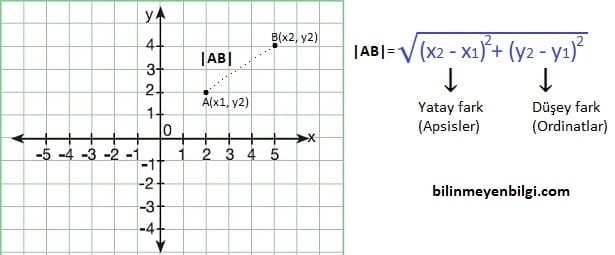
Kocaeli Üniversitesi

|  |
| --- |
| Bilgisayar Mühendisliği Bölümü  Programlama Laboratuvarı I  Minimum Çevreleyen Çember  Bünyamin Ekşici  170201014@kocaeli.edu.tr |



***2 boyutlu bir düzlemde verilen n tane noktanın koordinat düzleminde gösterilmesi ve bu noktaları çevreleyen minimum çemberi bulmak.***

***Anahtar Kelimeler: minimum çevreleyen çember, geometrik algoritma , koordinat sistemi***

**Projenin Özeti**

Programlama laboratuvarı 1 projesi olarak bizden “ Minimum Çevreleyen Çember ve B- spline ” adındaki bir uygulama yapılması istendi. Ben B-spline’ı yapamadım ve tabi ki matematiksel çözümlemesini yapamadım. O yüzden raporumdan B-spline dan bahsetmeyeceğim.

Ben proje için bizden istenen programlama dilinden C ve grafik kütüphanesinden Grapics.h kütüphanesini kullandım.

Projemizin isterleri üzerine kullanıcıdan alınacak olan x ve y koordinatlarını grafik kütüphanemizi kullanarak ekranda göstermeli ve bu noktaları kapsayan en küçük çember çizilmeliydi. Bende kullanıcıdan alınan noktaları input.txt dosyasından okuma yaparak ekranda gösterdim ve gerekli matematiksel mantık üzerine kurduğum algoritma ile çözüme ulaştım.

**I. GİRİŞ**

Projenin konusu verilen n tane noktayı minimum çevreleyen çemberi bulmak. Ben bu proje için OpenGl , Allegro , Grapics kütüphanelerinden grapics.h kütüphanesini tercih ettim. Bu kütüphanenin isteğime daha uygun ve kullanışlı olduğunu düşündüm.

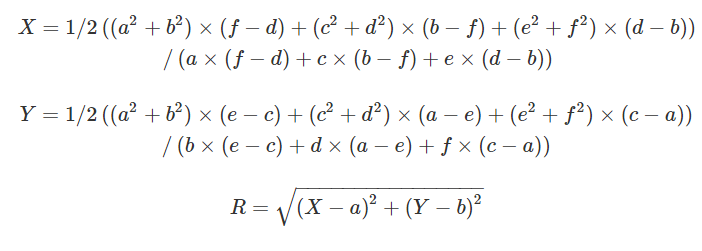
**II. yöntem**

Bu projede kullanılan yöntem 4 aşamada anlatılacaktır.

1.Aşama: Tabi ki ilk önce noktalar alınmalı ve bu noktaların koordinat sisteminde gösterilmeli. Sonra bütün bu noktaların birbirine olan uzaklığını bulmalıyız bunun içinde iki nokta arası uzaklık formülünü kullanırız.

Bütün noktaların birbirine olan uzaklığını bulduktan sonra en uzak iki noktayı seçeriz. Bu noktalar P1 ve P2 olsun. Bu iki noktanın üzerinden geçen bir çember çizeriz. Eğer verilen n noktanın hepsini çevreliyorsa minimum çevreleyen çember bulunmuştur. Aksi halde yani çemberin dışında kalan nokta var ise 2. Aşamaya geçilir.

2.Aşama: 1.aşamada düşündüğümüz çemberin merkezinden çemberin dışındaki en uzak nokta hesabı yapılır yani yine 1.aşamada olduğu gibi iki nokta arası uzaklık formülü kullanarak yapılır. Bu merkeze en uzak nokta P3 olsun. Şimdi ise bu 3 noktadan (P1(a,b) , P2(c,d) , P3(e,f)) bir çember çizilir ve bu çemberin merkezinin formülü ve yarıçapı aşağıdaki gibidir.

 Çizilen bu çember tüm noktaları içeren bir çember ise minimum çevreleyen çember bu çemberdir. Aksi halde 3.aşamaya geçilir.

3.Aşama: Bu aşamada 2.aşamadaki 3 nokta üzerinden geçen çemberin dışında kalan noktaların aşama 2 deki gibi merkeze en uzak nokta seçilir. Bu nokta P4 olur. Şu durumda minimum çember için P3,P4 ün üzerinden geçtiği ve 3.nokta olarakta P1 veya P2’nin üzerinden geçen çemberlerden hangisinin minimum olduğuna karar verilir. Yani elimizde 2 çember var P1 P3 P4 veya P2 P3 P4 peki hangi çember minimum 4.aşamada karar vereceğiz.

4.Aşama: Bu aşamaya gelindiyse elimizde 2 çember var ve bu çemberlerden P1 barındıran mı yoksa P2 barındıran çemberin mi minimum çember olduğunu bulacağız.

Bunun için P1 in P3 ve P4’e , P2’ninde P3 ve P4 e olan uzaklığı iki nokta arası uzaklık formülü ile hesaplanır. En kısa uzaklığı barındıran iki nokta bulunur. Bu iki noktadan birisi P1 ise minimum çember P2 P3 P4 ün üzerinden geçen çemberdir. Yine bu iki noktadan birisi P2 ise minimum çevreleyen çember P1 P3 P4 ün üzerinden geçen çemberdir. Bu çemberin yarıçapı ve merkezi aşama 2 deki üç noktadan geçen çemberin merkez ve yarıçapı formülünden bulunur.

Oluşturulan fonksiyonlar:

* iki\_nokta\_arasi\_uzaklik(double x1,double x2,double y1,double y2)
* uc\_noktasi\_verilen\_cemberin\_merkezi\_x(double a, double b, double c, double d, double e,double f )
* uc\_noktasi\_verilen\_cemberin\_merkezi\_y(double a, double b, double c, double d, double e, double f )

Kullanılan grafik kütüphanesi: Grapics.h

**SÖZDE KOD**

1-BAŞLA

2-OKU(input.txt)

3-NOKTALARI DİZİYE YERLEŞTİR

4-NOKTALAR ARASINDAKİ UZAKLIĞI BUL

5-EN UZAK 2 NOKTAYI SEÇ P1 VE P2

6- geçici\_merkez =SEÇİLEN NOKTANIN ORTA NOKTASINI

gecici\_yarıcap= iki\_nokta\_arası\_uzaklık(2 nokta )/2

7-IF(gecici\_yaricap>iki\_nokta\_arası\_uzaklık(gecici\_merkez,noktalar)){

YAZ(gecici\_yaricap)

YAZ(geçici\_merkez)

}

8-MERKEZE YARICAPTAN BÜYÜK EN UZAK NOKTAYI BUL BU NOKTA P3

9-P1 P2 P3 NOKTLARINDAN GEÇEN ÇEMBERİ CİZ

MERKEZİ HESAPLA

uc\_noktasi\_verilen\_cemberin\_merkezi\_x(P1 P2 P3)

uc\_noktasi\_verilen\_cemberin\_merkezi\_y(P1 P2 P3)

geçici\_merkez E ATA

gecici\_yaricap = iki\_nokta\_arası\_uzaklık(gecici\_merkez,P1)

10- IF(gecici\_yaricap>iki\_nokta\_arası\_uzaklık(gecici\_merkez,noktalar)){

YAZ(gecici\_yaricap)

YAZ(geçici\_merkez)

}

11- MERKEZE YARICAPTAN BÜYÜK EN UZAK NOKTAYI BUL BU NOKTA P4

12-P1 İLE P2 NİN P3 VE P4 E UZAKLIKLARINI BUL

13- IF EN KUCUK UZAKLIK P1 – P3 || P1-P4 İSE {

uc\_noktasi\_verilen\_cemberin\_merkezi\_x(P2 P3 P4)

uc\_noktasi\_verilen\_cemberin\_merkezi\_y(P2 P3 P4)

geçici\_merkez E ATA

gecici\_yaricap = iki\_nokta\_arası\_uzaklık(gecici\_merkez,P2)

YAZ(gecici\_yaricap)

YAZ(geçici\_merkez)

}ELSE İF EN KUCUK UZAKLIK P2 – P3 || P2-P4 İSE {

uc\_noktasi\_verilen\_cemberin\_merkezi\_x(P1 P3 P4)

uc\_noktasi\_verilen\_cemberin\_merkezi\_y(P1 P3 P4)

geçici\_merkez E ATA

gecici\_yaricap = iki\_nokta\_arası\_uzaklık(gecici\_merkez,P1)

YAZ(gecici\_yaricap)

YAZ(geçici\_merkez)

}

ELSE

YAZ(EROR)

14-BİTİR

**III. KARMAŞIKLIK ANALİZİ**

.

Algoritmanın karmaşık analizine baktığımızda döngü sayısına vs.

Zaman karmaşıklığı(time complexity) = T(n)

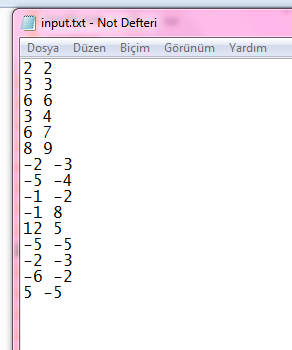
n^2 + 10n +c

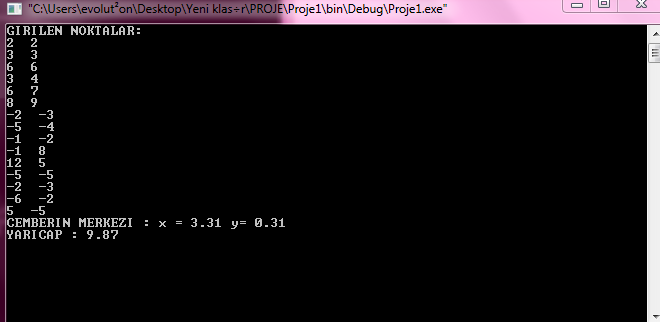
c=işlem sayısı

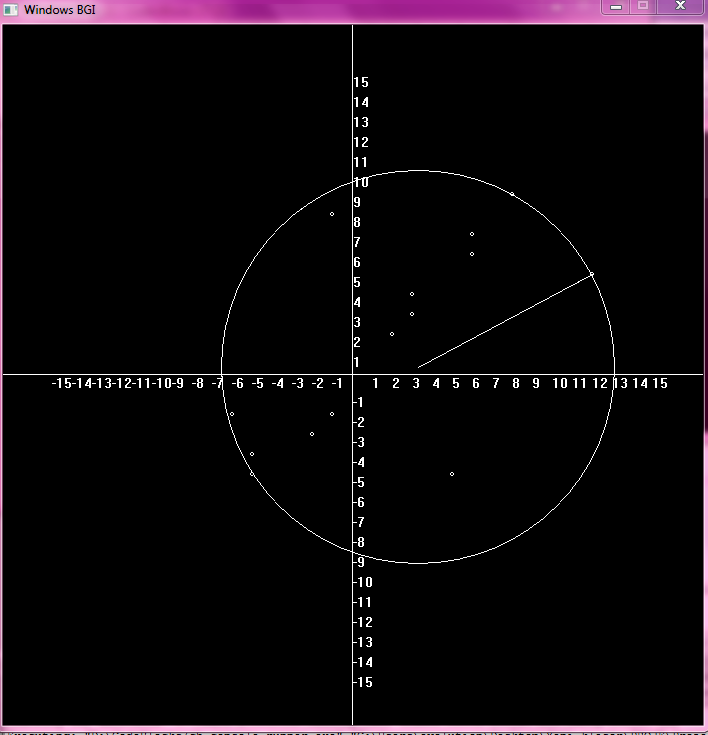
ve bu zaman karmaşıklığının Big O analizi ise

O(N^2)’dir.

**IV. DENEYSEL SONUÇLAR**



****

****

**V. sonuç**

Bu proje sayesinde c programlama dilinde kendime değer kattım özellikle grafik konusunda kendimi iyileştirdiğimi düşünüyorum. Aynı zamanda bir kütüphane nasıl indirilir, kurulur ve fonksiyonları nasıl araştırılır öğrendim. Son olarakta bir programın hangi aşamalardan geçtiğini ve ne kadar emek harcandığını gördüm. Teşekkür ederim.

**KAYNAKLAR**

1-https://bilinmeyenbilgi.com/wp-content/uploads/2019/05/iki-nokta-arasindaki-uzaklik.jpg

2- A new heuristic algorithm for the planar minimum covering circle problem.pdf

<https://www.tandfonline.com>

3- Palme Yayıncılık C İle Programlama kitabı

4- wikipedia

<https://en.wikipedia.org/wiki/Smallest-circle_problem>