**Minimum Çevreleyen Çember & B Spline Projesi**

***KOCAELİ ÜNİVERSİTESİ***

***BİLGİSAYAR MÜHENDİSLİĞİ(iö)***

Ali Atilla Aydemir Mehmet Alperen Onarır

***180202063*** 180202083

***atilla-734@hotmail.com*** [mehmetalperen95@gmail.com](mailto:mehmetalperen95@gmail.com)

***ÖZET***

***Bu projenin amacı; belirli bir txt dosyası üzerine kullanıcı tarafından girilen tam sayı kordinatları ile çizilebilecek minimum çevreleyen çember bilgilerini hesaplama ve ekrana yazma(a) , verilen kordinat bilgileri doğrultusunda noktaların en yakınından geçen eğriyi hesaplama(b) , hesaplanan verileri ve girilen noktaları kordinat düzleminde çizmektir.***

***GİRİŞ***

***C dili, Dev++ IDE ‘si ve graphics.h arayüz kütüphanesi yardımı ile yazılmış programımızda kullanıcı ilgili klasörün içerisindeki “noktalar” txt uzantılı dosyaya işletmek istediği noktaları girerek minimum çevreleyen çember ve B-spline’na ait bilgi ve görüntülere erişebilir. Kordinat bilgileri tam sayı olarak ve -+20 aralığında girilmelidir.***

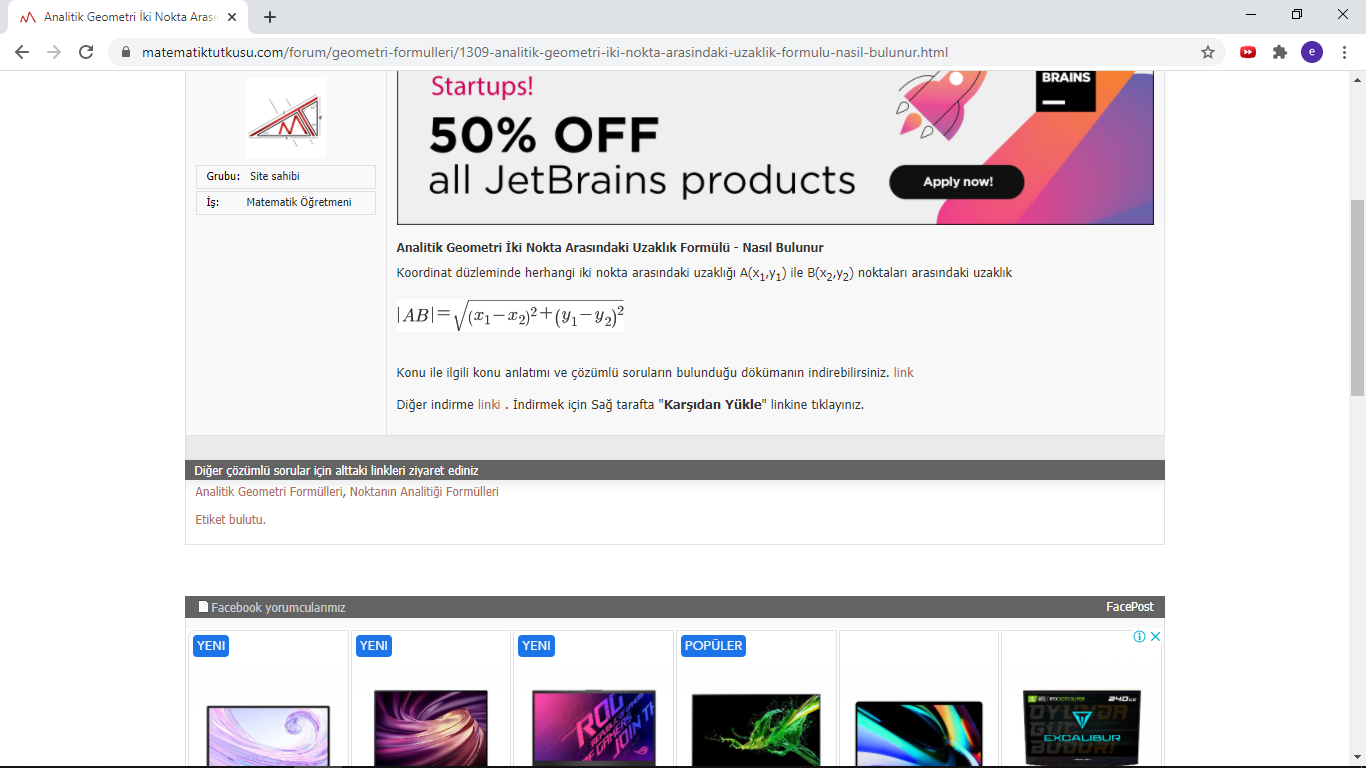
***YÖNTEM***

1. ***Minimum çevreleyen çember(mçç)***

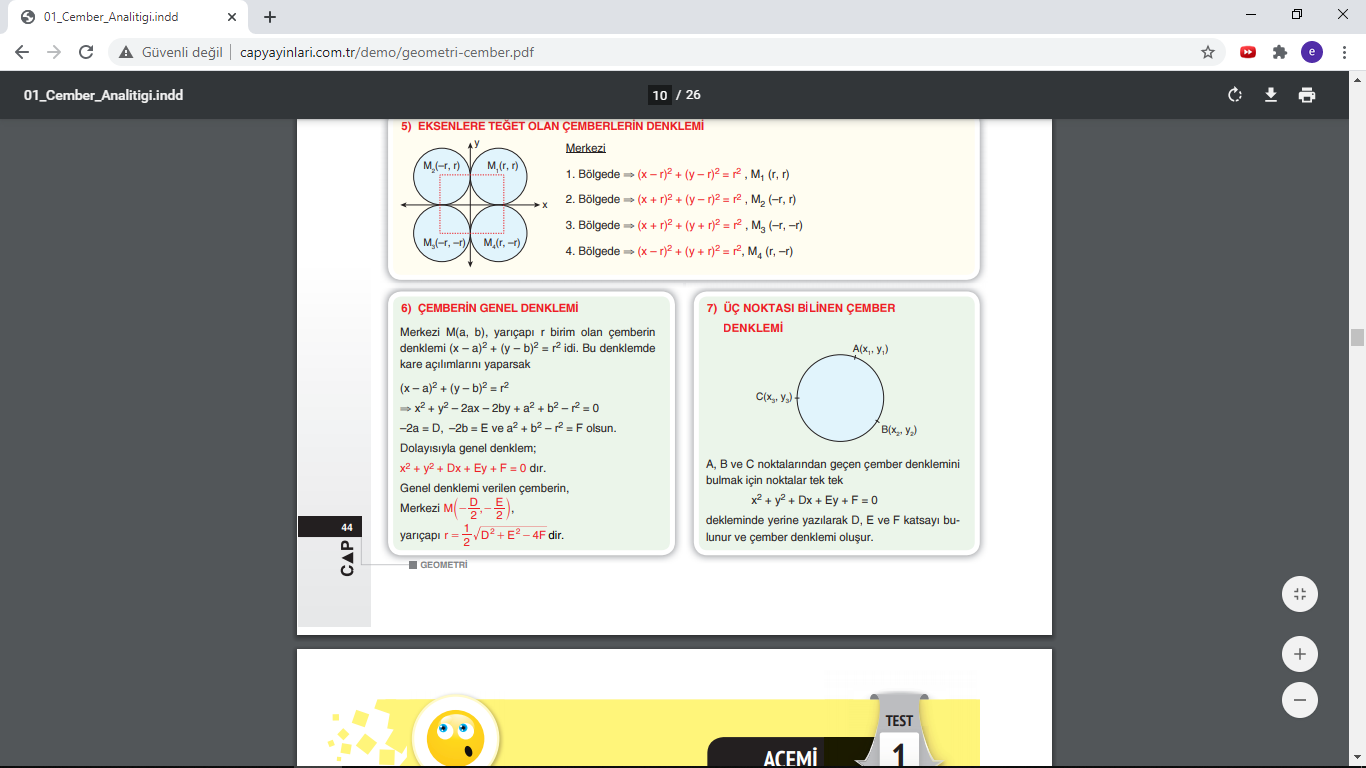
***Mçç hesaplaması yapılırken 3 kabul ve 2 geometri formulü kullanılmıştır.***

1. ***Kabul, çizilen çevreleyen çemberin çapı verilen noktaların 2 veya daha fazlasının birbiri ile olan mesafelerine eşit ise bu çember daha fazla küçülemez.***
2. ***Kabul, çizilen çevreleyen çember 1 nokta ile kesişiyorsa bu çember küçülebilir.***
3. ***Kabul, çizilen çevreleyen çember 1. Kabul şartlarına uygun değil ise küçülebilir***

***KULLANILAN GEOMETRİ FORMULLERİ***

***- iki nokta arası uzakluk formulü*** ***1***

***-Genel çember denklemi***

***2***

***Karşılaşılan promlem; denklemde /0 ifadesi olur ise hesap yapılamamaktadır. Aynı 3 nokta, denkleme farklı sıralarla gönderirilse bu problem büyük ölçüde ortadan kalkmaktadır. Bu sebeple kombinasyon değil permitasyon işlemi tercih edilmiş, 3lü noktaların sıralaması da işleme dahil edilmiştir.***

***Algoritma mantığı : 1.adım verilen N noktanın içerisindeki en büyük çap hesaplanır ve kapsayıcılığı kontrol edilir. Eğer kapsayıcı ise 1. Kabul gereği daha fazla küçültülemez ve mçç olarak kabul edilir.***

***2. adım 1. Adımda mçç bulunmaz ise, 3lü olarak çizilebilecek bütün çemberlerin içerisinden kapsayanlar stoklanır, stokların içerisinden en küçük yarıçapa sahip çember seçilir ve mçç olarak kabul edilir.***

***Deneysel sonuçlar- Girdi:****arr [] [] = {{0, 0}, {0, 1}, {1, 0}}* ***Çıktı:****Merkez = {0.5, 0.5}, Yarıçap = 0.7071*

***Girdi:****dizi [] [] = {{5, -2}, {-3, -2}, {-2, 5}, {1, 6}, {0, 2}}* ***Çıktı:****Merkez = {1.0, 1.0}, Yarıçap = 5.000*

1. ***B-spline***

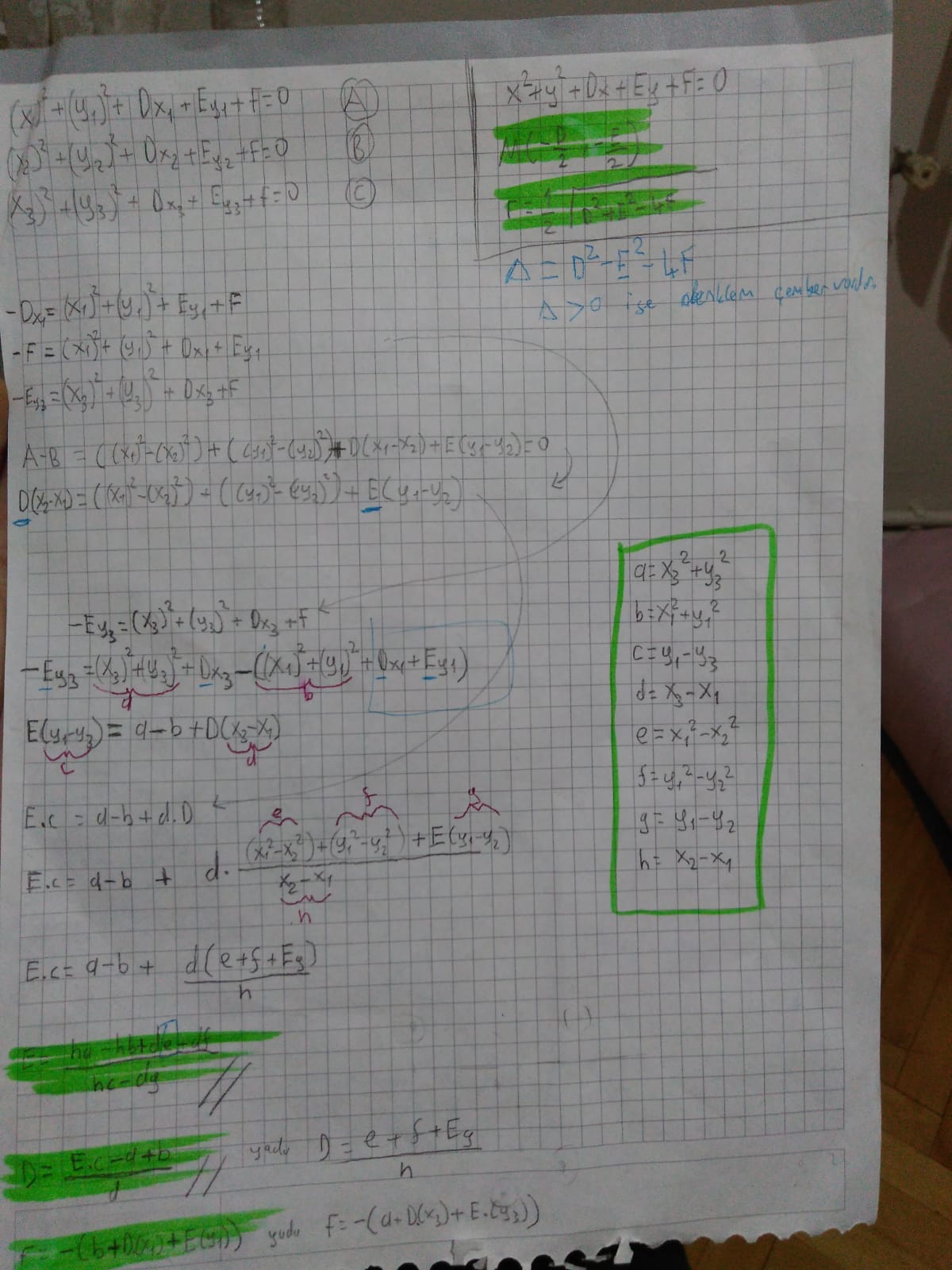
B-spline, bezier eğrilerinin özelleştirilmiş halidir. B-spline için noktaların yoğunluğuna göre, yönelim mekanizması oluşturulur. De Boor'un algoritması şu formülü temel almıştır:  Projede x ve y katsayılarına bağlı olarak, yukarıdaki artan yapıya esas olarak hazırlanan şu yapı kullanılmıştır:

 Bu formül p0 ve p3 noktaları arasında çizilecek, p1 ve p2 ile kavislenecek olan eğriyi belirtir. t değişkenin 0 ve 1 arasında olma sebebi eğrinin belli sınırlar dahilinde kalması dolayısıyladır. böylelikle katsayılar hızlı büyüyüp sınır dışına taşmazlar.

pow(1-t,3)\*dizix[i] + 3\*t\*pow(1-t,2)\*ortx[i]\*1.2 + 3\*t\*t\*(1-t)\*ortx[i]\*1.2 + pow(t,3)\*dizix[k];

Kod olarak yazdığımız bu formül kübik bezier yaklaşımının b-spline üzerindeki gösterimidir. B-spline dediğimiz eğriler 4 nokta ve üzeri meydandayken, kübik bezier eğrilerine uygulanan yöntemle oluşturulurlar. Kodumuzda önce dizileri sıralandı, daha sonra sıraladığımız bu dizilere orta noktalar atayarak “lokal” kontrol noktaları oluşturduk. P0 ve P3 arasında çizilecek eğrimiz P1 ve P2 orta noktaları ile kavis alarak birbirlerini tamamladılar ve asla kapalı bir cisim oluşturmayacak şekilde noktaların üzerlerinden geçtiler. Ayrıca b-spline eğrilerinin bezier eğrilerden farklı olarak global değil lokal, yani bütün cisim üzerinde değil de her 2 nokta arasında belirlenen mesafe üzerinde etkili olması, koda pratiklik katmış ve daha hızlı çalışmasına sebebiyet vermiştir. Bu eğri çizdirme işlemi t değişkeninin 0.001 gibi küçük bir aralıkla artması ve bunun ekrana pikseller olarak çizilmesiyle oluşur.



******

***\*Genel çember denkleminin koda uygun hale getirilmesi ile verilen 3 noktadan çizilebilecek çemberin yarıçapı ve merkez kordinatlarına erişim sağlanmıştır***

***KARMAŞILIK ANALİZİ VE KABA KOD***

***START***

***Struct noktalar() int x,y***

***Nokta\_okuma() file\*dosya fopen “noktalar.txt”***

***Fscanf &noktalar(i).x and y, sayac++ return sayac***

***İkili\_en\_büyük\_cap(noktalar) hesapla->***

***ikili max mesafe(R),Merkez kord(x,y)***

***Kapsama\_kontrol(R,x,y)***

***for 0 to N hesapla merkez-nokta(Ni) mesafe->R2***

***karşılaştır R –R2***

***if kapsama 1-> mçç bulundu print( mçç)***

***else break***

***kordinat\_sistemi\_noktalar (N) for 1 to N putpixel(Ni)***

***if(kapsama 0) ->***

***uclu\_kombinasyon\_sayısı\_bulma()***

***hesapla ve ata-> C= C(N,3)***

***uclu kombinasyon olusturma()***

***for i=1 to N***

***for j=1 to N***

***for k=1 to N***

***uc\_noktada\_cember\_denk(***

***xi = noktalar().x***

***yi = noktalar().y***

***xj = noktalar().x***

***yj = noktalar().y***

***xk = noktalar().x***

***yk = noktalar().y***

***)***

***Uc\_noktada\_cember\_denk(xi,yi,xj,yj,xk,yk)***

***İf delta> 0***

***Hesapla R ve merkez kord(xiy)***

***Else continue***

***FLAG=0;***

***For(i=1 to N) hesapla merkez – nokta mesafe -> R2***

***İf R2> R FLAG++***

***İF FLAG=0 stok\_dizi() = R,x,y;***

***STOK\_NO++;***

***For(1 to STOK\_NO) hesapla min R***

***Printf (minR x,y)***

***B spline-----***

***dizix[a]=noktalar[a].x;***

***diziy[a]=noktalar[a].y;***

***float ortx[20];***

***float orty[20];***

***hesapla for i = 1 toN***

***ortx[q]=(dizix[q]+dizix[q+1])/2;***

***orty[q]=(diziy[q]+diziy[q+1])/2;***

***for t=0.0 to 1 t+=0.0001***

***putpixel = bspline genel forumul***

***Karmaşıklık analizi : Mavi ile taralı kısımın içindeki sarı ile işratelenmiş döngüler en büyük üstel ifadeyi oluşturmaktadır. Programın diğer adımlarının karmaşıklık hesabı ekte verilmiş olup en büyük derece dışındaki hesaplamalar kural gereği gözardı edilmiştir. (n2=n kare, n3 = n küp, n4 = n üzeri 4.)***

***Nokta okuma -> O(n)***

***Kordinat\_sistemi\_noktalar -> O(n+600.1 +600.1 +n + n2)->O(n2)***

***Main() -> O(n)***

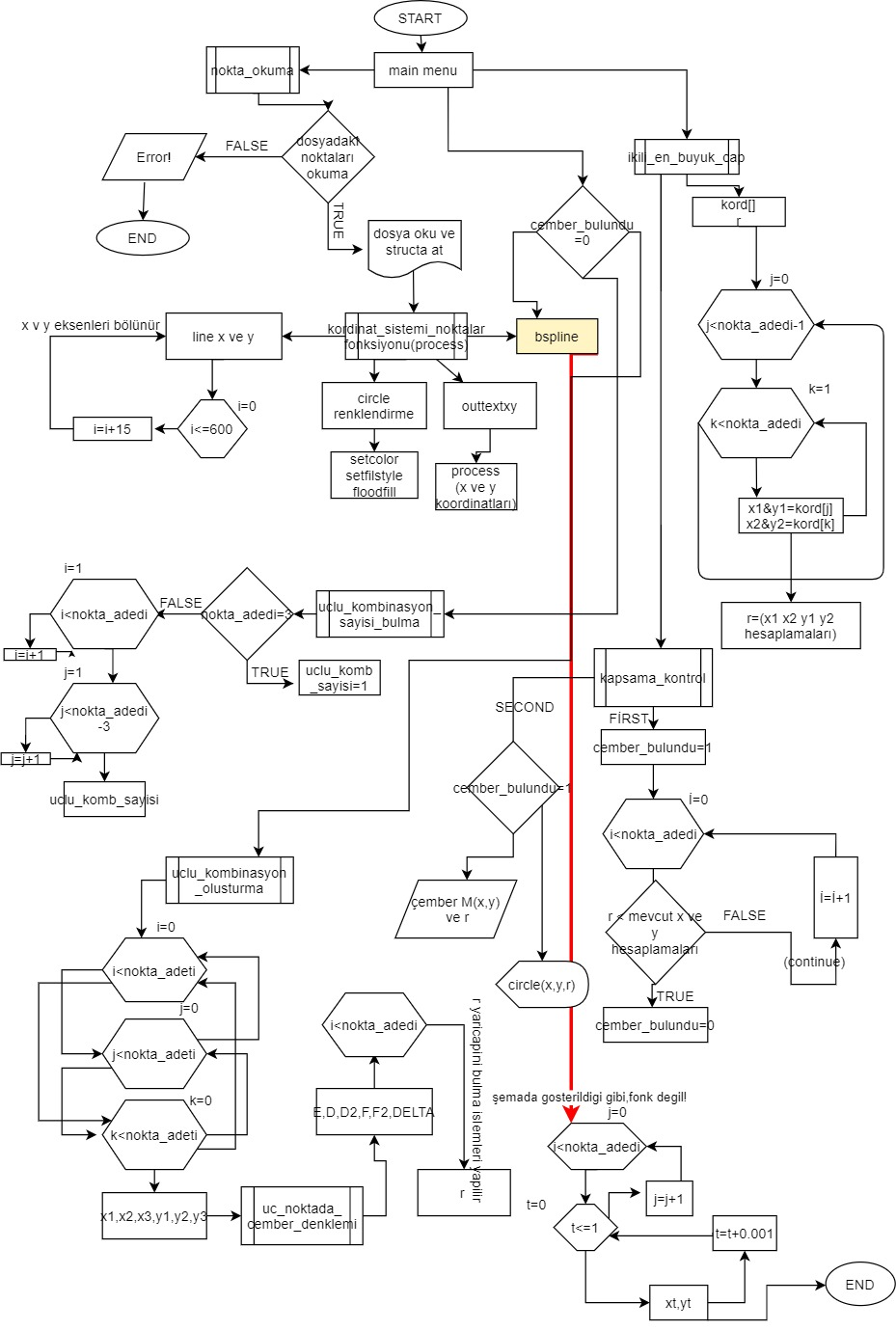
***İkili\_en\_büyük\_cap (kapsama kontrol() ) -> 0(n3)***

***Uclu\_kombinasyon\_sayisi\_bulma() -> O(n)***

***Uclu\_kombinasyon olusturma ( uclu\_cember\_denk() ) -> O(n4)***

***SONUÇ :***

***3lü permitasyon ilebelirlenen noktalar çember denklemine aktarılırken /0 ihtimali azaltılmış bununla beraber kodun karmaşıklığı artmıştır. /0 ihtimalinin daha da azaltılması için alternatif çember denklemleri kodda belirtilen bölgeye geliştirici mühendisler tarafından eklenebilir. Fakat bu işlemlerin karmaşıklık maliyeti olarak programa yansıyabileceği gözardı edilmemelidir.***



***\*AKIŞ DİYAGRAMI***

**KAYNAKÇA**

1. <https://www.matematiktutkusu.com/forum/geometri-formulleri/1309-analitik-geometri-iki-nokta-arasindaki-uzaklik-formulu-nasil-bulunur.html>
2. [***http://capyayinlari.com.tr/demo/geometri-cember.pdf***](http://capyayinlari.com.tr/demo/geometri-cember.pdf)
3. <https://en.wikipedia.org/wiki/B-spline>
4. https://www.geeksforgeeks.org/minimum-enclosing-circle-set-1/