

# GRAFİK RENKLENDİRME ALGORİTMASI KULLANILARAK OTOMATİK DERS PROGRAMI GELİŞTİRME YAZILIMI

**Hasan H. BALIK**

Fırat Üniversitesi  
Mühendislik Fakültesi  
Elektrik-Elektronik Müh.  
Bölümü  
[hasanbalik@gmail.com](mailto:hasanbalik@gmail.com)

**Ayhan AKBAL**

Fırat Üniversitesi  
Mühendislik Fakültesi  
Elektrik-Elektronik Müh.  
Bölümü  
[ayhanakbal@gmail.com](mailto:ayhanakbal@gmail.com)

**İ. Oktay BARUT**

Fırat Üniversitesi  
Mühendislik Fakültesi  
Elektrik-Elektronik Müh.  
Bölümü Öğrencisi

**Levent SOLAKOĞLU**

Fırat Üniversitesi  
Mühendislik Fakültesi  
Elektrik-Elektronik Müh.  
Bölümü Öğrencisi

## ÖZET

Dönem başlarındaki ders programlarının hazırlanması ve dönem sonlarındaki sınav programlarının yapılması oldukça zor ve karmaşıktır. Öğrencileri ve öğretim elemanlarını mağdur etmeyecek şekilde ders ve sınav programı hazırlama her üniversitede problem olmaktadır. Genellikle elle hazırlanmış programlarda bütün gayretlere rağmen sorunlar tam olarak giderilememiştir. Bu proje böyle bir sorunu ortadan kaldırmak için hazırlanmıştır. Projede Graph Coloring Algoritması kullanılmıştır.

## ABSTRACT

*To prepare time table at the beginning of each term and to organize exam table at the end of each term are very difficult and complex process. Each person desires to prepare time tables which satisfy both students and lecturers. Although officers' good willing, it is very hard to satisfy both sides with hand made time tables. The main purpose of this project is to develop time table preparation software to overcome these difficulties. In this project Graph Coloring Algorithm were used.*

**Anahtar Kelimeler:** Otomatik Ders Programı, Grafik Renklendirme Algoritması

## 1. GİRİŞ

Bilgisayar ve bilgisayar uygulamaları günlük hayatın her alanında önemli unsurlar olmaya başlayınca, daha önceden bir veya birden çok kişinin uzun sürede yaptığı oldukça karmaşık işlemler sonucunda gerçekleştirdikleri bir çok iş, bilgisayar ortamında çok hızlı çözülmeye başlamıştır. Bu işlemler için genellikle bir paket program hazırlanarak kullanıcılara sunulmuş, kullanıcılar bu programları kullanarak hem çok hızlı hem de hatasız olarak işlemlerini gerçekleştirmişlerdir.

Okuldaki ders programının ayarlanması çok önemli bir konudur. Tahmin edileceği gibi otomatik ders programında bazı zorluklar, çakışmalar, kısıtlamalar, kriterler vardır ve bunlar dikkate alınarak yapılacak programlar oldukça karmaşıktır.

Bu tür problemlerinin çözümünde kullanılan birçok yaklaşım bulunmaktadır. Bunlardan bazıları; Kümeleme, Genetik Algoritmalar, Ard Arda Sıralama, Mantıksal Programlama, Yerel Arama ve Graph Coloring'dir. Gerçekleştirilen bu projede bu yaklaşımlardan biri olan Graph Coloring Algoritması kullanılmıştır.

Üniversitelerdeki sınav programlarının hazırlanması da karmaşık ve hata oranı yüksek konuların başında gelmektedir. Bir çok üniversitede hala ciddi problemler çıkmakta ve her dönem sonunda bu sıkıntılar tekrarlanmaktadır. Bunun sonucu olarak sınavları öğrencinin aldığı derslere göre, çakıştırmadan dağıtacak bir programa ihtiyaç duyulmaktadır. Bu uygulama için Graph Coloring algoritması en kullanışlı algoritmalarından biridir.

Bu çalışmada, Graph Coloring algoritmalarından biri olan Welch ve Powel Algoritması kullanılmıştır. Bu algoritma genel olarak düğümlerin derecelerine dayanmaktadır. Bu yüzden dağıtım yapılacak derslerin bir grafa aktarılması ve bu dersler arasındaki komşulukların belirlenmesi gerekmektedir.

Bu çalışmada dersler düğüm olarak kabul edilmiş, tüm dersleri ve bunların arasındaki komşulukları gösteren graf oluşturulmuştur. Düğümler arasındaki komşuluklar, bir öğrenci tarafından alınan derslere bakılarak kurulmuştur. Aynı öğrencinin aldığı dersler komşu ilan edilmiştir. Düğüm derecelerine bakılarak Welch ve Powel algoritması tüm düğümler renklendirilinceye kadar uygulanmıştır. Renklendirme yapılırken kullanılan renk sayısı, ders yerleştirme yapılırken ihtiyaç duyulan saat sayısını vermektedir.

Projede C++ Builder 6 kullanılmış olup Paradox veritabanı kullanılarak hazırlanmıştır.

## 2. GRAPH COLORING ALGORİTMASI

Graf renklendirme, graf üzerinde birbirine komşu olan düğümlere farklı renk atama işlemidir. Amaç, en az sayıda renk kullanılarak tüm düğümlere komşularından farklı birer renk vermektir.

Renklendirmede kullanılan renk sayısı kromatik sayı olarak adlandırılır.

Uygulamada, graf renklendirmenin kullanılacağı alanların başında, ilk akla gelen, harita üzerindeki bölgelerin renklendirilmesi olmasına karşın, graf renklendirme bilgisayar biliminde ve günlük yaşamdaki birçok problemin çözümünde kullanılan bir yaklaşımdır. Örneğin, sınırlı sayıda işlemcisi olan bir sistemde proseslerin işlemcileri kullanma zamanları ve sıralamasının belirlenmesinde, üniversitenin bir fakültesinde sınav saat ve günlerinin çakışmayacak bir şekilde yerleştirilmesinde çözüm sunar, bir de tabii ki bölge renklendirmede en az sayıda renk kullanılmasını sağlar.

Graf renklendirmede kullanılan algoritmalarından birisi Welch ve Powel'in önerdiği yöntemdir. Bu yöntem genel olarak düğümlerin derecelerine dayanmaktadır.

**Welch ve Powel Algoritması:** Bu algoritmanın davranış adımları aşağıdaki gibidir.

- Düğümler derecelerine göre büyükten küçüğe doğru sıralanır
- İlk renk birinci sıradaki düğüme atanır ve daha sonra aynı renk birbirlerine bitişik olmayacak biçimde diğer düğümlere verilir.

- Bir sonraki renge geçilir, bu renk sıradaki derecesi en yüksek olan düğüme atanır; ve sonra bu renk, daha önce renklendirilmemiş düğümlere birbirlerine bitişik olmayacak şekilde atanır. Üçüncü adım tüm düğümlere renk verilince sonlandırılır [1].

### 3. PROGRAMIN GELİŞTİRİLMESİ

Programın geliştirilmesinde çeşitli verilerin tespiti gereklidir. Bunlar :

- Gün sayısı ve saat aralıkları: Hangi günler de ve hangi saat aralıklarına derslerin yerleştirileceği bilgisidir.
- Dersler: Veri tabanına hangi derslerin girileceği, bu derslerin kodu, adı, seviyesi, dersi alan öğrenci sayısıdır.
- Öğrenciler: Öğrencilerin numarası, adı, soyadı, sınıfı ve öğrencinin aldığı derslerdir.

Bu veriler doğrultusunda veritabanı tabloları oluşturulmuştur. Gerekli olan veritabanı tabloları:

- Öğrenci
- Öğrenciders
- Ders

dir. Ders dağılımı yapılırken çeşitli kriterler vardır. Bunlar;

- Dersler homojen dağıtılmalıdır.
- Öğrencilerin derslerinin çakışmamasına özen gösterilmelidir.

## 4. BÖLÜM 4

Projeyi hazırlarken öncelikle veritabanı kullanmadan program içerisinde girilen sabit değerlerle graf renklendirme algoritmasının doğru çalışıp çalışmadığı test edildi. Bunun için program içerisinde aşağıdaki gibi beş öğrenci (Şekil 1) ve beş ders (Şekil 2) örnek değer olarak girilmiştir.

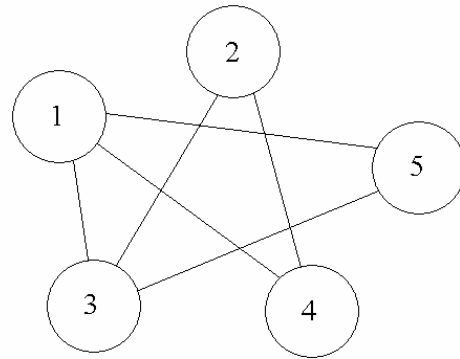
ogrnci	Ogr_no	Ogr_ad	Ogr_soyad	Ogr_sınıf
1	1	Oktay	Barut	1,00
2	2	Melikbah	Özdemir	2,00
3	3	Fadly	Tatar	3,00
4	4	Mehmet Levent	Aslan	4,00
5	5	Harun	Kanki	1,00

Şekil 1: Öğrenciler

ders	Ders_kodu	Ders_adi	Hocasi	Sınıf
1	1	Sayısal Haberleşme	Hasan H. BALIK	1,00
2	2	Elektronik Tüm Devreler	Fikret ATA	2,00
3	3	Mikroişlemciler	Melih C. YNCE	3,00
4	4	Otomatik Kontrol	Mustafa TÜRK	4,00
5	5	Yç Tesisat Projesi	Muhsin T. GENCOBLU	2,00

Şekil 2: Dersler

Veriler ile kullanılan algoritma doğrultusunda komşuluklar çıkarılacak ve bu komşuluklara ilişkin graf elde edilir. Bu durumda graf aşağıdaki gibi olur. (Şekil 3). Burada düğümler Ders Kodlarıdır.



Şekil-3- Komşuluklar

Yukarıdaki grafda komşuluklar açık bir şekilde gözükmektedir. Graf Coloring Algoritması gereği komşu olan hiçbir ders aynı saate konulamaz ve renklendirme bu koşul ile yapılabilir.

## 5. VERİ TABANI VE TABLOLAR

Bu bölümde veritabanı tablolarını ve bu tablolardaki alanlar tanıtılacaktır.

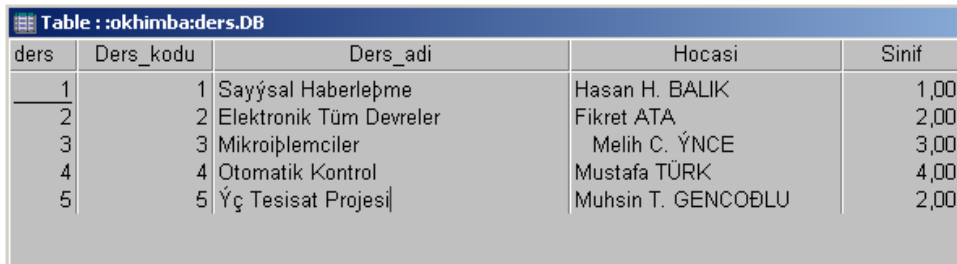
**Oğrenci tablosu :**



ogrenci	Ogr_no	Ogr_ad	Ogr_soyad	Ogr_sinif
1	1	Oktay	Barut	1,00
2	2	Melikpah	Özdemir	2,00
3	3	Fadly	Tatar	3,00
4	4	Mehmet Levent	Aslan	4,00
5	5	Harun	Kanki	1,00

Şekil 4: Öğrenci tablosu

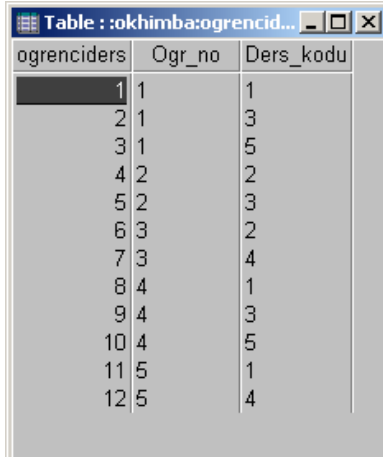
**Ders tablosu :**



ders	Ders_kodu	Ders_adi	Hocasi	Sinif
1	1	Sayısal Haberleşme	Hasan H. BALIK	1,00
2	2	Elektronik Tüm Devreler	Fikret ATA	2,00
3	3	Mikroişlemciler	Melih C. YNCE	3,00
4	4	Otomatik Kontrol	Mustafa TÜRK	4,00
5	5	Yç Tesisat Projesi	Muhsin T. GENCOBLU	2,00

Şekil 5: Ders tablosu

**Oğrenciders tablosu :**



ogrenciders	Ogr_no	Ders_kodu
1	1	1
2	1	3
3	1	5
4	2	2
5	2	3
6	3	2
7	3	4
8	4	1
9	4	3
10	4	5
11	5	1
12	5	4

Şekil 6: Hangi öğrencinin hangi dersleri aldığını gösteren ilişkisel tablo

butonuna basılarak geçilir.

## 6. OTOMATİK DERS YERLEŞTİRME YAZILIMI

Otomatik ders yerleştirme yazılımının kullanılması ve yazılan kodlar bu kısımda anlatılacaktır.

Üğrenci ve ders girme sayfasına git butonuna basılınca gelen formun şekli aşağıdaki gibidir (Şekil 9).

Şekil 7: Ders kaydı sayfası

Yazılım kurulduktan sonra açılışta DBGrid1 komponentinde öğrenciler ve sınıfları gözükmemektedir. Ders kaydı da açılış ekranında mevcuttur. Bu sayede geliştirilen programa has veritabanı oluşturma imkanı da sağlanmış olur.

suiDBLookupListBox1 komponentinde seçili öğrencinin ders kaydının yapılması için alabileceği dersler mevcuttur. Öğrenci seçildikten sonra aldığı dersler ya ders üzerine çift tıklanarak yada “ders ekle” butonuna basılarak dersler suiListBox1 içine atılır. Eğer seçili derslerden biri tekrar seçilirse Şekil 8’de verilen hata mesajıya karşılaşılr.



Şekil-8- Seçili derslerden biri tekrar seçilirse hata mesajı

“Ders Kaydını Yap” butonuna basılarak ders kaydı onaylanır. Form2 öğrenci ve ders giriş sayfasıdır. Form1 üzerindeki

Üğrenci ve ders girme sayfasına git

Şekil-9- Öğrenci ve der giriş sayfası

Bu formda Öğrencilerin Adı, Soyadı ve Sınıfı girilebilir. Editlere bilgiler girildikten sonra Öğrenci Gir butonuna basılarak öğrenci kaydı onaylanır. Ders ekleme kısmı yine bu formda yapılır.



```

AnsiString temp_ders,ders_kodu[100];

sifirla(komsuluk);
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("select ders_kodu
from ders");
Query1->Open();
for(i=0;i<=Query1->RecordCount;i++)
{
    komsuluk[i][0]=i+1;
}
for(i=0;i<Query1->RecordCount;i++)
{
    temp_ders=Query1-
>FieldByName("ders_kodu")->AsString;
    //ders_ad[i]=Query1-
>FieldByName("ders_adi")->AsString;
    ders_kodu[i]=temp_ders;
    Query4->Close();
    Query4->SQL->Clear();
    //
    Query4->SQL->Add("select distinct
ders_kodu from ogrenciders where ogr_no in(select
distinct ogr_no from ogrenciders where
ders_kodu=\""+temp_ders+"\" ) and
ders_kodu!=\""+temp_ders+"\" ");
    Query4->Open();
    count[i][0]=i+1;
    count[i][1]=Query4->RecordCount+1;
    for(k=0;k<=Query4-
>RecordCount+1;k++)
    {
        komsuluk[i][k+1]=Query4-
>FieldByName("ders_kodu")->AsString;
        Query4->Next();
    }
    Query1->Next();
}
kayitsay=Query1->RecordCount;
for(i=0;i<Query1->RecordCount;i++)
{
    suiListBox2->Items-
>Add(komsuluk[i][0]);

    for(k=1;k<=count[i][1]-1;k++)
    {

        suiListBox2->Items-
>Strings[i]=suiListBox2->Items-
>Strings[i]+' '+komsuluk[i][k];
    }
}
Sirala(count,99);
for(i=0;i<10;i++)
{

```

```

ListBox1->Items->Add(count[i][0]);
ListBox1->Items->Strings[i]=ListBox1-
>Items->Strings[i]+' '+count[i][1];
}
suiButton3->Enabled=false;
}

```

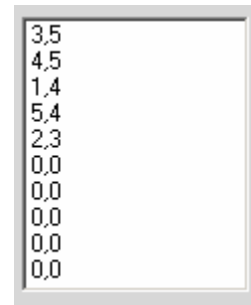
ve komşuluk çıkarma işlemi başarıyla tamamlanır. Komşuluk çıkarma işleminden sonra suiListBox2'nin ve butonun şekli aşağıdaki gibidir (Şekil 11).



Şekil 11:Komşuluk çıkarma



butonuna basıldığında icra edilen ikinci kısım ise kodda görüldüğü gibi hangi dersin kaç komşusu olduğu kısmıdır. Bu kısım algoritmada önemli yer tutar çünkü en fazla komşuluğa sahip dersten başlanarak yerleştirme yapılacaktır. ListBox1 de bu dizi görüntülenir ve aşağıdaki gibi olur (Şekil 12).



Şekil12: Hangi dersin kaç komşusu var

Formdaki bir diğer görülebilecek şey hangi dersten kaç tane bulunduğu bilgisidir. Bunu görmek için



butonuna tıklanarak sonuç görülebilir. Bu butonun altında icra edilen satırlar şunlardır.

```

void __fastcall
TForm3::suiButton2Click(TObject *Sender)
{

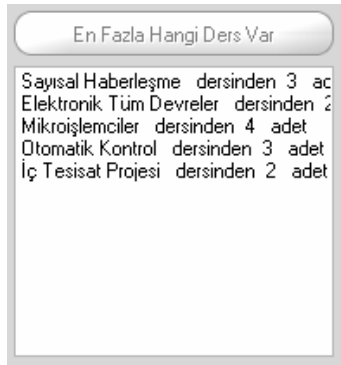
```

```


int i;
String str;
for(i=1;i<=Query2->RecordCount;i++)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("select * from
ogrenciders where ders_kodu="+IntToStr(i)+"");
Query1->Open();
Query3->Close();
Query3->SQL->Clear();
Query3->SQL->Add("select * from ders
where ders_kodu="+IntToStr(i)+"");
Query3->Open();
str=Query3-
>FieldByName("Ders_adi")->AsString+" dersinden
"+IntToStr(Query1->RecordCount)+" adet";
suiListBox1->Items->Add(str);
}
suiButton2->Enabled=false;
}


```

İcra edilen bu satırlar sonrasında Form3 üzerinde bulunan suiListBox1'i şu şekilde görebiliriz(Şekil-13).



Şekil-13- En fazla hangi ders var

Projedeki en önemli kısım  butonu altında icra edilen komutlardır. Bu buton altında daha önce bulunan komşuluklardan ve en fazla komşuluğa kim sahip gibi bilgiler yardımıyla bunların doğrultusunda en iyi şekilde ders programı çıkarılması amaçlanmıştır. Ve hiçbir öğrenciyi mağdur etmeden bunu başarabilmektedir.

 butonuna basıldığında şu satırlar icra edilir:

```

int komsumu(int deger,int indis)
{
int i,j,ara;
for(i=1;renkler[indis][i]!='0';i++)
{

```

```

ara=StrToInt(renkler[indis][i]);
for(j=0;komsuluk[ara][j]!=0;j++)
{
if(komsuluk[ara-1][j]==deger)
return 0;
}
}
return 1;
}
void yerlestir(int deger,int indis)
{
int i;
for(i=1;renkler[indis][i]!=0;i++)
{}
renkler[indis][i]=deger;
}
}
void __fastcall
 TForm3::suiButton7Click(TObject *Sender)
{
bool tamam;
int i,k,deger;
for(i=0;count[i][0]!=0;i++)
{
deger=count[i][0];
k=0;
tamam=true;
while(tamam)
{
if(renkler[k][1]==0)
{
renkler[k][1]=deger;
tamam=false;
}
else
{
if(komsumu(deger,k)!=0)
{
yerlestir(deger,k);
tamam=false;
}
else
{
k++;
}
}
}
}
}
for(i=0;i<15;i++)
{
suiListBox3->Items->Add(renkler[i][0]);

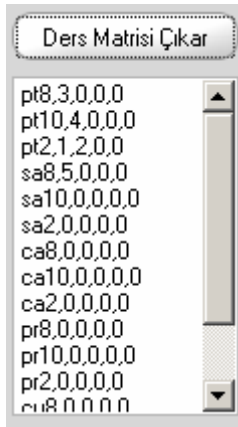
```

```

suiListBox3->Items-
>Strings[i]=suiListBox3->Items-
>Strings[i]+'+'renkler[i][1];
suiListBox3->Items-
>Strings[i]=suiListBox3->Items-
>Strings[i]+'+'renkler[i][2];
suiListBox3->Items-
>Strings[i]=suiListBox3->Items-
>Strings[i]+'+'renkler[i][3];
suiListBox3->Items-
>Strings[i]=suiListBox3->Items-
>Strings[i]+'+'renkler[i][4];
}

```

Ve sonuçta kullanıcı aşağıdaki bir görüntüyle karşılaşır (Şekil 14):



Şekil 14: Ders matrisi çıkar

Haftalık Ders Programı				
	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
Pazartesi 8:00			Mikroişlemciler	
Pazartesi 10:00				Otomatik Kontrol
Pazartesi 12:00	Sayısal Haberleşme	Elektronik Tüm D		
Pazartesi 14:00		İç Tesisat Projesi		
Salı 8:00				
Salı 10:00				
Salı 12:00				
Salı 14:00				

Şekil-15- Ders programının yerleşik hali

## 7. SONUÇ

Sonuç olarak bu çalışmada üniversitelerin büyük bir sıkıntısı olan ders çakışmasını ortadan kaldıran bir yazılım programı geliştirilmiştir.. Geliştirilen bu programın kullanımı basit ve son derece işlevseldir.

## KAYNAKLAR

- [1] ÇÖLKESEN, R, Veri Yapıları ve Algoritmalar-2002
- [2] [www.delphiturkiye.com](http://www.delphiturkiye.com)
- [3] UYSAL, M, C ile Programlama-2000
- [4] YANIK, M, Borland C++ Builder-2000