

YÖNTEM :**Renkleri Sayılara Çevirme :**

Öncelikle kullanıcıdan N değeri alınır ve dinamik bir matris oluşturulur ardından ilk satır için alınan stringler de ayrı bir renk dizisine atılır ve daha sonrasında tüm satırlar için alınan stringlerin renk dizisinde karşılık geldiği indisler bu matrise aktarılır bu sayede mesela 3X3 lük bir matriste ilk satıra kırmızı yeşil mor girdik, burada kırmızı = 0 , yeşil = 1 , mor = 2 numaralı indise karşılık gelecek şekilde kaydedilir ve tüm satırlarda girilen string (renk) bu renkler matrisinde hangi indise karşılık geliyorsa o sayı yazılır ve düzenleme-kontrol işlemi sayılar üzerinden gerçekleştirilir.

Sayı Matrisini Düzenleme :

Bi kısımda tahmin edileceği üzere rekürsif backtracking algoritması kullanılmıştır. İşlem yapılacak satırın indisi , matris ve her satırın kaç kez kaydırıldığını tutan dizi parametre olarak alınır. İlk satır önce değiştirilmeden satır indisi + 1 için rekürsif olarak fonksiyon tekrar çağırılır. Sonrasında o satırdaki her bir sütun için o değerin üst satırlarda olup olmadığı kontrol edilir herhangi bir şekilde eşleşme varsa ilgili satır sağa doğru bir kez shift edilir ve bu satırın numrasına karşılık gelen kaydırmaların tutulduğu dizideki matris 1 arttırılır , sonrasında sütunları gezdiğimiz indis sıfırlanır ve en baştan tekrar başlanır bu sayede 1 2 3 gibi bir matris için aşağıdaki satır bir kez kaydırılır,

2 1 3

Hala istenmeyen bir matris olduğu anlaşılır çünkü 2 1 3 satırı bir kez shift edilip 3 2 1 olunca bu sefer de 2 ler çakışacaktır bunun da kontrolü için sütun indisinin sıfırlanması gerekir. Shift edilme ya da hiç edilmeme olayı bittikten sonra bir sonraki satır için fonksiyon tekrar çağırılır ve aynı işlemler tekrarlanır. En aşağı satıra gelindiğinde yani 3X3 lük matriste 0 , 1 geçilip 2 de bittikten sonra satır indisi 3 olup fonksiyon çağırıldığında, fonksiyonun başındaki kontrolden geçemeyip geri döner böylece sonsuza kadar kendisini çağırmaya çalışmaz.

Fonksiyonlardan geri dönüş olduğunda kaydırmaların olduğu dizi kontrol edilir, örnek olarak 4.satırdan geri dönülüp 3.satıra gelindiğinde 4.satırın kaç kez kaydırıldığı kontrol edilir eğer N kez kaydırıldıysa herhangi bir kombinasyon bulunamamış ve eski haline geri dönmüştür bu durumda 3.satır 1 kez daha kaydırılıp 4.satır için fonksiyon tekrar çağırılır ancak bu işlem de sonsuza kadar çağırılmaz eğer bir satır N kez kaydırıldıysa mevcut fonksiyon sonlandırılır bir üst satıra geçilir ve alt satırın kaydırma sayısı sıfırlanır bu sayede 3.satır shift edilip tekrar 4.satır çağırıldığında kombinasyonlar tekrar denenir. Eğer en üst N kez shift edilirse bu , aşağıdaki satırların defalarca shift edilip hala kombinasyon bulunamamasından demektir. Bunun sonucunda program ilk çıktığı geri verir ve herhangi bir düzenleme mümkün değil şeklinde uyarı verir.

UYGULAMA

4X4 lük bir matris oluşturup renkleri verip basic mode da deniyoruz ve sıralanmış halini görüyoruz

```
Enter 1 to Form a Matrix :
Enter 9 to Exit :
1
Enter The Size of Board : 4
Enter the 1. Rows Colors
1. Color : sari
2. Color : yesil
3. Color : kirmizi
4. Color : mavi
Enter the 2. Rows Colors
1. Color : yesil
2. Color : mavi
3. Color : kirmizi
4. Color : sari
Enter the 3. Rows Colors
1. Color : sari
2. Color : mavi
3. Color : kirmizi
4. Color : yesil
Enter the 4. Rows Colors
1. Color : yesil
2. Color : sari
3. Color : kirmizi
4. Color : mavi
Enter 2 to Basic Mode :
Enter 3 to Detailed Mode :
Enter 8 to Form a New Matrix or Exit :
2

-----Before-----
sari yesil kirmizi mavi
yesil mavi kirmizi sari
sari mavi kirmizi yesil
yesil sari kirmizi mavi

-----After-----
sari yesil kirmizi mavi
mavi kirmizi sari yesil
yesil sari mavi kirmizi
kirmizi mavi yesil sari
```

Burada da detailed mode da düzenlenmiş matrisin her bir satırının kaç kez shift edildiğini görüyoruz

```
Enter 2 to Basic Mode :  
Enter 3 to Detailed Mode :  
Enter 8 to Form a New Matrix or Exit :  
3
```

```
-----Before-----
```

```
sari yesil kirmizi mavi  
mavi kirmizi sari yesil  
yesil sari mavi kirmizi  
kirmizi mavi yesil sari
```

```
-----After-----
```

```
sari yesil kirmizi mavi  
mavi kirmizi sari yesil  
yesil sari mavi kirmizi  
kirmizi mavi yesil sari  
1. Row Rotated 0 Times  
2. Row Rotated 3 Times  
3. Row Rotated 1 Times  
4. Row Rotated 2 Times
```

Aşağıda verilen örnekte ise verilen matrisin uygun bir kombinasyonunun bulunamadığı uyarısını ve aynı matrisi bize geri yazdığını görüyoruz ve dikkat edersek detailed modda çalıştırdığımız için kaç kez shift edildiğini de görüyoruz hepsinin N 'e eşit olması demek hepsinin yeteri kadar shift edildiği halde uygun kombinasyonu bulamadığını gösterir

```
Enter 8 to Form a New Matrix or Exit :
8
Enter 1 to Form a Matrix :
Enter 9 to Exit :
1
Enter The Size of Board : 3
Enter the 1. Rows Colors
1. Color : kirmizi
2. Color : mavi
3. Color : yesil
Enter the 2. Rows Colors
1. Color : yesil
2. Color : mavi
3. Color : kirmizi
Enter the 3. Rows Colors
1. Color : kirmizi
2. Color : yesil
3. Color : mavi
Enter 2 to Basic Mode :
Enter 3 to Detailed Mode :
Enter 8 to Form a New Matrix or Exit :
3

-----Before-----
kirmizi mavi yesil
yesil mavi kirmizi
kirmizi yesil mavi

-----After-----
There Is No Arrangement Available !
Your Entered Matrix Would Remain Unchanged
kirmizi mavi yesil
yesil mavi kirmizi
kirmizi yesil mavi
1. Row Rotated 3 Times
2. Row Rotated 3 Times
3. Row Rotated 3 Times
```

SONUÇ :

Zaman Karmaşıklığı :

Shifter fonksiyonu = $O(n)$ o satırın tamamını gezip elemanların yerlerini değiştiriyor

Kontrol etme bölümü = $O(N^2)$ her bir satırın ilgili sütunu için yukarıdaki tüm satırları gezer

Düzenleme fonksiyonu = $O(N^N)$ Her bir satır için kendisini çağırıyor

Dolayısıyla ana algoritmanın karmaşıklığı $O(N^N)$ dir

Yer Karmaşıklığı :

$N \times N$ lik bir matris, N lik shift counter dizisi , N lik renk dizisi yani totalde $O(N^2)$ yer karmaşıklığına sahiptir

Video Linkli :

<https://www.youtube.com/watch?v=NaRjdL3Yp9w>