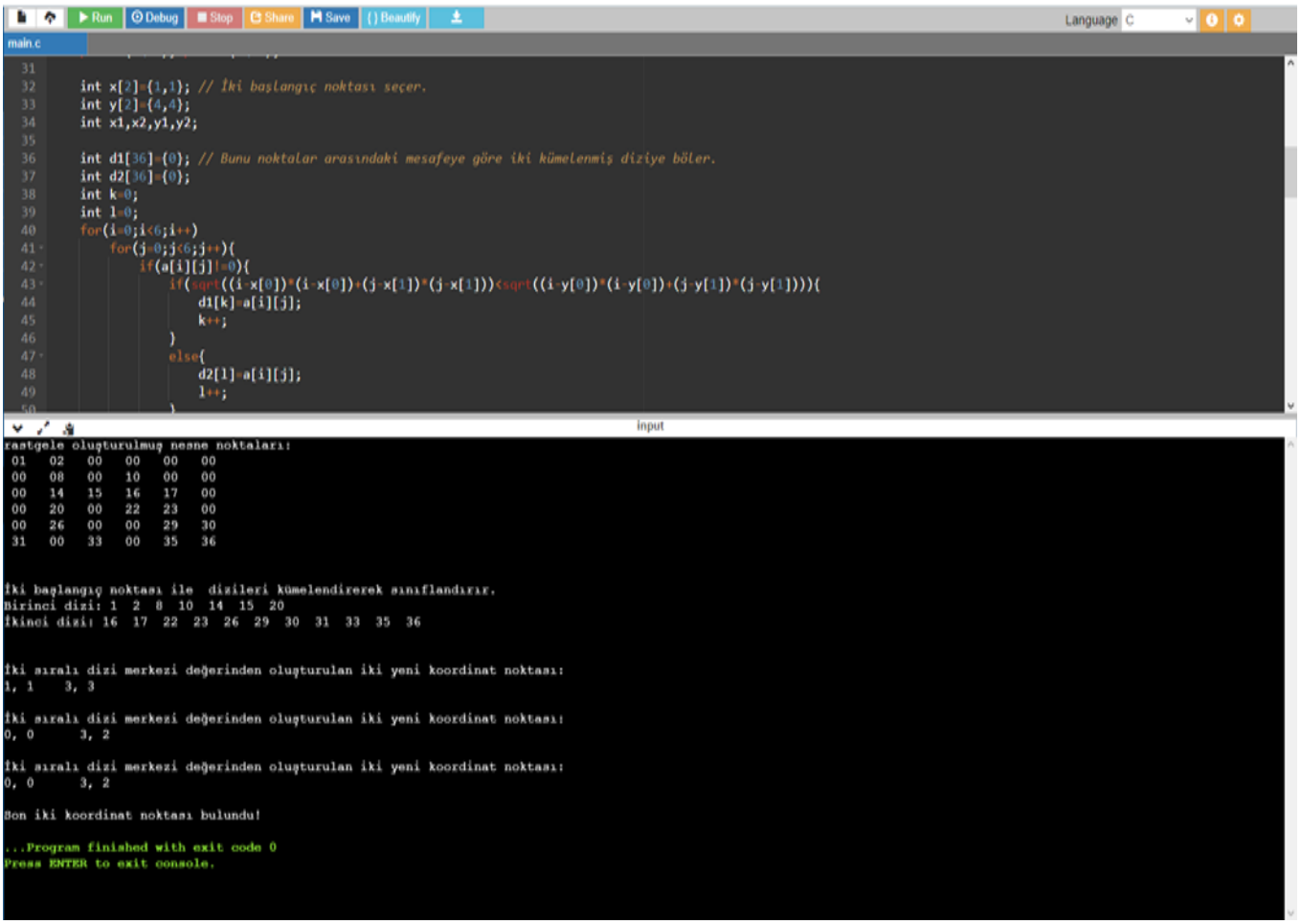
```
   for(j=0;j<6;j++){
      if(a[i][j]==1)
         a[i][j]=i*6+j+1;
   }
printf("rastgele oluşturulmuş nesne noktaları: \n");
for(i=0;i<6;i++)
   for(j=0;j<6;j++){
      if(j!=5)
         printf(" %02d  ",a[i][j]);
      else
         printf(" %02d  \n",a[i][j]);
   }
printf("\n"); printf("\n");

int x[2]={1,1}; // İki başlangıç noktası seçer.
int y[2]={4,4};
int x1,x2,y1,y2;

int d1[36]={0}; // Bunu noktalar arasındaki mesafeye göre iki kümelenmiş diziye böler.
int d2[36]={0};
int k=0;
int l=0;
for(i=0;i<6;i++)
   for(j=0;j<6;j++){
      if(a[i][j]!=0){
         if(sqrt((i-x[0])*(i-x[0])+(j-x[1])*(j-x[1]))<sqrt((i-y[0])*(i-y[0])+(j-y[1])*(j-y[1]))){
            d1[k]=a[i][j];
            k++;
         }
         else{
            d2[l]=a[i][j];
```

```
            l++;
          }
        }
    }
printf("İki başlangıç noktası ile  dizileri kümelendirerek sınıflandırır. \n");
printf("Birinci dizi: ");
for(i=0;i<k;i++)
   printf("%d  ",d1[i]);
printf("\n");
printf("İkinci dizi: ");
for(i=0;i<l;i++)
   printf("%d  ",d2[i]);

printf("\n");
int o=0;int p=0;
int q; //Her noktanın satır numarası
for(i=0;i<k;i++){
   q=(d1[i]-1)/6;
   o=q+o;
   p=d1[i]-q*6-1+p;
}
x[0]=o/k;x[1]=p/k;
o=0;p=0;
for(i=0;i<l;i++){
   q=(d2[i]-1)/6;
   o=q+o;
   p=d2[i]-q*6-1+p;
}
y[0]=o/l;y[1]=p/l;
printf("\n");printf("\n");
printf("İki sıralı dizi merkezi değerinden oluşturulan iki yeni koordinat noktası: \n");
```

```
printf("%d, %d    ",x[0],x[1]);
printf("%d, %d \n\n",y[0],y[1]);

while(1){
   x1=x[0];x2=x[1];y1=y[0];y2=y[1];
   for(i=0;i<36;i++){
      d1[i]=0;d2[i]=0;}
   k=0;
   l=0;
   for(i=0;i<6;i++)
      for(j=0;j<6;j++){
         if(a[i][j]!=0){
            if(sqrt((i-x[0])*(i-x[0])+(j-x[1])*(j-x[1]))<sqrt((i-y[0])*(i-y[0])+(j-y[1])*(j-y[1]))){
               d1[k]=a[i][j];
               k++;
            }
            else{
               d2[l]=a[i][j];
               l++;
            }
         }
      }
   o=0;p=0;q=0;
   for(i=0;i<k;i++){
      q=(d1[i]-1)/6;
      o=q+o;
      p=d1[i]-q*6-1+p;
   }
   x[0]=o/k;x[1]=p/k;
   o=0;p=0;q=0;
   for(i=0;i<l;i++){
```

```
        q=(d2[i]-1)/6;
        o=q+o;
        p=d2[i]-q*6-1+p;
    }
    y[0]=o/l;y[1]=p/l;
    printf("İki sıralı dizi merkezi değerinden oluşturulan iki yeni koordinat noktası: \n");
    printf("%d, %d      ", x[0],x[1]);
    printf("%d, %d \n\n", y[0],y[1]);

    if(x1==x[0] && x2==x[1] && y1==y[0] && y2==y[1]){
        printf("Son iki koordinat noktası bulundu! \n");
        t = clock() - t;
        double time_taken = ((double)t)/CLOCKS_PER_SEC;
        printf("main() , %f saniyede çalıştı \n", time_taken);
        for(i=1;i<11;i++){
        printf("%d boyutlu dizi icin calisma zamani: \t",i); // 10 boyutlu diziye kadar çalışma zamanı
        for(j=1;j<=i;j++){

        printf("*");
    }
        printf("\n");

  }
        return 0;

    }

  }


}
```

EKRAN ÇIKTILARI



```
Run  Debug  Stop  Share  Save  { } Beautify                                  Language C
main.c
31
32      int x[2]={1,1}; // İki başlangıç noktası seçer.
33      int y[2]={4,4};
34      int x1,x2,y1,y2;
35
36      int d1[36]={0}; // Bunu noktalar arasındaki mesafeye göre iki kümelenmiş diziye böler.
37      int d2[36]={0};
38      int k=0;
39      int l=0;
40      for(i=0;i<6;i++)
41          for(j=0;j<6;j++){
42              if(a[i][j]!=0){
43                  if(sqrt((i-x[0])*(i-x[0])+(j-x[1])*(j-x[1]))<sqrt((i-y[0])*(i-y[0])+(j-y[1])*(j-y[1]))){
44                      d1[k]=a[i][j];
45                      k++;
46                  }
47                  else{
48                      d2[l]=a[i][j];
49                      l++;
50                  }
```

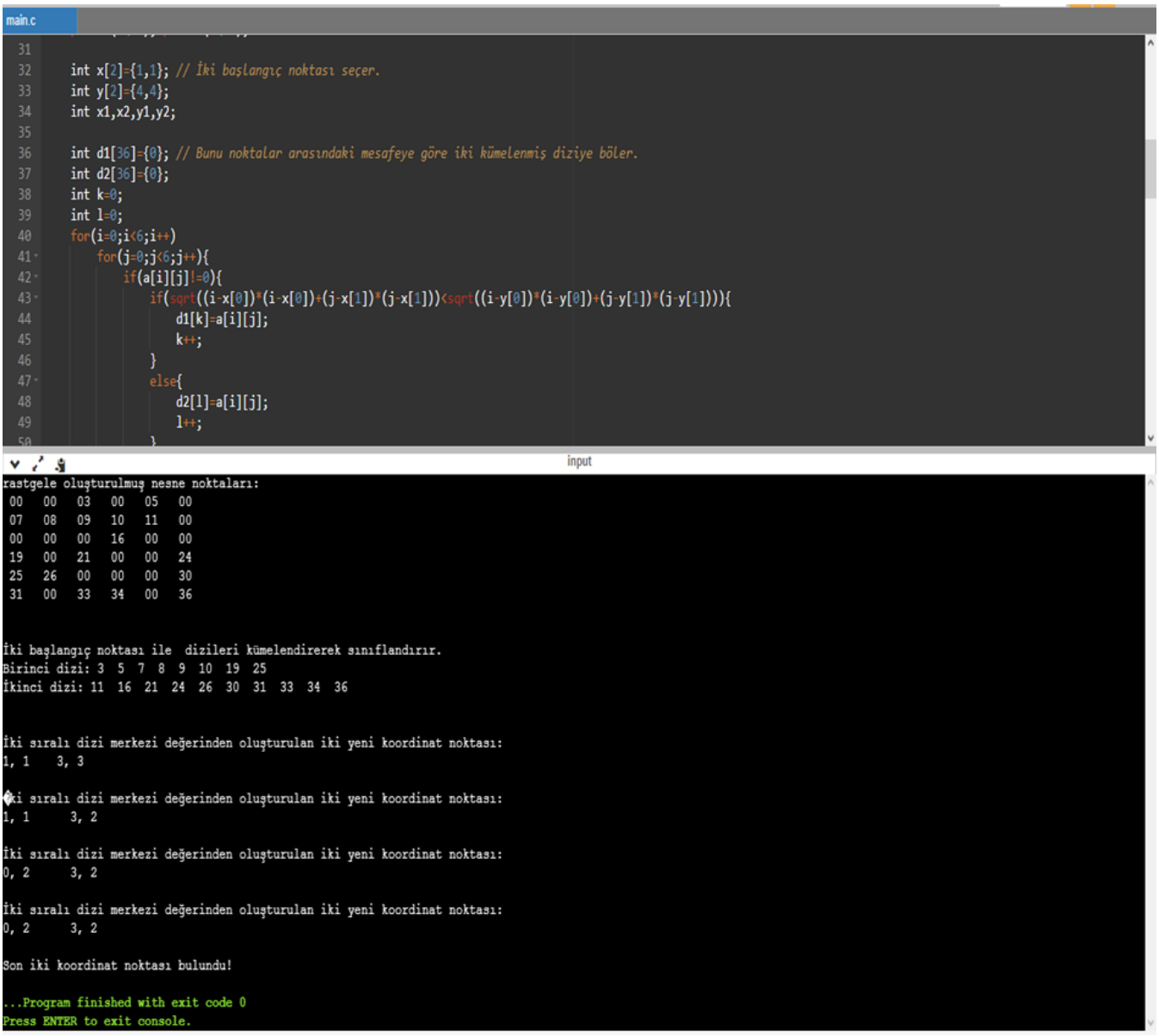
```
input
rastgele oluşturulmuş nesne noktaları:
 01   02   00   00   00   00
 00   08   00   10   00   00
 00   14   15   16   17   00
 00   20   00   22   23   00
 00   26   00   00   29   30
 31   00   33   00   35   36


İki başlangıç noktası ile  dizileri kümelendirerek sınıflandırır.
birinci dizi: 1  2  8  10  14  15  20
İkinci dizi: 16  17  22  23  26  29  30  31  33  35  36


İki sıralı dizi merkezi değerinden oluşturulan iki yeni koordinat noktası:
1, 1    3, 3

İki sıralı dizi merkezi değerinden oluşturulan iki yeni koordinat noktası:
0, 0      3, 2

İki sıralı dizi merkezi değerinden oluşturulan iki yeni koordinat noktası:
0, 0      3, 2

Son iki koordinat noktası bulundu!

...Program finished with exit code 0
Press ENTER to exit console.
```

main.c

```
78      printf("\n");printf("\n");
79      printf("İki sıralı dizi merkezi değerinden oluşturulan iki yeni koordinat noktası: \n");
80      printf("%d, %d    ",x[0],x[1]);
81      printf("%d, %d \n\n",y[0],y[1]);
82
83      while(1){
84          x1=x[0];x2=x[1];y1=y[0];y2=y[1];
85          for(i=0;i<36;i++){
86              d1[i]=0;d2[i]=0;}
87          k=0;
88          l=0;
89          for(i=0;i<6;i++)
90              for(j=0;j<6;j++){
91                  if(a[i][j]!=0){
92                      if(sqrt((i-x[0])*(i-x[0])+(j-x[1])*(j-x[1]))<sqrt((i-y[0])*(i-y[0])+(j-y[1])*(j-y[1]))){
93                          d1[k]=a[i][j];
94                          k++;
95                      }
96                      else{
97                          d2[1]=a[i][j];
98                          l++;
99                      }
100                 }
101             }
```

input

```
Rastgele oluşturulmuş nesne noktaları:
 00   02   03   04   00   06
 00   08   00   10   11   00
 13   14   00   16   00   18
 19   00   21   22   00   24
 00   00   00   28   29   30
 00   32   00   34   35   00


İki başlangıç noktası ile  dizileri kümelendirerek sınıflandırır.
Birinci dizi: 2  3  4  8  10  13  14  19
İkinci dizi: 6  11  16  18  21  22  24  28  29  30  32  34  35


İki sıralı dizi merkezi değerinden oluşturulan iki yeni koordinat noktası:
1, 1    3, 3

İki sıralı dizi merkezi değerinden oluşturulan iki yeni koordinat noktası:
1, 1      3, 3

Son iki koordinat noktası bulundu!

...Program finished with exit code 0
Press ENTER to exit console.
```



KAYNAKLAR

- Google Akademik – K Means Advantages and Disadvantages
- Wikipedia – K Means Clustering
- Youtube – Introduction to K Means Clustering
- Youtube – K Means Tekniği
- Geeksforgeeks – K Means Clustering Introduction

- Tureng – Yabancı Kelimelerin Türkçe Karşılıkları
- Stack Exchange
- Stack Overflow – K Means Clustering Time Complexity