

02.01.2021

YILDIZ TEKNİK ÜNİVERSİTESİ  
ELEKTRİK-ELEKTRONİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM3021 ALGORİTMA ANALİZİ 4.ÖDEV RAPORU

Backtracking Yöntemi ile Renk Matrisi Oluşturma

AHMET SAİD SAĞLAM

17011501

## KONU

### Backtracking

Bu ödevde,  $N \times N$  boyutundaki matriste her satırda aynı  $N$  rengin farklı sıra ile dizildiği durumda, her sütunda her renkten sadece 1 adet olacak şekilde matrisin geri-izleme (backtracking) yöntemi ile rekürsif olarak düzenlenmesi istenmiştir.

## ÇÖZÜM

### Kütüphane Eklenmesi ve Bilgi Yorumları

```
1  /*
2  @file
3  BLM3021 2020-2021 GÜZ ODEV-4
4  Bu programda verilen N boyutuna(3 <= N <= 8) göre NXN boyutunda bir renk matrisi oluşturulmuş ve renk bilgileri kullanicidan alınarak doldurulmuştur.
5  Bu matrisin her bir sütununda geçen renkler 1'den fazla kez tekrar etmek zorunda mı değil mi backtracking yöntemiyle kontrol edilmiştir.
6  İlgili durum bilgisi ve matrisin son hali ekrana yazdırılmıştır.
7
8  @author
9  İsim: Ahmet Said SAĞLAM
10 Öğrenci No: 17011501
11 Tarih: 02.01.2021
12 E-Mail: l1117501@std.yildiz.edu.tr
13 Compiler: TDM-GCC 4.9.2 64 bit-Release
14 IDE: DEV-C++ (version 5.11)
15 İşletim Sistemi: Windows 10 Pro 64 bit
16 */
17
18 #include <stdio.h>
19 #include <stdlib.h>
20 #include <string.h>
21 #include <conio.h>
22 #include <stdbool.h>
23 #include <ctype.h>
24
```

Kodun ilk kısmında gerekli olabilecek kütüphaneler eklenmiş ve programın geliştiricisi ve geliştirilip çalıştırıldığı ortam hakkında bilgiler verilmiştir.

## colorIntAllocator Fonksiyonu

```
25 //int renk matrisi icin yer acan fonksiyon
26 int **colorIntAllocator(int N) {
27     int **colorMatrix; //matris icin yer acmak adina pointer to pointer
28     int i; //dongu degiskeni
29
30     //calloc fonksiyonu ile dinamik olarak renk matrisinin sayisal degerlerine karsilik olacak matris icin yer acilir
31     colorMatrix = (int**) calloc(N, sizeof(int*));
32     for(i = 0; i < N; i++) {
33         colorMatrix[i] = (int*) calloc(N, sizeof(int));
34     }
35     return colorMatrix; //matris dondurulur
36 }
```

colorIntAllocator fonksiyonu, kullanıcının girdiği renkler için renklerin int değer karşılıklarını tutacak olan ve üzerinde işlemlerin yapılacağı matris için yer açan fonksiyondur. İçerisine matrisin boyutunu belirlemek için kullanıcının girdiği N değerini alır ve dışarıya yer açtığı matrisi döner.

## getColors Fonksiyonu

```
39 //kullanicidan renkler alinarak renk matrisi doldurulur
40 void getColors(int N, char colors[8][8], int **colorMatrix) {
41     int i, j; //dongu degiskenleri
42     int x = 0;
43     char *temp = (char*) calloc(8, sizeof(char)); //kullanicidan alinacak renk icin char dizisi
44     //bilgilendirme printleri
45     printf("\n");
46     printf("Secebileceginiz renkler : ");
47     for(i = 0; i < 7; i++) {
48         printf("%s, ", colors[i]);
49     }
50     printf("%s\n\n", colors[7]);
51     //matrisin her bir gozunu gez
52     for(i = 0; i < N; i++) {
53         for(j = 0; j < N; j++) {
54             printf("Renk Matrisinin %d. satir - %d. sutun rengini giriniz : ", i+1, j+1);
55             scanf("%s", temp); //ilgili goz icin gecerli renk degerini al
56             printf("\n");
57             //colorMatrix ilgili rengin int degeri ile doldurulur
58             x = 0;
59             //renklerle karsilastir
60             while(strcmp(colors[x], temp) != 0) {
61                 x++;
62             }
63             colorMatrix[i][j] = x; //hangi renk degerine karsiliksa o rengin int degeri ile matrisin ilgili gozu doldur
64         }
65     }
66     free(temp); //free islemi
67 }
```

getColors fonksiyonu kullanıcıdan matrisin her bir gözü için renkleri string olarak alır ve bunları colors matrisindeki renklerle karşılaştırır. Sistemdeki renklerin bu matristeki indislerine göre her renk için bir int değer karşılığı ile colorMatrix matrisi doldurulur.

## printColors Fonksiyonu

```
69 //renk matrisini ekrana yazdıran fonksiyon
70 void printColors(int N, int **colorMatrix, char colors[8][8]) {
71     int i, j;
72     // printf("INT MATRIS : \n");
73     // for(i = 0; i < N; i++) {
74     //     for(j = 0; j < N; j++) {
75     //         printf("%d\t", colorMatrix[i][j]);
76     //     }
77     //     printf("\n");
78     // }
79     //int renk matrisinin her bir gozunu gez
80     printf("RENK TABLOSU : \n");
81     for(i = 0; i < N; i++) {
82         for(j = 0; j < N; j++) {
83             printf("%s\t", colors[colorMatrix[i][j]]); //o gozdeki int degere karsilik rengi yazdir
84         }
85         printf("\n");
86     }
87 }
88 }
```

printColors fonksiyonu renk matrisini string olarak (renkleri isimleri ile) ekrana yazdırır.

## shiftRight Fonksiyonu

```
90 //verilen satirdaki elemanlari 1 kez saga shift eden fonksiyon
91 void shiftRight(int **colorMatrix, int N, int row_number) {
92     int i; //dongu degiskeni
93     int temp; //temporary degisken
94     temp = colorMatrix[row_number][N-1]; //matrisin o satirindaki son eleman saklanir
95     //matrisin ilgili satiri son elemandan basa dogru 1 kez saga kaydirilir
96     for(i = N-2; i >= 0; i--) {
97         colorMatrix[row_number][i+1] = colorMatrix[row_number][i];
98     }
99     colorMatrix[row_number][0] = temp; //satirdaki son eleman basa donmeli. Ilk elemanin yerine yerlestirilir
100 }
```

shiftRight fonksiyonu verilen int matrisin ilgili satırındaki değerleri 1 kez sağa shift eder.

## isSuitable Fonksiyonu

```
102 //ust satirlardaki kolon degerleri verilen satirdaki kolon degerlerine esit degilse 1, esitlik bulunursa 0 donduren fonksiyon
103 int isSuitable(int **colorMatrix, int N, int row_number) {
104     int i, j;
105     //0. satir degil ise ust satirlara bakmak icin dongulere girilir
106     if(row_number > 0) {
107         //ilgili satirdan bir oncesi satirdan itibaren
108         for(i = row_number - 1; i >= 0; i--) {
109             //sutunlari gez
110             for(j = 0; j < N; j++) {
111                 //eger esitlik varsa matris uygun degildir
112                 if(colorMatrix[row_number][j] == colorMatrix[i][j]) {
113                     return 0; //0 don
114                 }
115             }
116         }
117     }
118     return 1; //esitlige rastlanmadi ise veya zaten 0. satir kontrol ediliyorsa uygunlugu bildirmek icin 1 dondur
119 }
```

isSuitable fonksiyonu int matriste ilgili satırdan itibaren üst satırlara bakar. Sütunlardaki int değerler sadece 1 kez tekrar ediyorsa 1, birden fazla kez tekrar ediyorsa 0 döndürür. En üst satır kontrol ediliyorsa yine dışarı 1 döndürür.

## backTrack Fonksiyonu

```
121 //recursive bir sekilde matrisin satirlarini sıra ile gezen ve gerektiğinde backtrack ile bir ust satira gecerek yeniden shift ve kontrol yapan fonksiyon
122 //matris uygunsa main'e 1 degil ise 0 dondurur
123 int backTrack(int **colorMatrix, char colors[8][8], int N, int row_number) {
124     int i; //dongu degiskeni
125     int control = 0; //donus degerini kontrol eden degisken / 0 ile ilklendirilir
126     //matrisin satirlari recursive sekilde ilerlendiginde son satir gecildi ise matris uygundur
127     if(row_number >= N) {
128         //printf("Matris uygun!\n");
129         return 1; //matris uygunsa 1 dondur
130     }
131     else {
132         //sutun sayisi kadar donen for
133         for(i = 0; i < N; i++) {
134             //ilgili satir uygun mu diye kontrol edilir (donus degeri 1 degilse)
135             if(isSuitable(colorMatrix, N, row_number) == 1 && control == 0) {
136                 //satir uygunsa bilgilendirme printi atilir(matris yazdirilir)
137                 if(row_number != 0) {
138                     printf("\nktd. satir uygun!\n", row_number + 1);
139                     printColors(N, colorMatrix, colors);
140                 }
141                 control = backTrack(colorMatrix, colors, N, row_number + 1); //fonksiyon bir sonraki satira gecmek icin kendini cagirir
142             }
143             //satir uygun degilse ve for dongusunda satir sayisinin henuz 1 eksigi kadar donulmediyse
144             if(i < N-1 && control == 0) {
145                 shiftRight(colorMatrix, N, row_number); //satiri 1 kez saga shift et
146             }
147         }
148     }
149     return control; //fonksiyon control degiskenini dondurur
150 }
151 }
```

backTrack fonksiyonu esas işlemlerin gerçekleştiği ve recursion'ın olduğu fonksiyondur. İçerisine aldığı int renk matrisinin her satırı için isSuitable fonksiyonun çağırır. Eğer isSuitable'dan 1 değeri döndüyse recursive bir şekilde fonksiyon kendini sıradaki satırı kontrol etmek için çağırır. Eğer isSuitable'dan 0 döndüyse ilgili satır en fazla sütun sayısının 1 eksigi kadar sağa shift edilir ve her shift işleminden sonra yeniden isSuitable ile uygunluk kontrol edilir. Eğer sütunlarda renkler birden fazla kez tekrar ediyorsa fonksiyon 0 döndürür. Böylece 1 üst satıra geçilmiş olur ve o satır sağa shift edilerek işlemler yeniden yapılır. Hiçbir türlü uygunluk elde edilemiyorsa en sonunda fonksiyon dışarıya 0 dönmüş olur.

Eğer ki son satır da uygun ise böylece fonksiyon dışarı 1 dönerek maine kadar gelir ve maine 1 dönmüş olur. Böylece main fonksiyona dönüldüğünde matrisin son hali ve matrisin renkler bakımından uygun şartları sağlayıp sağlamadığının bilgisi sistemde bulunuyor olmaktadır. Fonksiyonda ilk (en üstteki) satır hariç her satır için uygunluk bulunduğunda matrisin o anki hali ekrana printColors fonksiyonu ile yazdırılır.

## main Fonksiyonu

```
155 int main() {
156
157     char colors[8][8] = {
158         "kirmizi",
159         "yesil",
160         "mavi",
161         "mor",
162         "pembe",
163         "sari",
164         "turuncu",
165         "siyah"
166     }; //sistemdeki renkleri tutan matris
167
168     int i; //dongu degiskeni
169     int N; //kullanıcının girecegi renk matrisinin (NxN) boyutu
170     int **colorMatrix; //renk matrisinin sayi karsiliklarini tutacak olan matris
171
172     int cont = 1; //yeni girisler icin kullanıcının istegini kontrol eden degisken
173     int control; //recursive fonksiyondan donen degeri kontrol etmek icin degisken
```

main'de ilk olarak sistemdeki renkler tanımlanır ve bir char matriste tutulur. Döngü değişkenleri ve N boyutu için tanımlamalardan sonra renk matrisinin int karşılıklarını tutacak olan colorMatrix için yer açılmak üzere pointer to pointer tanımlanır. cont değişkeni kullanıcının yeni hesaplamalar yapıp yapmayacağını bilgisini tutarken control ise backTrack fonksiyonundan dönen değeri tutar.

```

176 while(cont) {
177
178     printf("Renk matrisinin boyutunu (N) 3 ile 8 arasında bir deger olacak sekilde giriniz : ");
179     scanf("%d",&N); //matrisin boyutu alinir
180
181     colorMatrix = colorIntAllocator(N); //renk numaralarını tutan matris için yer acilir
182
183     getColors(N, colors, colorMatrix); //renkler kullanicidan alinir
184
185     printf("\n\nGirmis oldugunuz renk tablosu : \n\n");
186     printColors(N, colorMatrix, colors);
187     printf("\nHesaplamalar basliyor...\n\n");
188
189     control = backTrack(colorMatrix, colors, N, 0); //recursive cozum fonksiyonu cagirilir ve sonuc alinir
190     //matriste her sutundaki renkler bir defa tekrar ediyorsa gecerli sonuc uretilmistir
191     printf("\nHesaplamalar sonlandi...\n");
192     if(control) {
193         printf("\nMatriste her sutundaki renkler birbirlerinden farkli!\nSart saglandi! Renk Matrisi : \n\n");
194         printColors(N, colorMatrix, colors); //renk matrisi ekrana yazdirilir
195     }
196     //matriste her sutundaki renkler birden fazla kez tekrar ediyorsa gecerli sonuc uretilememistir
197     else {
198         printf("\nMatriste her sutundaki renkler birbirlerinden farkli olamiyor!\nSart saglanamadi!");
199     }
200
201
202     //dinamik olarak acilan yerler free edilir
203     for(i = 0; i < N; i++) {
204         free(colorMatrix[i]);
205     }
206     free(colorMatrix);
207
208     printf("\n\nYeni bir renk matrisi girmek için 1'e, cikmak için 0'a basiniz : ");
209     scanf("%d",&cont); //kullanicidan yeni veri girişi olup olmayacağını bilgisi alinir
210     printf("\n");
211 }
212
213 return 0; //end of main function
214 }

```

Kullanıcı istediği sürece dönecek olan bir while döngüsünün içinde kullanıcıdan önce N değeri alınır ve matris için yer açılır. Ardından renk matrisi için renkler kullanıcıdan alınır ve matrisin ilk hali ekrana yazdırılır. Hemen peşine backTrack fonksiyonu çağırılarak hesaplamalar başlar ve fonksiyonun dönüş değerine göre kullanıcıya bilgi verilir. Kullanıcıya yeni veri girişi yapıp yapmayacağı sorulur ve alınan cevaba göre yeni veriler kullanıcıdan alınır veya program sonlandırılır.

## PROGRAM ÇIKTILARI

### N = 4 için Sonuç Elde Edilen Renk Matrisi

```
C:\Users\Lenovo\Desktop\alg4\17011501.exe
Renk matrisinin boyutunu (N) 3 ile 8 arasında bir deger olacak sekilde giriniz : 4
Secebileceginiz renkler : kirmizi, yesil, mavi, mor, pembe, sari, turuncu, siyah
Renk Matrisinin 1. satir - 1. sutun rengini giriniz : sari
Renk Matrisinin 1. satir - 2. sutun rengini giriniz : mavi
Renk Matrisinin 1. satir - 3. sutun rengini giriniz : mor
Renk Matrisinin 1. satir - 4. sutun rengini giriniz : yesil
Renk Matrisinin 2. satir - 1. sutun rengini giriniz : sari
Renk Matrisinin 2. satir - 2. sutun rengini giriniz : mor
```

```
Renk Matrisinin 4. satir - 3. sutun rengini giriniz : sari
Renk Matrisinin 4. satir - 4. sutun rengini giriniz : yesil
```

Girmis oldugunuz renk tablosu :

```
RENK TABLOSU :
sari    mavi    mor    yesil
sari    mor    yesil  mavi
sari    mavi    yesil  mor
mor     mavi    sari   yesil
```

Hesaplamalar basliyor...



Hesaplamalar basliyor...

2. satir uygun!

RENK TABLOSU :

sari	mavi	mor	yesil
mor	yesil	mavi	sari
sari	mavi	yesil	mor
mor	mavi	sari	yesil

3. satir uygun!

RENK TABLOSU :

sari	mavi	mor	yesil
mor	yesil	mavi	sari
yesil	mor	sari	mavi
mor	mavi	sari	yesil

4. satir uygun!

RENK TABLOSU :

sari	mavi	mor	yesil
mor	yesil	mavi	sari
yesil	mor	sari	mavi
mavi	sari	yesil	mor

Hesaplamalar sonlandi...

Hesaplamalar sonlandi...

Matriste her sutundaki renkler birbirlerinden farkli!

Sart saglandi! Renk Matrisi :

RENK TABLOSU :

sari	mavi	mor	yesil
mor	yesil	mavi	sari
yesil	mor	sari	mavi
mavi	sari	yesil	mor

Yeni bir renk matrisi girmek icin 1'e, cikmak icin 0'a basiniz : 0

-----

Process exited after 335.2 seconds with return value 0

Press any key to continue . . .

## N = 5 için Sonuç Elde Edilen Renk Matrisi

```
C:\Users\Lenovo\Desktop\alg4\17011501.exe
Renk matrisinin boyutunu (N) 3 ile 8 arasında bir deger olacak sekilde giriniz : 5
Secebileceginiz renkler : kirmizi, yesil, mavi, mor, pembe, sari, turuncu, siyah
Renk Matrisinin 1. satir - 1. sutun rengini giriniz : pembe
Renk Matrisinin 1. satir - 2. sutun rengini giriniz : siyah
Renk Matrisinin 1. satir - 3. sutun rengini giriniz : turuncu
Renk Matrisinin 1. satir - 4. sutun rengini giriniz : yesil
Renk Matrisinin 1. satir - 5. sutun rengini giriniz : sari
Renk Matrisinin 2. satir - 1. sutun rengini giriniz : pembe
```

```
Renk Matrisinin 5. satir - 4. sutun rengini giriniz : yesil
Renk Matrisinin 5. satir - 5. sutun rengini giriniz : sari
```

Girmis oldugunuz renk tablosu :

RENK TABLOSU :

pembe	siyah	turuncu	yesil	sari
pembe	siyah	turuncu	yesil	sari
pembe	siyah	turuncu	yesil	sari
pembe	siyah	turuncu	yesil	sari
pembe	siyah	turuncu	yesil	sari

Hesaplamalar basliyor...

Hesaplamalar basliyor...

2. satir uygun!

RENK TABLOSU :

pembe	siyah	turuncu	yesil	sari
sari	pembe	siyah	turuncu	yesil
pembe	siyah	turuncu	yesil	sari
pembe	siyah	turuncu	yesil	sari
pembe	siyah	turuncu	yesil	sari

3. satir uygun!

RENK TABLOSU :

pembe	siyah	turuncu	yesil	sari
sari	pembe	siyah	turuncu	yesil
yesil	sari	pembe	siyah	turuncu
pembe	siyah	turuncu	yesil	sari
pembe	siyah	turuncu	yesil	sari

4. satir uygun!

RENK TABLOSU :

pembe	siyah	turuncu	yesil	sari
sari	pembe	siyah	turuncu	yesil
yesil	sari	pembe	siyah	turuncu
turuncu	yesil	sari	pembe	siyah
pembe	siyah	turuncu	yesil	sari

4. satir uygun!

RENK TABLOSU :

pembe	siyah	turuncu	yesil	sari
sari	pembe	siyah	turuncu	yesil
yesil	sari	pembe	siyah	turuncu
turuncu	yesil	sari	pembe	siyah
pembe	siyah	turuncu	yesil	sari

5. satir uygun!

RENK TABLOSU :

pembe	siyah	turuncu	yesil	sari
sari	pembe	siyah	turuncu	yesil
yesil	sari	pembe	siyah	turuncu
turuncu	yesil	sari	pembe	siyah
siyah	turuncu	yesil	sari	pembe

Hesaplamalar sonlandi...

```
Hesaplamalar sonlandi...

Matriste her sutundaki renkler birbirlerinden farkli!
Sart saglandi! Renk Matrisi :

RENK TABLOSU :
pembe   siyah   turuncu  yesil   sari
sari     pembe   siyah    turuncu yesil
yesil    sari    pembe    siyah   turuncu
turuncu  yesil    sari     pembe   siyah
siyah    turuncu yesil    sari    pembe

Yeni bir renk matrisi girmek icin 1'e, cikmak icin 0'a basiniz : 0

-----
Process exited after 185.7 seconds with return value 0
Press any key to continue . . .
```

---

## N = 6 için Sonuç Elde Edilen Renk Matrisi

```
C:\Users\Lenovo\Desktop\alg4\17011501.exe
Renk matrisinin boyutunu (N) 3 ile 8 arasında bir deger olacak sekilde giriniz : 6

Secebileceginiz renkler : kirmizi, yesil, mavi, mor, pembe, sari, turuncu, siyah

Renk Matrisinin 1. satir - 1. sutun rengini giriniz : kirmizi
Renk Matrisinin 1. satir - 2. sutun rengini giriniz : mavi
Renk Matrisinin 1. satir - 3. sutun rengini giriniz : mor
Renk Matrisinin 1. satir - 4. sutun rengini giriniz : siyah
Renk Matrisinin 1. satir - 5. sutun rengini giriniz : yesil
Renk Matrisinin 1. satir - 6. sutun rengini giriniz : pembe
Renk Matrisinin 2. satir - 1. sutun rengini giriniz : pembe
```

Renk Matrisinin 6. satir - 5. sutun rengini giriniz : yesil

Renk Matrisinin 6. satir - 6. sutun rengini giriniz : pembe

Girmis oldugunuz renk tablosu :

RENK TABLOSU :

kirmizi	mavi	mor	siyah	yesil	pembe
pembe	kirmizi	mavi	mor	siyah	yesil
kirmizi	mavi	mor	siyah	yesil	pembe
yesil	pembe	kirmizi	mavi	mor	siyah
siyah	yesil	pembe	kirmizi	mavi	mor
kirmizi	mavi	mor	siyah	yesil	pembe

Hesaplamalar basliyor...

Hesaplamalar basliyor...

2. satir uygun!

RENK TABLOSU :

kirmizi	mavi	mor	siyah	yesil	pembe
pembe	kirmizi	mavi	mor	siyah	yesil
kirmizi	mavi	mor	siyah	yesil	pembe
yesil	pembe	kirmizi	mavi	mor	siyah
siyah	yesil	pembe	kirmizi	mavi	mor
kirmizi	mavi	mor	siyah	yesil	pembe

3. satir uygun!

RENK TABLOSU :

kirmizi	mavi	mor	siyah	yesil	pembe
pembe	kirmizi	mavi	mor	siyah	yesil
yesil	pembe	kirmizi	mavi	mor	siyah
yesil	pembe	kirmizi	mavi	mor	siyah
siyah	yesil	pembe	kirmizi	mavi	mor
kirmizi	mavi	mor	siyah	yesil	pembe

4. satir uygun!

RENK TABLOSU :

kirmizi	mavi	mor	siyah	yesil	pembe
pembe	kirmizi	mavi	mor	siyah	yesil
yesil	pembe	kirmizi	mavi	mor	siyah
siyah	yesil	pembe	kirmizi	mavi	mor
siyah	yesil	pembe	kirmizi	mavi	mor
kirmizi	mavi	mor	siyah	yesil	pembe

```
5. satir uygun!
RENK TABLOSU :
kirmizi mavi mor siyah yesil pembe
pembe kirmizi mavi mor siyah yesil
yesil pembe kirmizi mavi mor siyah
siyah yesil pembe kirmizi mavi mor
mor siyah yesil pembe kirmizi mavi
kirmizi mavi mor siyah yesil pembe

6. satir uygun!
RENK TABLOSU :
kirmizi mavi mor siyah yesil pembe
pembe kirmizi mavi mor siyah yesil
yesil pembe kirmizi mavi mor siyah
siyah yesil pembe kirmizi mavi mor
mor siyah yesil pembe kirmizi mavi
mavi mor siyah yesil pembe kirmizi

Hesaplamalar sonlandi...
```

```
Hesaplamalar sonlandi...

Matriste her sutundaki renkler birbirlerinden farkli!
Sart saglandi! Renk Matrisi :

RENK TABLOSU :
kirmizi mavi mor siyah yesil pembe
pembe kirmizi mavi mor siyah yesil
yesil pembe kirmizi mavi mor siyah
siyah yesil pembe kirmizi mavi mor
mor siyah yesil pembe kirmizi mavi
mavi mor siyah yesil pembe kirmizi

Yeni bir renk matrisi girmek icin 1'e, cikmak icin 0'a basiniz : 1

Renk matrisinin boyutunu (N) 3 ile 8 arasinda bir deger olacak sekilde giriniz :
```

## N = 4 için Sonuç Elde Edilemeyen Renk Matrisi

```
C:\Users\Lenovo\Desktop\alg4\17011501.exe
Renk matrisinin boyutunu (N) 3 ile 8 arasında bir deger olacak sekilde giriniz : 4
Secebileceginiz renkler : kirmizi, yesil, mavi, mor, pembe, sari, turuncu, siyah
Renk Matrisinin 1. satir - 1. sutun rengini giriniz : sari
Renk Matrisinin 1. satir - 2. sutun rengini giriniz : mavi
Renk Matrisinin 1. satir - 3. sutun rengini giriniz : mor
Renk Matrisinin 1. satir - 4. sutun rengini giriniz : yesil
Renk Matrisinin 2. satir - 1. sutun rengini giriniz : sari
Renk Matrisinin 2. satir - 2. sutun rengini giriniz : mor
```

```
Renk Matrisinin 4. satir - 4. sutun rengini giriniz : sari
```

Girmis oldugunuz renk tablosu :

RENK TABLOSU :

sari	mavi	mor	yesil
sari	mor	yesil	mavi
sari	mavi	yesil	mor
mavi	mor	yesil	sari

Hesaplamalar basliyor...

Hesaplamalar basliyor...

2. satir uygun!

RENK TABLOSU :

sari	mavi	mor	yesil
mor	yesil	mavi	sari
sari	mavi	yesil	mor
mavi	mor	yesil	sari

3. satir uygun!

RENK TABLOSU :

sari	mavi	mor	yesil
mor	yesil	mavi	sari
yesil	mor	sari	mavi
mavi	mor	yesil	sari

2. satir uygun!

RENK TABLOSU :

yesil	sari	mavi	mor
sari	mor	yesil	mavi
mavi	yesil	mor	sari
mor	yesil	sari	mavi

3. satir uygun!

RENK TABLOSU :

yesil	sari	mavi	mor
sari	mor	yesil	mavi
mavi	yesil	mor	sari
mor	yesil	sari	mavi

2. satir uygun!

RENK TABLOSU :

mor	yesil	sari	mavi
mavi	sari	mor	yesil
yesil	mor	sari	mavi
yesil	sari	mavi	mor

3. satir uygun!

RENK TABLOSU :

mor	yesil	sari	mavi
mavi	sari	mor	yesil
sari	mavi	yesil	mor
yesil	sari	mavi	mor

2. satir uygun!

RENK TABLOSU :

mavi	mor	yesil	sari
yesil	mavi	sari	mor
mor	sari	mavi	yesil
sari	mavi	mor	yesil

3. satir uygun!

RENK TABLOSU :

mavi	mor	yesil	sari
yesil	mavi	sari	mor
mor	sari	mavi	yesil
sari	mavi	mor	yesil

Hesaplamalar sonlandi...

Hesaplamalar sonlandi...

Matriste her sutundaki renkler birbirlerinden farkli olamiyor!

Sart saglanamadi!

Yeni bir renk matrisi girmek icin 1'e, cikmak icin 0'a basiniz :



## SOURCE CODE

```
1.  /*
2.  @file
3.  BLM3021 2020-2021 GUZ ODEV-4
4.  Bu programda verilen N boyutuna(3 <= N <= 8) göre NXN boyutunda bir renk matrisi olu
   sturulmus ve renk bilgileri kullanicidan alinarak doldurulmustur.
5.  Bu matrisin her bir sutununda gecen renkler 1'den fazla kez tekrar etmek zorunda mı
   degil mi backtracking yontemiyle kontrol edilmistir.
6.  İlgili durum bilgisi ve matrisin son hali ekrana yazdirilmistir.
7.
8.  @author
9.  İsim: Ahmet Said SAĞLAM
10. Öğrenci No: 17011501
11. Tarih: 02.01.2021
12. E-Mail: 11117501@std.yildiz.edu.tr
13. Compiler: TDM-GCC 4.9.2 64 bit-Release
14. IDE: DEV-C++ (version 5.11)
15. İşletim Sistemi: Windows 10 Pro 64 bit
16. */
17.
18. #include <stdio.h>
19. #include <stdlib.h>
20. #include <string.h>
21. #include <conio.h>
22. #include <stdbool.h>
23. #include <ctype.h>
24.
25. //int renk matrisi icin yer acan fonksiyon
26. int **colorIntAllocator(int N) {
27.     int **colorMatrix; //matris icin yer acmak adina pointer to pointer
28.     int i; //dongu degiskeni
29.
30.     //calloc fonksiyonu ile dinamik olarak renk matrisinin sayisal degerlerine karsi
   lik olacak matris icin yer acilir
31.     colorMatrix = (int**) calloc(N, sizeof(int*));
32.     for(i = 0; i < N; i++) {
33.         colorMatrix[i] = (int*) calloc(N, sizeof(int));
34.     }
35.     return colorMatrix; //matris dondurulur
36. }
37.
38.
39. //kullanicidan renkler alinarak renk matrisi doldurulur
40. void getColors(int N, char colors[8][8], int **colorMatrix) {
41.     int i, j; //dongu degiskenleri
42.     int x = 0; //colors matrisinin satirlarina erismek icin kullanılan degisken
43.     char *temp = (char*) calloc(8, sizeof(char)); //kullanicidan alinacak renk ici
   n char dizisi
44.     //bilgilendirme printleri
45.     printf("\n");
46.     printf("Secebileceginiz renkler : ");
47.     for(i = 0; i < 7; i++) {
48.         printf("%s, ", colors[i]);
49.     }
50.     printf("%s\n\n", colors[7]);
51.     //matrisin her bir gozunu gez
52.     for(i = 0; i < N; i++) {
53.         for(j = 0; j < N; j++) {
54.             printf("Renk Matrisinin %d. satir -
   %d. sutun rengini giriniz : ", i+1, j+1);
55.             scanf("%s", temp); //ilgili goz icin gecerli renk degerini al
56.             printf("\n");
```

```

57.         //colorMatrix ilgili rengin int degeri ile doldurulur
58.         x = 0; //0. satirdan itibaren colors'i gez
59.         //renklerle karsilastir
60.         while(strcmp(colors[x], temp) != 0) {
61.             x++;
62.         }
63.         colorMatrix[i][j] = x; //hangi renk degerine karsiliksa o rengin int de
geri ile matrisin ilgili gozu doldir
64.     }
65. }
66. free(temp); //free islemi
67. }
68.
69. //renk matrisini ekrana yazdiran fonksiyon
70. void printColors(int N, int **colorMatrix, char colors[8][8]) {
71.     int i, j;
72.     // printf("INT MATRIS : \n");
73.     // for(i = 0; i < N; i++) {
74.     //     for(j = 0; j < N; j++) {
75.     //         printf("%d\t",colorMatrix[i][j]);
76.     //     }
77.     //     printf("\n");
78.     // }
79.     //int renk matrisinin her bir gozunu gez
80.     printf("RENK TABLOSU : \n");
81.     for(i = 0; i < N; i++) {
82.         for(j = 0; j < N; j++) {
83.             printf("%s\t",colors[colorMatrix[i][j]]);    //o gozdeki int degere karsi
lik rengi yazdir
84.         }
85.         printf("\n");
86.     }
87.
88. }
89.
90. //verilen satirdaki elemanlari 1 kez saga shift eden fonksiyon
91. void shiftRight(int **colorMatrix, int N, int row_number) {
92.     int i; //dongu degiskeni
93.     int temp; //temporary degisken
94.     temp = colorMatrix[row_number][N-
1]; //matrisin o satirindaki son eleman saklanir
95.     //matrisin ilgili satiri son elemandan basa dogru 1 kez saga kaydirilir
96.     for(i = N-2; i >= 0; i--) {
97.         colorMatrix[row_number][i+1] = colorMatrix[row_number][i];
98.     }
99.     colorMatrix[row_number][0] = temp; //satirdaki son eleman basa donmeli. Ilk ele
manin yerine yerlestirilir
100. }
101.
102. //ust satirlardaki kolon degerleri verilen satirdaki kolon degerlerine esit d
egilse 1, esitlik bulunursa 0 donduren fonksiyon
103. int isSuitable(int **colorMatrix, int N, int row_number) {
104.     int i, j;
105.     //0. satir degil ise ust satirlara bakmak icin dongulere girilir
106.     if(row_number > 0 ) {
107.         //ilgili satirdan bir onceki satirdan itibaren
108.         for(i = row_number - 1; i >= 0; i--) {
109.             //sutunlari gez
110.             for(j = 0; j < N; j++) {
111.                 //eger esitlik varsa matris uygun degildir
112.                 if(colorMatrix[row_number][j] == colorMatrix[i][j]) {
113.                     return 0; //0 don
114.                 }
115.             }
116.         }
117.     }

```

```

118.         return 1; //esitlige rastlanmadi ise veya zaten 0. satir kontrol ediliy
        orsa uygunlugu bildirmek icin 1 dondur
119.     }
120.
121.     //recursive bir sekilde matrisin satirlarini sıra ile gezen ve gerektiğinde b
acktrack ile bir üst satıra geçerek yeniden shift ve kontrol yapan fonksiyon
122.     //matris uygunsa main'e 1 değil ise 0 dondurur
123.     int backtrack(int **colorMatrix, char colors[8][8], int N, int row_number) {
124.         int i; //dongu degiskeni
125.         int control = 0; //donus degerini kontrol eden degisken / 0 ile ilkle
ndirilir
126.         //matrisin satirlari recursive sekilde ilerlendiginde son satir gecildi i
se matris uygundur
127.         if(row_number >= N) {
128.             //printf("Matris uygun!\n");
129.             return 1; //matris uygunsa 1 dondur
130.         }
131.         else {
132.             //sutun sayisi kadar donen for
133.             for(i = 0; i < N; i++) {
134.                 //ilgili satir uygun mu diye kontrol edilir (donus degeri 1 degil
se)
135.                 if(isSuitable(colorMatrix, N, row_number) == 1 && control == 0) {
136.
137.                     //satir uygunsa bilgilendirme printi atilir(matris yazdirilir
)
138.                     if(row_number != 0) {
139.                         printf("\n%d. satir uygun!\n", row_number + 1);
140.                         printColors(N, colorMatrix, colors);
141.                     }
142.                     control = backtrack(colorMatrix, colors, N, row_number + 1);
143.                     //fonksiyon bir sonraki satira gecmek icin kendini cagirir
144.                     //satir uygun degilse ve for dongusunda satir sayisinin henuz 1 e
ksigi kadar donulmediyse
145.                     if(i < N-1 && control == 0) {
146.                         shiftRight(colorMatrix, N, row_number); //satiri 1 kez saga s
hift et
147.                     }
148.                 }
149.             }
150.             return control; //fonksiyon control degiskenini dondurur
151.         }
152.
153.
154.
155.     int main() {
156.
157.         char colors[8][8] = {
158.             "kirmizi",
159.             "yesil",
160.             "mavi",
161.             "mor",
162.             "pembe",
163.             "sari",
164.             "turuncu",
165.             "siyah"
166.         }; //sistemdeki renkleri tutan matris
167.
168.         int i; //dongu degiskeni
169.         int N; //kullanicinin girecegi renk matrisinin (NxN) boyutu
170.         int **colorMatrix; //renk matrisinin sayi karsiliklarini tutacak olan ma
tris
171.

```

```

172.         int cont = 1;    //yeni girisler icin kullanicinin istegini kontrol eden d
           egisken
173.         int control;    //recursive fonksiyondan donen degeri kontrol etmek icin
           degisken
174.
175.         //kullanıcı istediggi surece don
176.         while(cont) {
177.
178.             printf("Renk matrisinin boyutunu (N) 3 ile 8 arasinda bir deger olaca
           k sekilde giriniz : ");
179.             scanf("%d",&N); //matrisin boyutu alinir
180.
181.             colorMatrix = colorIntAllocator(N); //renk numaralarını tutan matris
           icin yer acilir
182.
183.             getColors(N, colors, colorMatrix); //renkler kullanicidan alinir
184.
185.             printf("\n\nGirmis oldugunuz renk tablosu : \n\n");
186.             printColors(N, colorMatrix, colors);
187.             printf("\nHesaplamalar basliyor...\n\n");
188.
189.             control = backTrack(colorMatrix, colors, N, 0); //recursive cozum fon
           ksiyonu cagirilir ve sonuc alinir
190.             //matriste her sutundaki renkler bir defa tekrar ediyorsa gecerli son
           uc uretilmistir
191.             printf("\nHesaplamalar sonlandi...\n");
192.             if(control) {
193.                 printf("\nMatriste her sutundaki renkler birbirlerinden farkli!\n
           Sart saglandi! Renk Matrisi : \n\n");
194.                 printColors(N, colorMatrix, colors);    //renk matrisi ekrana yaz
           dirilir
195.             }
196.             //matriste her sutundaki renkler birden fazla kez tekrar ediyorsa gec
           erli sonuc uretilememistir
197.             else {
198.                 printf("\nMatriste her sutundaki renkler birbirlerinden farkli ol
           amiyor!\nSart saglanamadi!");
199.             }
200.
201.
202.             //dinamik olarak acilan yerler free edilir
203.             for(i = 0; i < N; i++) {
204.                 free(colorMatrix[i]);
205.             }
206.             free(colorMatrix);
207.
208.             printf("\n\nYeni bir renk matrisi girmek icin 1'e, cikmak icin 0'a ba
           siniz : ");
209.             scanf("%d",&cont); //kullanicidan yeni veri girisi olup olmayacagini
           n bilgisi alinir
210.             printf("\n");
211.         }
212.
213.         return 0;    //end of main function
214.     }

```