

MINIMUM ÇEVRELEYEN ÇEMBER-B SPLİNE

Emre SEVİNDİK, Emir AVCI
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
emre.sevindik99@gmail.com, emr9229@gmail.com

Özet

Dosya işlemlerinden yararlanarak sayısı belirsiz koordinat noktalarının dosyadan okuması gerçekleştirildi. Okunan bu noktalar boyutu dinamik bir dizide saklandı.

Matematiksel yöntemlerden yararlanarak en uzak noktayı bulduk ve bu noktalardan geçen ve diğer noktaları içine alan çember graphics.h kütüphanesi yardımı ile çizilerek çemberin yarıçapı ve merkez noktası hesaplandı ayrıca bu noktaların üzerinden geçen bir eğri Bernstein polinomları yöntemi kullanılarak çizildi. Çizilen bu eğri yine graphics.h kütüphanesi kullanılarak arayüzde gösterildi.

Giriş

Çalışmada amaç kullanıcı tarafından tam sayı koordinatlı, iki boyutlu bir düzlemde N nokta verildiğinde tüm noktaları içeren minimum çemberi ve verilen N noktanın en yakınından geçen eğriyi çizdirmektir. Ayrıca çizdirmiş olduğumuz dairenin yarıçapını ve merkezini hesaplamamız amaçlanmıştır. Bizden “.txt” uzantılı dosyadan sayısı belirsiz x ve y koordinatlarını okumamız, bu noktaları bir diziye atmamız, bu noktaları içeren en küçük daireyi çizdirmemiz, bu dairenin yarıçapını bulmamız, dairenin merkezini bulmamız, bunları komut ekranında göstermemiz ve graphics.h kütüphanesinin fonksiyonlarını kullanarak grafik ara yüzü oluşturmamız istenmiştir. Bunun için öncelikle “.txt” uzantılı dosyadaki koordinatlar okunarak bir struct dizisinde saklandı. Her koordinat noktası arası mesafe hesaplanarak bu mesafenin en uzak

olduğu iki nokta tespit edildi. Mesafenin yarısı hesaplanarak yarıçapı bulundu. mesafenin orta noktası hesaplanarak merkezi bulundu. eğer bu iki nokta arasındaki mesafeden daha uzak bir nokta varsa tespit edildi ve çemberin merkezi, yarıçapı buna göre tekrar hesaplandı. Çember bu noktalara göre tekrardan çizildi. Dosyadan okunan koordinatlar arayüz ekranında gösterildi. Bernstein polinomları yöntemi kullanılarak tüm noktaların en yakınından geçen bir eğri çizdirildi.

Yöntem

Struct tipinde içinde double değişkeni ile x ve y koordinatlarını tuttuğumuz nokta değişkeni oluşturduk. “.txt” uzantılı dosyadan koordinat sayısını satır satır okumak için fgetc() fonksiyonunu kullandık. Her bir satır indiğinde koordinat sayısını 1 artırarak kaç tane koordinat olduğunu bulduk. Fscanf fonksiyonu ile struct olarak oluşturduğumuz p dizisine dosyadaki koordinatları atadık sonra dosyadan diziye atadığımız koordinatları ekranda gösterdik. İki nokta arası mesafeyi bulmak için math.h kütüphanesini ve sqrt ve pow fonksiyonlarını kullandık. Önce diğer noktalara göre en uzak noktayı bulmak için bir fonksiyon yazdık. Bu fonksiyon tüm noktalar arasındaki mesafeyi hesaplayarak en uç noktayı bulur ve indeksini döndürür. Bu noktaya göre en uzak bir diğer noktayı bulmak için başka bir fonksiyon yazdık. Bu noktaya en uzak noktayı bulmak için diğer noktalarla arasındaki mesafeyi tek tek hesapladık. Bu noktanın indeksini fonksiyonla döndürdük. Main içinde fopen() fonksiyonu kullanarak dosyayı açtık sonra struct nokta tipinde bir

p dizisi oluşturduk. Bu p dizisi dosyadan okunan koordinat sayısı kadar eleman içerir. İnt tipinde n1 değişkenini en uzak ilk nokta olarak atadık. N2 değişkenini ise en uzak 2. Nokta olarak atadık. Double tipinde yarıçap tanımladık math.h kütüphanesinden yararlanılarak oluşturduğumuz mesafe fonksiyonuna bu iki noktayı atayıp sonucu ikiye bölerek yarıçapı bulduk. Struct nokta tipinde bir merkez tanımladık uç noktalardan ikisinin x noktalarını toplayıp ikiye bölerek merkezin x koordinatını bulduk aynısını y koordinatı içinde yaptık. Eğer merkeze uzak bir 3. Nokta daha varsa ve çember içinde değilse bunu kontrol etmek için double tipinde bir yarıçap kontrol değişkeni tanımladık ve bu değişkene yarıçapı atadık ve bir 3. Nokta tanımladık . Kordinatın sayısına göre bir for döngüsü açtık. Döngü tüm noktaları kontrol eder eger merkeze uzaklığı yarıçaptan büyük başka bir nokta varsa bunu bulur ve nokta3 değişkenine atar. $x^2 + y^2 + 2ax + 2by + c = 0$ denklemini her üç nokta için yazdık bu 4 bilinmeyenli 3 değişkeni yok etme metodu ile çözerek tekrardan çemberin merkezinin x y koordinatlarını ve yarıçapını bulduk.İlk olarak allegro.h kütüphanesini kurmayı denedik fakat kurmakta zorlanınca graphics.h kütüphanesini kurduk. Graphics.h kütüphanesinin fonksiyonu olan initgraph kullanarak bir arayüz açtırdık.line fonksiyonu ile koordinat düzlemini çizdirdik. 2 tane for döngüsü açarak line fonksiyonu ile x ve y koordinatlarını 8 birim arayla ölçeklendirdik. Circle fonksiyonuna merkezin x y yarıçapını

atayarak çemberi çizdirdik. Circle fonksiyonuna yine merkezin koordinatlarını atayarak arayüzde merkezi gösterdik. For döngüsü açarak yine circle fonksiyonu ile struct tipinde oluşturduğumuz p dizisindeki koordinatlara x'i 8 ile çarparak ve ekranın yarısı olan 320 ekleyerek ve y noktasınıda 8 ile çarpıp 240 ile çıkararak arayüzde gösterdik. Daha sonra dosyadan okuttuğumuz noktaları ekranda göstererek bu noktalardan geçen eğriyi çizmek için bernstein polinomlarından yararlandık.

$$\gamma(t) = \sum_{i=0}^n P_i \frac{n!}{i!(n-i)!} (1-t)^{n-i} t^i,$$

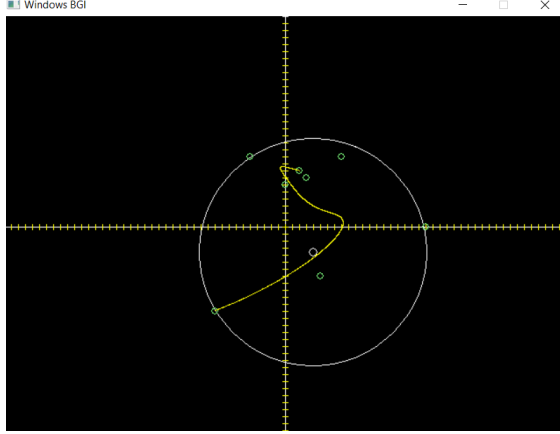
Polinom, belirlediğimiz kontrol noktalarına kadar putpixel fonksiyonu ile eğriyi çizdirir.

Sonuçlar

Tüm bu işlemler sonucunda dosyadan koordinatların okunma işlemi başarıyla gerçekleştirildi. Program dosyadan aldığı noktaları başarıyla okudu ve dinamik bir diziye atadı. Gerekli matematiksel hesaplamalar yapıldı ve çemberin yarıçapı, merkezi ve noktaların koordinatları ekrana yazdırıldı. Ayrıca çember, noktalar ve noktaların yakınından geçen eğri graphics.h kütüphanesi fonksiyonları ile çizdirilip ekranda gösterildi.

Çıktılar

Çember,noktalar,merkez ve eğri



Noktalar ve merkezin koordinatları ve yarıçap

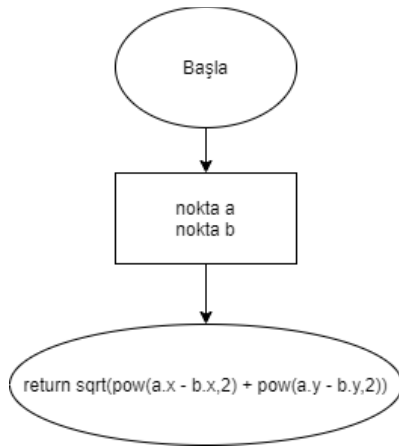
```
"C:\Users\msi\Desktop\c projeleri\deneme\bin\Debug\deneme.exe"
Dosya oluşturuldu
Noktalar: {2,8},{-5,-10},{3,7},{0,6},{5,-7},{8,10},{20,0},{-10,-12},
Merkez: {4,0,-3.6}
Yarıçap: 16.4
```

Dosyadan okunan noktalar

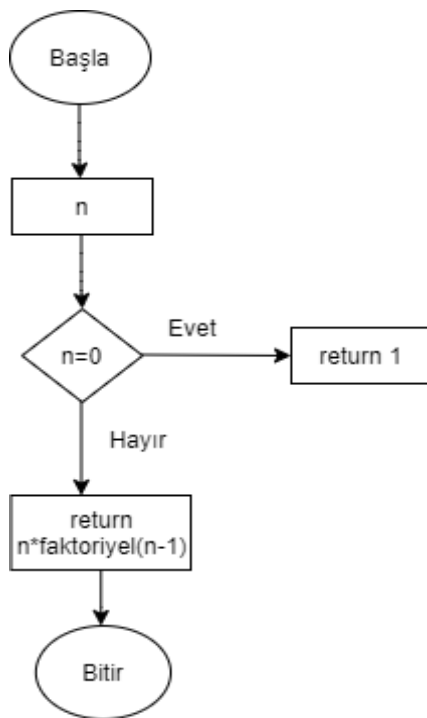
```
noktalar.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
2 8
-5 -10
3 7
0 6
5 -7
8 10
20 0
-10 -12
```

AKIŞ ŞEMALARI:

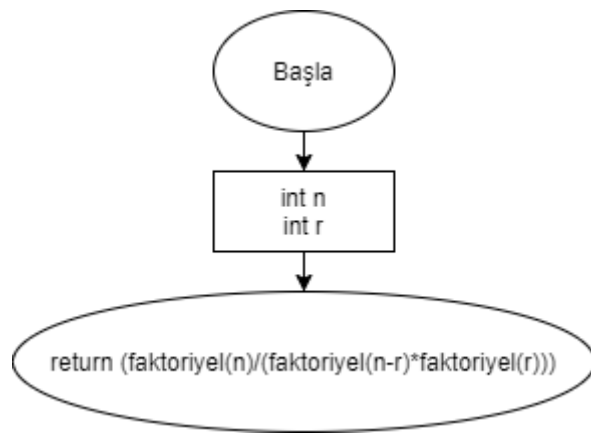
Mesafe()



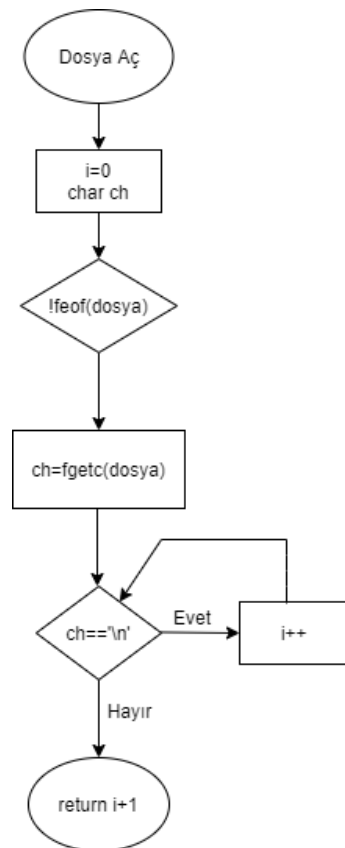
Faktöriyel(int n)



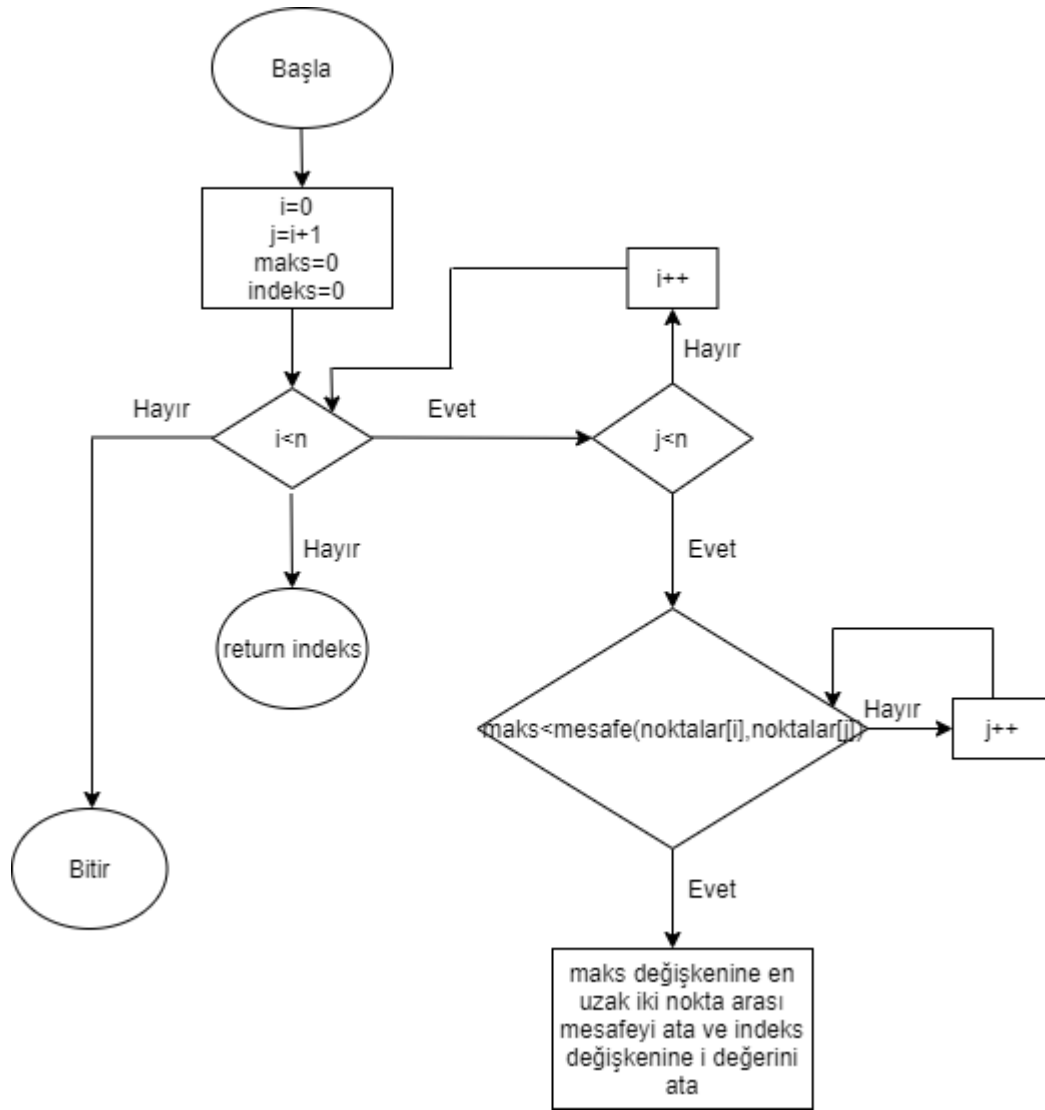
Kombinasyon()



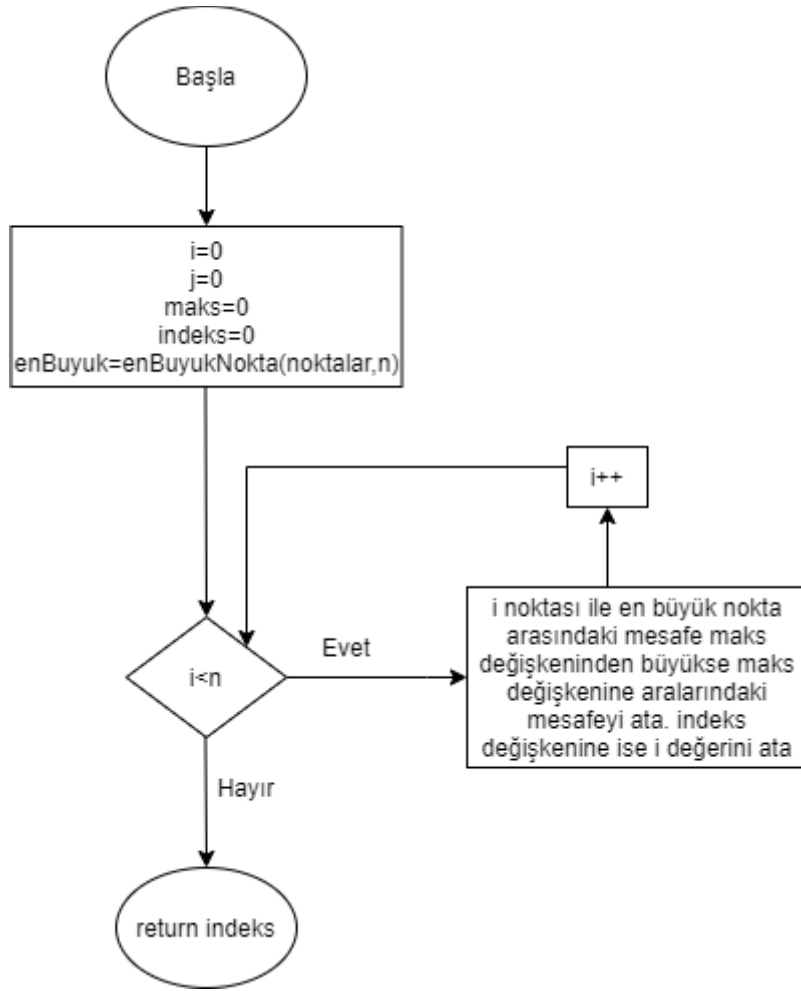
satirOku()



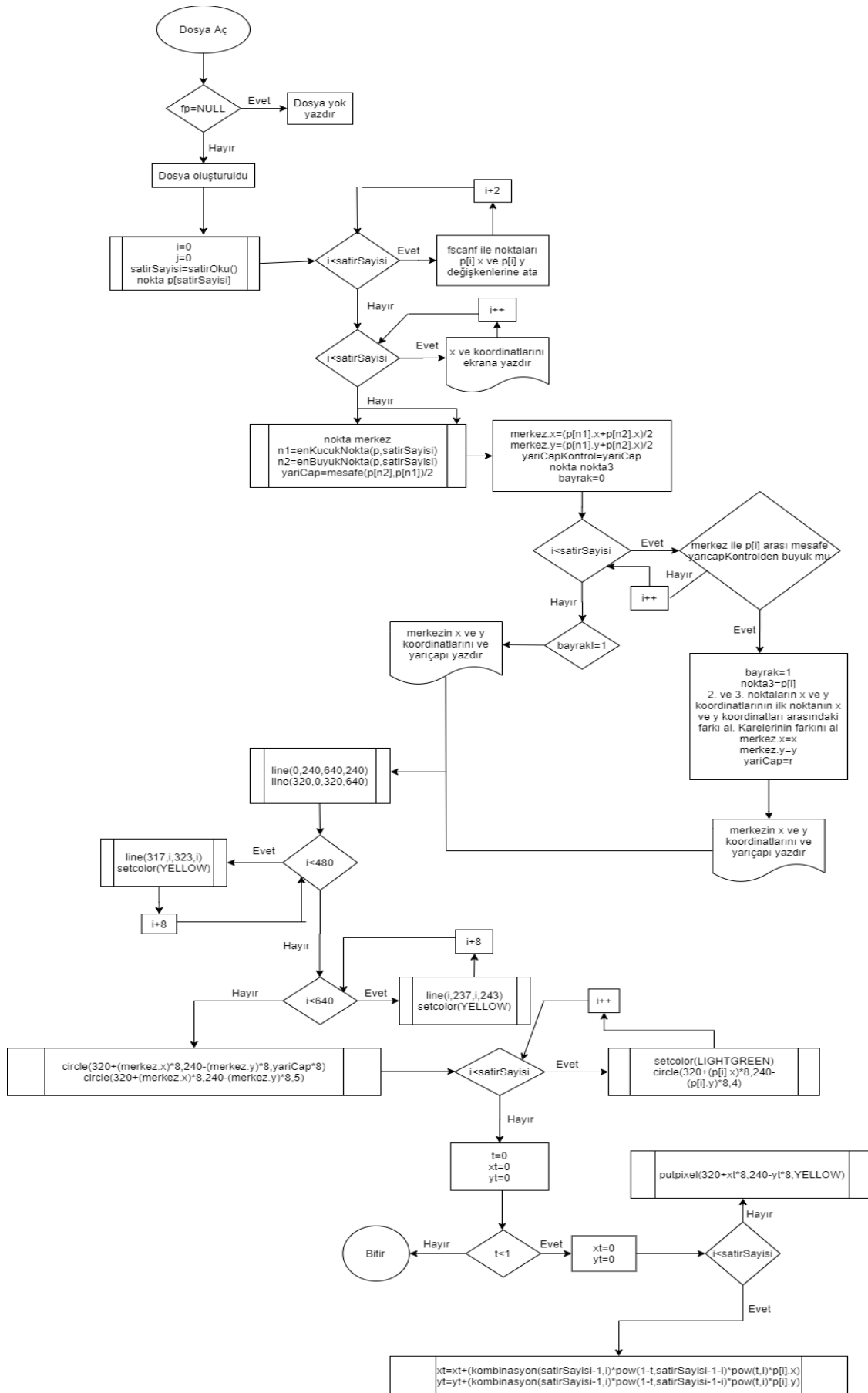
enBuyukNokta()



enKucukNokta(nokta noktalar[], int n)



Main()



Referanslar

- https://tr.wikipedia.org/wiki/B%C3%A9zier_e%C4%9Frisi
- https://en.wikipedia.org/wiki/Bernstein_polynomial
- <https://www.math-only-math.com/circle-passing-through-three-given-points.html>
- <https://www.geeksforgeeks.org/include-graphics-h-codeblocks/>
- <http://winbgim.codecutter.org/>
- <https://www.wikihow.com/Find-the-Distance-Between-Two-Points>
- <https://www.w3resource.com/c-programming-exercises/file-handling/c-file-handling-exercise-5.php>