



ALGORİTMA ANALİZİ

ÖDEV 4 - GRUP 1

İrem ATILGAN

17061036



03.01.2021

1. ÇÖZÜM

1.1 Fonksiyonlar

void printMatrix(int mat, int N)** : Matris yazdırma fonksiyonudur. Matriste hangi sayıya hangi renk denk düşüyorsa o rengin ismi yazdırılır.

int getMatrix(int N)** : Matris okuma fonksiyonudur. Kullanıcıdan alınan değerler tek tek matrise yazılır. Sonuç olarak matris döndürülür.

int* swapRow(int* row, int dim) : Satırın sağa kaydırıldığı fonksiyondur. Parametre olarak satırın adresini ve sütun sayısını alır.

int checkRows(int mat, int index, int dim)** : Çakışma için üst satırların kontrol edildiği fonksiyondur. Parametre olarak sayı matrisini, o anda bulunulan satır indisini ve matris boyutlarını alır. Eğer farklı satır, aynı sütunlarda çakışma bulunursa 1, hiç çakışma bulunamaması durumunda 0 döndülür.

int swapMatrix(int mat, int N, int rowIndex)** : Backtracking yönteminin kullanıldığı fonksiyondur.

- Parametre olarak matrisi, matris boyutunu ve o anda bulunulan satırın indisi alınır.
- Matrisin herhangi bir satırındayken (son satır hariç) üst satırlarla herhangi bir çakışma bulunamamışsa (checkRows fonksiyonu ile kontrol edilir) rowIndex bir artırılarak fonksiyon yeniden çağırılır. Eğer matrisin son satırına gelinmişse ve çakışma bulunamamışsa 1, herhangi bir çakışma bulunması halinde 0 döndülür.
- Fonksiyonun sıfır döndüğü durumlarda, eğer satır sütun sayısı kadar sağa kaydırılmadıysa, yeniden sağa kaydırma işlemi yapılarak üst satırlar kontrol edilir ve program bu şekilde çalışmaya devam eder.
- Sütun sayısı kadar satır sağa kaydırılmışsa, kaydırma işlemi durdurularak 0 döndülür.

2. PROGRAM

2.1 Program Kodları

2.1.1 main Fonksiyonu

```
126 int main()
127 {
128     int N; //Matris satir/sutun sayisi
129     int result; //Sutunlari cakismayan bir matrisin olusturulup olusturulamadiginin sonucunu tutan degisken
130     int** mat; //Renk degerlerinin tutuldugu sayi matrisi
131
132     printf("\n\t\t\t");
133     printf("=== KURALLAR ===");
134     printf("\n\t\t\t");
135     printf("1. Renk Listesi");
136     printf("\n\t\t\t");
137     printf("1 = Kirmizi\t2 = Yesil\t3 = Mavi\t4 = Mor\t5 = Turuncu\t6 = Siyah\t7 = Beyaz\t8 = Sari");
138     printf("\n\t\t\t");
139     printf("Matrisin boyutlarini girip matrisi doldurduktan sonra saga kaydirma islemi gerceklestirebilirsiniz.");
140     printf("\n\t\t\t");
141     printf("Renk matrisinin boyutlarini giriniz (3 <= N <= 8) : ");
142     scanf("%d",&N);
143
144     mat = getMatrix(N); //Matris degerleri kullanicidan alinir
145
146     printf("\n\t\t\t\tBASLANGIC MATRISI = ");
147     printf("\n\t\t\t\t");
148     printMatrix(mat,N);
149     printf("\n\t\t\t\t");
150     result = swapMatrix(mat,N,1); //Backtracking fonksiyonu
151     if(result) //Eger istediklerimize uyan matris olusturulabildiyse yazdirilir
152     {
153         printf("\n\t\t\t\tSONUC MATRISI = ");
154         printf("\n\t\t\t\t");
155         printMatrix(mat,N);
156     }
157     else
158         printf("Matris donusturulemiyor!..");
159
160     return 0;
161 }
```

2.1.2 getMatrix ve swapRow Fonksiyonu

```
30 //Matris okuma fonksiyonu
31 int** getMatrix(int N)
32 {
33     int i;
34     int j;
35     int** mat; //renklerin tutuldugu sayi matrisi
36     mat = (int**)calloc(N,sizeof(int*));
37     for(i = 0; i < N; i++)
38         mat[i] = (int*)calloc(N,sizeof(int));
39
40     //matris degerleri kullanicidan alinir
41     for(i = 0; i < N; i++)
42     {
43         for(j = 0; j < N; j++)
44         {
45             printf("\n\t\t\t");
46             printf("MAT[%d][%d] = ",i,j);
47             scanf("%d",&mat[i][j]);
48         }
49     }
50     return mat;
51 }
52
53 //Satir kaydirma fonksiyonu
54 int* swapRow(int* row, int dim)
55 {
56     int tmp = row[dim-1]; //son sutun elemani gecici degiskende tutulur
57     int i;
58     for(i = dim-1; i > 0; i--)
59     {
60         row[i] = row[i-1]; //son sutundan baslayarak kaydirma yapilir
61     }
62     row[i] = tmp;
63
64     return row;
65 }
```

2.1.3 swapMatrix Fonksiyonu

```
84 //Backtracking fonksiyonu
85 int swapMatrix(int** mat, int N, int rowIndex)
86 {
87     int count; //mevcut satirda kac kez saga kaydirilma islemi yapildigini tutar
88     int result; //ust satirlarla cakisma olup olmadiginin sonucunu tutar
89     int swap_result; //backtracking fonksiyonunun sonucunu tutar
90
91     //Son satir gecilmemisse
92     if(rowIndex <= N)
93     {
94         count = 0;
95         //sutun sayisi kadar saga kaydirma islemi yapilmadiysa
96         while(count < N)
97         {
98             //ust satirlarla cakisma var mi diye kontrol edilir
99             result = checkRows(mat, rowIndex, N);
100             if(result) //cakisma varsa
101             {
102                 mat[rowIndex-1] = swapRow(mat[rowIndex-1],N); //satir saga kaydirilir
103             }
104             else //cakisma yoksa
105             {
106                 swap_result = swapMatrix(mat,N,rowIndex+1); //bir sonraki satir icin fonksiyon cagirilir
107                 if(swap_result) //eger matris basarili bir sekilde ayarlanabildiyse matris yazdirilir ve 1 donulur
108                 {
109                     printMatrix(mat,N);
110                     return 1;
111                 }
112                 mat[rowIndex-1] = swapRow(mat[rowIndex-1],N); //matris ayarlanamadiysa mevcut satir yeniden saga kaydirilir
113                 count++; //yapilan kaydirma islemi icin sayac bir artirilir
114             }
115             return 0;
116         }
117     }
118     else //Son satir gecilmisse matris basarili bir sekilde ayarlanmis demektir
119     {
120         return 1;
121     }
122 }
123
124 }
```

2.1.4 checkRows Fonksiyonu

```
67 //Ust satirlardaki cakismalarin kontrol edildigi fonksiyon
68 int checkRows(int** mat, int index, int dim)
69 {
70     int i;
71     int j;
72     //Mevcut satirdan bir önceki satirla baslanarak kontrol yapilir
73     for(i = index-2; i >= 0; i--)
74     {
75         for(j = 0; j < dim; j++)
76         {
77             if(mat[i][j] == mat[index-1][j])
78                 return 1; //Herhangi iki sutunda cakisma oldugu an 1 donulur
79         }
80     }
81     return 0; //Cakisma hic olmadiyse 0 donulur
82 }
```

2.1.5 printMatrix Fonksiyonu

```
10 //Matris yazdirma fonksiyonu
11 void printMatrix(int** mat, int N)
12 {
13     int i;
14     int j;
15     //sayilarin renk karsiliklarini tutan char matrisi
16     char colors[8][20] = {"Kirmizi", "Yesil", "Mavi", "Mor", "Turuncu", "Siyah", "Beyaz", "Sari"};
17     printf("\n\t\t\t");
18     for(i = 0; i < N; i++)
19     {
20         for(j = 0; j < N; j++)
21         {
22             printf("%s\t\t", colors[mat[i][j]-1]);
23         }
24         printf("\n\t\t\t");
25     }
26 }
27
28 }
```

2.2 Program Çıktıları

```
=== KURALLAR ===
1. Renk Listesi
1 = Kirmizi    2 = Yesil    3 = Mavi    4 = Mor 5 = Turuncu    6 = Siyah    7 = Beyaz    8 = Sari
Matrisin boyutlarini girip matrisi doldurduktan sonra saga kaydirma islemi gerceklestirebilirsiniz.
Renk matrisinin boyutlarini giriniz (3 <= N <= 8) : 4

MAT[0][0] = 4
MAT[0][1] = 5
MAT[0][2] = 6
MAT[0][3] = 7
MAT[1][0] = 4
MAT[1][1] = 7
MAT[1][2] = 1
MAT[1][3] = 2
MAT[2][0] = 3
MAT[2][1] = 1
MAT[2][2] = 2
MAT[2][3] = 6
MAT[3][0] = 4
MAT[3][1] = 5
MAT[3][2] = 2
MAT[3][3] = 3
```

BASLANGIC MATRISI =

Mor	Turuncu	Siyah	Beyaz
Mor	Beyaz	Kirmizi	Yesil
Mavi	Kirmizi	Yesil	Siyah
Mor	Turuncu	Yesil	Mavi
Mor	Turuncu	Siyah	Beyaz
Yesil	Mor	Beyaz	Kirmizi
Mavi	Kirmizi	Yesil	Siyah
Turuncu	Yesil	Mavi	Mor
Mor	Turuncu	Siyah	Beyaz
Yesil	Mor	Beyaz	Kirmizi
Mavi	Kirmizi	Yesil	Siyah
Turuncu	Yesil	Mavi	Mor
Mor	Turuncu	Siyah	Beyaz
Yesil	Mor	Beyaz	Kirmizi
Mavi	Kirmizi	Yesil	Siyah
Turuncu	Yesil	Mavi	Mor

SONUC MATRISI =

Mor	Turuncu	Siyah	Beyaz
Yesil	Mor	Beyaz	Kirmizi
Mavi	Kirmizi	Yesil	Siyah
Turuncu	Yesil	Mavi	Mor

```

=== KURALLAR ===
1. Renk listesi
1 = Kirmizi      2 = Yesil      3 = Mavi      4 = Mor 5 = Turuncu      6 = Siyah      7 = Beyaz      8 = Sari
Matrisin boyutlarini girip matrisi doldurduktan sonra saga kaydirma islemi gerceklestirebilirsiniz.
Renk matrisinin boyutlarini giriniz (3 <= N <= 8) : 3

MAT[0][0] = 1
MAT[0][1] = 2
MAT[0][2] = 3
MAT[1][0] = 3
MAT[1][1] = 2
MAT[1][2] = 1
MAT[2][0] = 1
MAT[2][1] = 3
MAT[2][2] = 2

BASLANGIC MATRISI =

Kirmizi      Yesil      Mavi
Mavi      Yesil      Kirmizi
Kirmizi      Mavi      Yesil

Matris donusturulemiyor!..

```

```

=== KURALLAR ===
1. Renk listesi
1 = Kirmizi      2 = Yesil      3 = Mavi      4 = Mor 5 = Turuncu      6 = Siyah      7 = Beyaz      8 = Sari
Matrisin boyutlarini girip matrisi doldurduktan sonra saga kaydirma islemi gerceklestirebilirsiniz.
Renk matrisinin boyutlarini giriniz (3 <= N <= 8) : 3

MAT[0][0] = 5
MAT[0][1] = 4
MAT[0][2] = 2
MAT[1][0] = 4
MAT[1][1] = 2
MAT[1][2] = 5
MAT[2][0] = 5
MAT[2][1] = 2
MAT[2][2] = 4

BASLANGIC MATRISI =

Turuncu      Mor      Yesil
Mor      Yesil      Turuncu
Turuncu      Yesil      Mor

Matris donusturulemiyor!..

```

```
=== KURALLAR ===
1. Renk Listesi
1 = Kirmizi    2 = Yesil    3 = Mavi    4 = Mor 5 = Turuncu    6 = Siyah    7 = Beyaz    8 = Sari
Matrisin boyutlarini girip matrisi doldurduktan sonra saga kaydirma islemi gerceklestirebilirsiniz.
Renk matrisinin boyutlarini giriniz (3 <= N <= 8) : 5

MAT[0][0] = 3
MAT[0][1] = 4
MAT[0][2] = 5
MAT[0][3] = 2
MAT[0][4] = 1
MAT[1][0] = 2
MAT[1][1] = 6
MAT[1][2] = 1
MAT[1][3] = 4
MAT[1][4] = 5
MAT[2][0] = 2
MAT[2][1] = 1
MAT[2][2] = 3
MAT[2][3] = 6
MAT[2][4] = 7
MAT[3][0] = 5
MAT[3][1] = 4
MAT[3][2] = 2
MAT[3][3] = 1
MAT[3][4] = 6
MAT[4][0] = 4
MAT[4][1] = 5
MAT[4][2] = 1
MAT[4][3] = 2
MAT[4][4] = 3

BASLANGIC MATRISI =

Mavi      Mor      Turuncu    Yesil     Kirmizi
Yesil     Siyah    Kirmizi    Mor       Turuncu
Yesil     Kirmizi   Mavi      Siyah     Beyaz
Turuncu   Mor      Yesil     Kirmizi   Siyah
Mor       Turuncu   Kirmizi   Yesil     Mavi

Matris donusturulemiyor!..
```