

**KOCAELİ ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**YAZILIM LABARATUVARI 2**

**2. PROJE**

140201082 SÜMEYRA ÇAKMAK

140201083 MELEK PAZARBAŞI

**Multi Thread Kullanarak Karar Ağacı Oluşturma**

1.Melek Pazarbaşı 2.Sümeyra Çakmak

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

melekp30@gmail.com sumeyra.cak@gmail.com

1.1 PROBLEM TANIMI

Projede multi thread yapısını kullanarak bize verilen veri seti üzerinden karar ağacı oluşturmamız istenmektedir. Karar ağacı oluşturma aşamasında C4.5 algoritmasını kullanmamız istendi.

Verilen "haberman.txt" dosyasında bulunan veri setlerine göre en iyi bölümlemenin bulunması istenmiştir. Veri setinde 3 özellik bulunmaktadır. Bu özelliklerden birincisi 30 ile 83 aralığında değer almaktadır; ikincisi 58 ile 69 aralığında değer almaktadır; üçüncüsü 0 ile 52 aralığında değer almaktadır. Bölümleme işlemi ilk özellik 50-60-70 için, ikincisi 62-63-64 için ve üçüncüsü 5-10-19 için bölümlenecektir.

Elde edilen verilere göre binary tree oluşturmamız istenmiştir. Daha sonra bu iki dal için veri setinde bulunan diğer bir özelliği ağaca eklemek için oluşan iki düğüme ait iki thread çalıştırılacaktır. Veri setinde yer alan tüm özellikler ağaca yerleştirilene kadar bu işlem devam edecektir. Ağacın her seviyesinde yer alacak özelliği seçerken yine entropi hesabı yapılması gerekmektedir.

Aynı zamanda Karar ağacında kök düğümden sonraki tüm seviyelerde oluşacak düğüm sayısı kadar thread oluşturmamız ve her düğüm için bu threadleri eş zamanlı olarak çalıştırmamız istenmektedir.

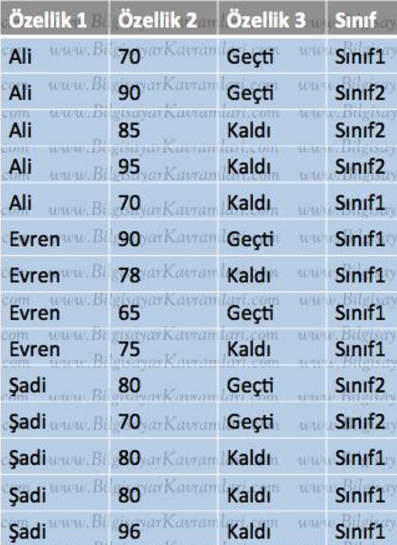
Tüm işlemler bittikten sonra elde ettiğimiz binary treenin grafik ekranında çiziminin yapılması istenmiştir.

**1.2 YAPILAN ARAŞTIRMALAR**

Threadler, threadların kullanımı, entropi hesaplama ve karar ağaçlarıyla ilgili araştırmalar yapıldı.

*Decision Tree(Karar Ağaçları)* :

Karar ağacı yaklaşımı, hedef fonksiyonlarını yaklaşık olarak hesaplamak için kullanılan ve öğrenme fonksiyonunun karar ağacı ile gösterildiği bir yöntemdir. Bir karar ağacı ise ağaç görünümünde tanımlayıcı ve tahmin edici bir modeldir . Bu model, karar alıcıya karar alırken hangi faktörlerin göz önüne alınması ve her bir faktörün kararın farklı çıktıları ile geçmişte nasıl ilişkili olduğunun belirlenmesi konularında yardımcı olur.



Şekil 1: Örnek Veri Kümesi

Yukarıdaki veri kümesi için C4.5 ağacı oluşturmak istiyorsak ilk adım olarak bilgi kazanımı(information gain) hesaplanır. Bilgi kazanımı hesaplanırken, o anda veri kümesinde bulunan bütün veriler ve hesaplanması istenen belirli bir verinin üzerinden gidilir. Bilgi (information) hesaplaması sırasında kullanılacak olan formül aşağıdaki şekildedir.



Şekil 2:Bilgi Hesaplaması Formülü

Yukarıdaki şekilde her örnek için bilgi değeri hesaplandıktan sonra kazanım (gain) hesaplanması mümkündür.



Şekil 3: Her Bir Parça İçin Bilgi Formülü

Yukarıdaki formülde her bir i parçası için yapılan bilgi hesaplaması verilmektedir.

Kazanım ise bu durumda aşağıdaki şekilde hesaplanabilir.

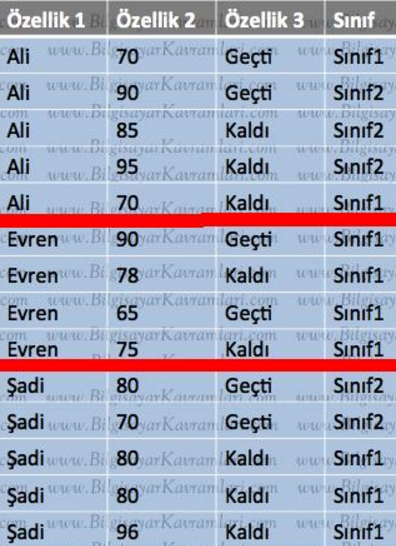
4

Şekil 4: Kazanım Hesabı

Yukarıda verilen veri kümesi için bilgi kazanımı hesaplanırsa 14 toplam satırdan 5 tanesi sınıf 2 ve 9 tanesinin sınıf1 olduğunu dikkate alarak aşağıdaki eşitlik yazılır.Önce bilgi değerleri hesaplanır sonra da bilgi kazanımı bulunur.



İlk bilgi değeri bütün parçanın hesaplandığı yani 14 satırın tamamının dikkate alındığı ve 9/14 ve 5/14 olarak iki ihtimalin hesaba katıldığı durumdur. Bu durum aynı zamanda entropi de denilebilir. İkinci bilgi hesabımızda özellik 1 kullanılacak. Buna göre veri kümemizin ilk 5 satırında Ali, sonraki 4 satırında Evren ve son 5 satırında Şadi özellikleri var. Buna göre tabloyu 3 parçaya bölersek ;



*Şekil 5* : Örnek Veri Kümesi

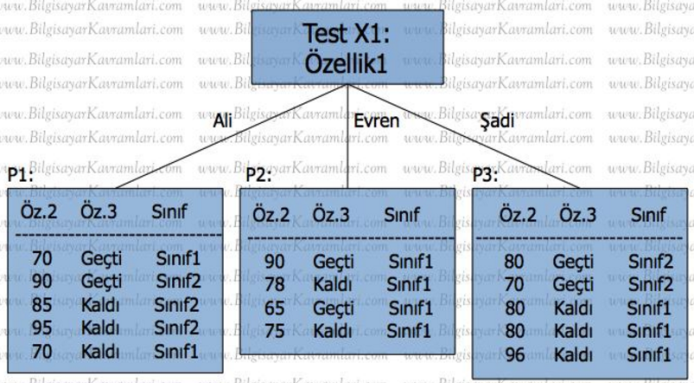
Yukarıdaki yeni tabloya göre her özellik parçasının ayrı ayrı hesaplanarak denklemde yerine yazılması gerekir:



Yukarıdaki formülde mavi renkle belirtilen durum 1. özellik için (x1) 5, 4 ve 5 parçadan oluşan ve her parça için ayrı ayrı Sınıf1 ve Sınıf2 değerlerinin sayıldığı durumdur. Son adımda bu iki değer arasındaki farkı hesaplayabiliriz:

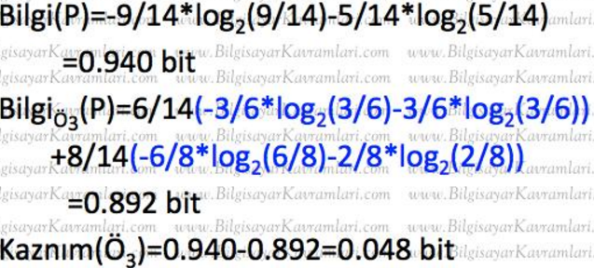
8

Yukarıda bulunan bilgi kazanımı, bütün veri kümesindeki Özellik1 için bütün sınıflar arasındaki kazanımı göstermektedir. Bu değerleri kullanarak aslında veri kümemizi 3 parçaya bölmüş ve her birisi için bilgi kazanımının olası değerini hesaplamış oluyoruz:



Şekil 6: Karar Ağacı

Yukarıdaki gösterim karar ağacının alabileceği olasılıklardan birisidir ve ağacın Özellik 1’de bulunan isimlere göre karar noktası oluşturulması halinde alacağı vaziyeti gösterir.Yukarıdaki hesapların aynısı Özellik 3 için yapılırsa

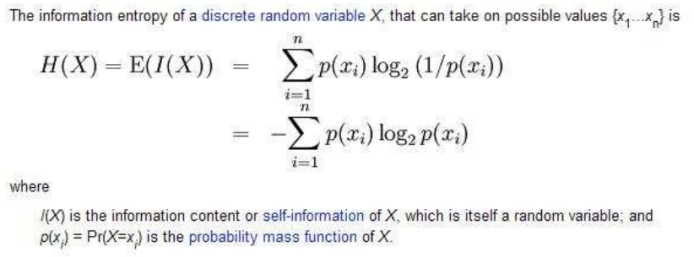


Bu durumda C4.5 ağacı en yüksek kazanıma sahip olan değeri alacaktır. Bu değer Özellik 1 için isimler olduğundan karar ağacının bu adımda Özellik 1’e göre karar noktası eklemesi doğru olacaktır.

*Karar Ağaçları: En ayırt edici nitelik nasıl bulunur?*

Bilgi Kazancı (Information Gain):ID3, C4.5 gibi karar ağacı metotlarında en ayırt edici niteliği belirlemek için her nitelik için bilgi kazancı ölçülür. Bilgi Kazancı ölçümünde Entropy kullanılır.

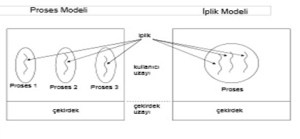
Entropy rastgeleliği, belirsizliği ve beklenmeyen durumun ortaya çıkma olasılığını gösterir.



*Şekil 7* : Entropi hesabı

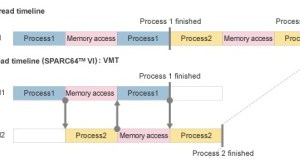
*THREAD*

Thread (iplik) kavramı, aynı [process](http://bilgisayarkavramlari.sadievrenseker.com/2007/11/18/islem-process/" \o "process" \t "_blank) ortamında birden fazla iş yürütme imkanı sağlar. Bir process’in çalışmaya başlaması ile birlikte bir thread ([main thread](http://bilisim-kulubu.com/sozluk/sozluk.php?e=Main+Thread" \o "main thread" \t "_blank)) oluşturulur ve bu process içerisinde birden fazla iplik ([multi-thread](http://bilisim-kulubu.com/sozluk/sozluk.php?e=Main+Thread" \o "multi thread" \t "_blank) ) oluşturulabilir. Yaratılan iplikler aynı adres uzayında koşarlar. Her ipliğin kendine has program sayacı, saklayıcıları, yığını ve durumu(yok edilme, askıya alınma, önceliğinin değiştirilmesi) vardır. Bir iplikteki (thread) değişkeni aynı işlem tarafından üretilmiş bütün iplikler değiştirebilir veya okuyabilir.

[](https://i2.wp.com/www.kadinyazilimci.com/wp-content/uploads/2014/09/a.jpg)

*Şekil 8*: Process ve iplik modelinin görünümü

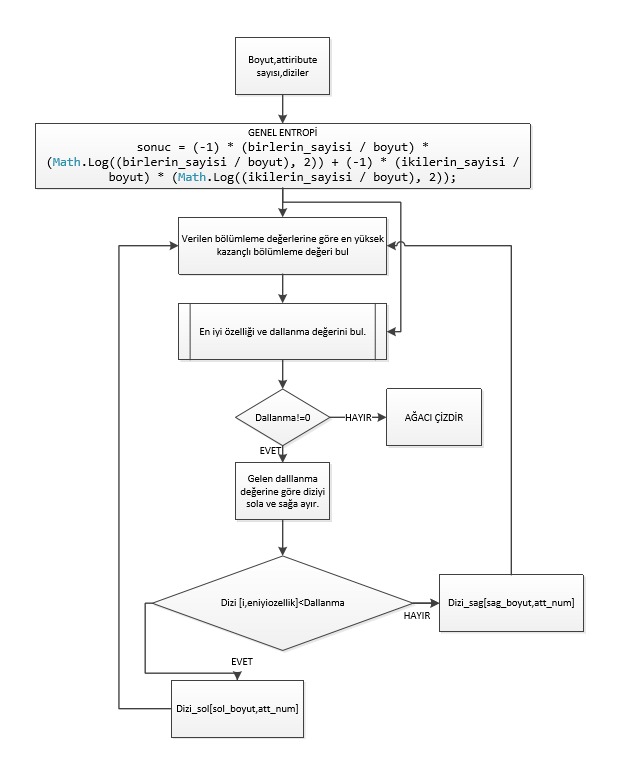
Threadler iş parçacıkları olup, çoklu görevlerde [senkronizasyonu](http://tr.wikipedia.org/wiki/Senkronizasyon) sağlamak için kullanılır. Her threadin farklı bir görevi vardır. Eğer bu threadler aynı anda aynı bölgeye girerse hatalar ([deadlock](http://www.csharpnedir.com/articles/read/?id=742" \o "deadlock" \t "_blank)) meydana gelebilir.

[](https://i2.wp.com/www.kadinyazilimci.com/wp-content/uploads/2014/09/thread1.jpg)

*Şekil 9:* Threadlerin senkronizasyonla çalışma şekli

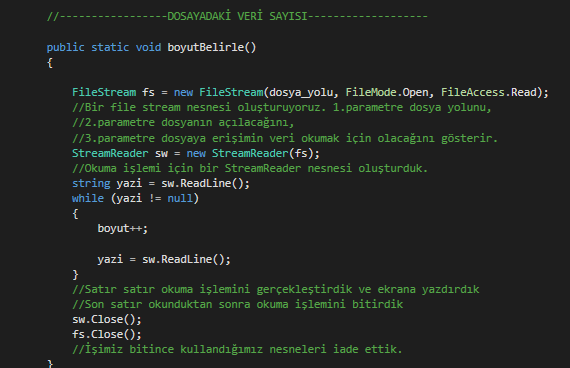
**1.3 TASARIM**

**1.3.1 AKIŞ ŞEMASI:**

****

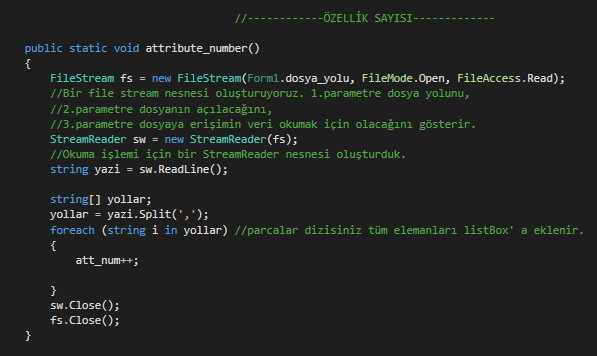
*Şekil 10* : Akış şeması

**1.3.2 YAZILIM MİMARİSİ:**



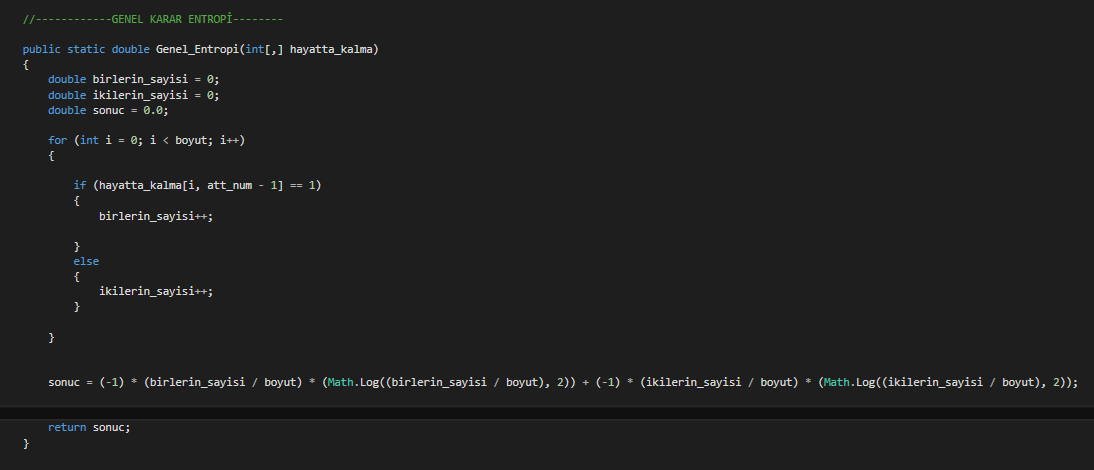
*Şekil 11* : boyutBelirle() fonksiyonu

Oluşturulan dizlerin dinamik olması açısından ilk olarak elimizde bulunan veri setinin boyutunu analiz ettik.Bu boyutu kullanarak tanımlayacağımız dizilerde rastgele olarak olar verilen her veri setinde çalışabilirliğini sağlayacak.



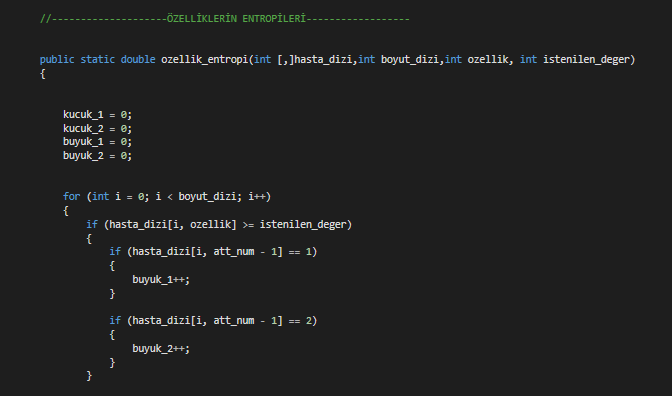
*Şekil 12* : attribute\_number() fonksiyonu

Veri setindeki sütun sayılarına bağlı olarak oluşturulan bu fonksiyonda bize hangi veri seti verilirse verilsin dinamik olarak çalıştırılacaktır.Dosyadan satır satır okuma ile yapılmış bu fonksiyonda satırlar yazi string veri tipli değişkende tutuluyor olup “,” imlecine göre parçalanır ve yollar dizisine atanır, foreach döngüsü yardımıyla yollardaki her bir indis i dizisine giderken att\_num arttırılır.Burada att\_num sütuna bağlı olarak artar.



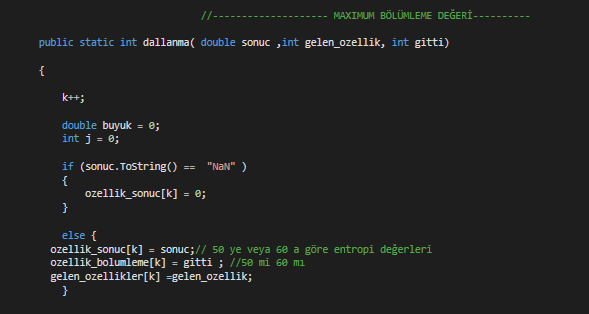
*Şekil 13* : Genel\_Entropi() fonksiyonu

Genel karar entropisini hesaplayan bu fonksiyon ise hastaların tüm özelliklerini içerisinde bulunduran bir 2 boyutlu matris tarafından ve bu matrisin sütun indisi 3 olan tüm değerlerine bağlı olarak bir matematiksel hesaplama sonucunda bulunuyor.3.sütundaki 1 ve 2 değerleri ayrı hesaplanıp bir olasılık temelli matematiksel hesaplamanın üzerinde birleştiriliyor.



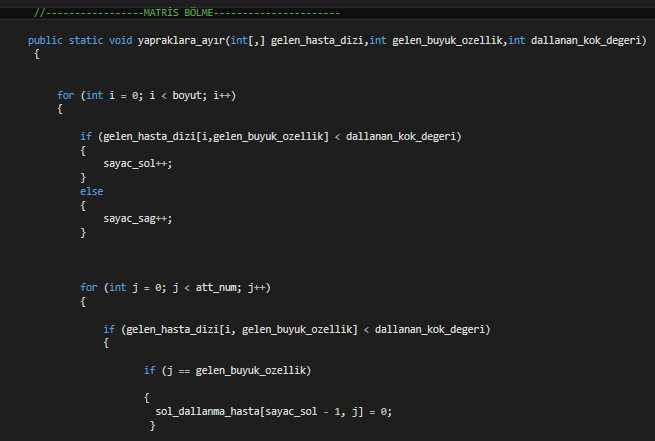
*Şekil 14* : ozellik\_entropi() fonksiyonu

Özelliklerin entropisini bulduğumuz bu fonksiyonda ise elimize verilen bölümlere değerlerinin manuel olarak girilmesi sonucunda teker teker her bölümlemeye ait bir gain bulunması ve bu gainlerin içerisinden en yüksek olanı seçip o kazancı kök yaparak işlemi tamamlar.



*Şekil 15* : dallanma() fonksiyonu

Seçilen özelliğin hangi bölümleme değerinden dallanacağı bilgisini veren bu fonksiyonun parametreleri ozellik\_entropi() fonksiyonunun içerisinde tanımlanmış olup burada özellik entropinin içerisinden gelecek entropi değerine istenilen özelliğe ve istenilen deger parametrelerine göre çalışacaktır.Bu işlem sonucunda fonksiyondan bölümleme değeri gelecektir(özellik 1 için 50 değeri gibi).



*Şekil 16* : yapralara\_ayir() fonksiyonu

Matris bölme işlemi yapan bu fonksiyon ise 1.Kökün bulunmasına ve dallanma değerinin bulunmasına bağlı olarak bu dallanma değerinden küçük olanları sol\_dallanma\_hasta dizi sinde, bu dallanma değerinden büyük veya bu dallanma değerine eşit olan değerlerin hepsini sağ\_dallanma\_hasta dizisinde tutar .Önceki en yüksek gelen değerin sütununa görü tüm satırları 0 yapar.

**1.4 GENEL YAPI:**

Projede multi thread yapısını kullanarak bize verilen veri seti üzerinden karar ağacı oluşturmamız ve karar ağacı oluşturma aşamasında C4.5 algoritmasını kullanmamız istenmişti.

Projeyi gerçekleyebilmek için önce yapmış olduğumuz arayüzden haberman.txt dosyasını seçiyoruz. Seçtiğimiz dosyadan gelen verileri hasta[ , ] adında matriste tutuluyor. Veriler alındıktan sonra genel entropi bulunup ekrana yazdırılıyor.

Hasta[ , ] matrisinin sütunları bazı özellikleri tutuyor. Hastanın yaşı, çalışma yılı, hayatta kalma durumu vb. Bu özelliklerin ayrı olarak entropisi hesaplanıyor. Her özellik için 3 ayrı değerle entropi hesabı yapılıyor. 1. özellik için 50, 60 ve 70'le, 2. özellik için 62, 63, 64'le, 3. özellik için 5, 10, 19'la hesaplanıyor ve her özellik için en iyi entropi değeri bulunuyor. Bu işlemler sonrasında her özellik için ayrı ayrı bulunan en iyi entropi de kendi aralarında karşılaştırılıyor ve en iyi entropi - özellik bulunuyor.

Veri setine ve bölümleme değerlerine bağlı olarak genel entropi bulunur. Bu genel entropi veri setini tuttuğumuz hasta[,] matrisinin 3. sütunudur. Bu sütundaki değerleri 1 ve 2 değerlerine göre ayırarak olasılık hesaplaması yapan matematiksel işlem sonucu bir genel entropi bulduk.

Verilen bölümleme değerlerine göre ise en yüksek kazançlı entropiyi bularak 1.kökün hangi özelliği ve dallanma olarak nitelendirdiğimiz o özelliğe ait en iyi bölümleme durumu bulunmuş oldu.Daha sonra bu matrisin tüm özellikleri bölümleme durumundan küçük olanları sol\_dallanma\_hasta olarak nitelendirdiğimiz matrise atanır, bölümleme durumundan büyük veya eşit olan değerlere karşılık gelen satır değerleri yani tüm özellikler sag\_dallanma\_hasta olan matrise atanır.Bu büyük matrisin düğümleri olan bumatrislerede aynı işlemler kullanılarak matris küçültülür ve 3.satırda 1 ve 2 değerleri ayrı olana kadar bu işleme devam edilir.

Ağacı çizdirme işlemleri için önce yeni bir formda panel açılıp grafik işlemleri bunun üzerinde yapılıyor. Önce çizilecek ağacın kök ve düğümleri çember olarak ekrana çiziliyor. Sonra da bu düğümler arasında bağ sağlayacak linelar çiziliyor. Ardından agac[ ] dizisinden gelen değerlere göre binary tree oluşturmak için TreeNode.cs ve Tree.cs adında iki class oluşturuyoruz. agac[ ] içerisindeki verileri Tree'ye insert ediyoruz. Bu işlemlerden sonra verileri ağaca rahatça yazabilmek preorder biçiminde sıralıyoruz, sıraladığımız verileri preOrder[ ] dizisine atıyoruz. preOrder[ ] dizisinden gelen değerlere göre de çizmiş olduğumuz ağaca verileri yazdırıyoruz.

# 1.5 REFERANSLAR

1. http://muratbilginerncfkr.blogspot.com.tr/2013/11/29-c-gorsel-programlama-29-grafik.html
2. https://www.youtube.com/watch?v=Jc5eXYwTROg
3. http://www.bilisimogretmeni.com/visual-studio-c/visual-studio-c-dersleri-c-ile-grafik-islemleri-2-kare-dikdortgen-daire-elips-cizme.html
4. https://cmengcompany.wordpress.com/2011/06/21/c-thread-kullanimi/
5. http://stackoverflow.com/questions/831009/thread-with-multiple-parameters
6. http://www.emreyldrm.com/p/csharp-dersleri-43-multithreading-ve-thread-classi-kullanimi
7. http://mail.baskent.edu.tr/~20410964/DM\_8.pdf
8. https://www.slideshare.net/oguzzhanoguzz/karar-aalar-ve-entropi-kds