



## Algoritma Analizi Ödev -4

Öğrenci Adı: Ömer Diner

Öğrenci Numarası: 20011017

Dersin Öğretmeni: M. Amaç Güvensan

Video Linki: [https://drive.google.com/file/d/1Dfu-cRUS0\\_2nl-bR-x6da7a\\_gMdJS0Ob/view?usp=drive\\_link](https://drive.google.com/file/d/1Dfu-cRUS0_2nl-bR-x6da7a_gMdJS0Ob/view?usp=drive_link)

## 1- Problem Tanımı:

Bu ödevde, n-Queen problemini çözmek için C dilinde bir algoritma yazılması istenmektedir. Amaç ,  $n \times n$  boyutundaki bir satranç tahtasına  $n$  adet vezirin birbirlerini alamayacak şekilde yerleştirildiği tüm çözümleri göstermektir. Vezirler, satranç tahtasında bulundukları satır, sütun ve çaprazlar üzerindeki diğer vezirleri tehdit ettiği için bu kontroller yapılmalıdır. Ödev kapsamında 5 farklı modlu bir program yazılması bekleniyor:

Brute Force Modu: Tüm olası konfigürasyonları deneyerek çözüm bulmaya çalışan yaklaşım.

OPTIMIZED\_1 Modu: Vezirlerin aynı satırda olamayacağı bilinerek optimize edilen yaklaşım.

OPTIMIZED\_2 Modu: Vezirlerin aynı satır ve aynı sütunda olamayacağı bilinerek optimize edilen bir yaklaşım.

Backtracking Modu: Backtracking algoritması ile daha verimli çözüm sunan bir yaklaşım.

Tüm yöntemler: Üstteki tüm modları çalıştırıp sonuçlarını gösteren mod.

## 2- Problemin Çözümü:

Tanımda anlatılan yöntemler kodlanmış ve bunların çalışma süreleri karşılaştırılmıştır:

1. Brute Force: Vezirler boarda olabilecek tüm şekillerde yerleştirilmiş ve kuralları sağlayıp sağlamadıkları kontrol edilmiştir. Çalışma süreleri uzun olmuş ve büyük  $N$  değerleri için verimli olmamıştır.

2.OPTIMIZED\_1: Bu modda, vezirlerin aynı satırda olamayacağı bilgisi kullanılarak yazılan bu modda her satıra bir vezir yerleştirme yöntemi izlenerek olası kombinasyonlar azaltılmıştır. Bu yöntem, brute force moduna göre daha hızlı çalışmış ve büyük  $N$  değerleri için daha uygun olmuştur.

3. OPTIMIZED\_2: Bu modda, vezirlerin aynı satırda ve aynı sütunda olamayacağı bilgisi kullanılarak bir optimizasyon yapılmıştır. Bir satıra vezir yerleştirildiğinde, aynı sütundaki diğer karelerin elenmesi sağlanmıştır.

OPTIMIZED\_1 moduna göre daha verimli bir çözüm sunmuş ve çalışma süresi iyileşmiştir.

4. BACKTRACKING Modu: Bu mod, backtracking algoritması kullanarak vezirleri tehdit oluşturmayacak şekilde yerleştirmeyi sağlamıştır. Backtracking yöntemi, her adımda olası konumları deneyerek uygun olmayan bir durumla karşılaşıldığında geri dönmeyi sağlar. Backtracking, diğer tüm modlardan daha hızlı çalışarak en hızlı çözümü sunmuştur.

5. Tüm Yaklaşımların Karşılaştırılması: Program, tüm modları ardışık olarak çalıştırarak her bir modun çalışma sürelerini ekrana yazdırmış ve karşılaştırma yapmıştır.

### **3- Karşılaşılan Sorunlar:**

Brute force modu büyük N değerleri için( $N \geq 8$ ) aşırı uzun sürelerde tamamlandı. Bu modun sadece küçük N değerleri için kullanılabilir olduğu görüldü.

#### 4. Çıktılar

N=4 değeri için backtracking

```
-----
1-Brute Force
2-Backtracking
3-Optimized 1
4-Optimized 2
5-Tum yontemler
0-Cikis
Seciminiz: 2
Cozum 1:
. . Q .
Q . . .
. . . Q
. Q . .

Cozum 2:
. Q . .
. . . Q
Q . . .
. . Q .

BACKTRACKING
Toplam cozum: 2
Calisma suresi: 0.016000 saniye
```

N=6 değeri için tüm modların tek tek denenmesi

```
N-Queens problemi icin tahta boyutunu giriniz: 6
1-Brute Force
2-Backtracking
3-Optimized 1
4-Optimized 2
5-Tum yontemler
0-Cikis
Seciminiz: 1
Cozum 1:
. Q . . . .
. . . Q . .
. . . . . Q
Q . . . . .
. . Q . . .
. . . . Q .

Cozum 2:
. . Q . . .
. . . . . Q
. Q . . . .
. . . . Q .
Q . . . . .
. . . Q . .

Cozum 3:
. . . Q . .
Q . . . . .
. . . . Q .
. Q . . . .
. . . . . Q
. . Q . . .

Cozum 4:
. . . . Q .
. . Q . . .
Q . . . . .
. . . . . Q
. . . Q . .
. Q . . . .

BRUTE FORCE
Toplam cozum: 4
Calisma suresi: 0.333000 saniye
```

Seciminiz: 2

Cozum 1:

```
. . . Q . .  
Q . . . . .  
. . . . Q .  
. Q . . . .  
. . . . . Q  
. . Q . . .
```

Cozum 2:

```
. . . . Q .  
. . Q . . .  
Q . . . . .  
. . . . . Q  
. . . Q . .  
. Q . . . .
```

Cozum 3:

```
. Q . . . .  
. . . Q . .  
. . . . . Q  
Q . . . . .  
. . Q . . .  
. . . . Q .
```

Cozum 4:

```
. . Q . . .  
. . . . . Q  
. Q . . . .  
. . . . Q .  
Q . . . . .  
. . . Q . .
```

BACKTRACKING

Toplam cozum: 4

Calisma suresi: 0.026000 saniye

Seciminiz: 3

Cozum 1:

```
. Q . . . .  
. . . Q . .  
. . . . Q  
Q . . . . .  
. . Q . . .  
. . . . Q .
```

Cozum 2:

```
. . Q . . .  
. . . . Q  
. Q . . . .  
. . . . Q .  
Q . . . . .  
. . . Q . .
```

Cozum 3:

```
. . . Q . .  
Q . . . . .  
. . . . Q .  
. Q . . . .  
. . . . Q  
. . Q . . .
```

Cozum 4:

```
. . . . Q .  
. . Q . . .  
Q . . . . .  
. . . . Q  
. . . Q . .  
. Q . . . .
```

OPTIMIZED 1

Toplam cozum: 4

Calisma suresi: 0.023000 saniye

Seciminiz: 4

Cozum 1:

```
. Q . . . .  
. . . Q . .  
. . . . . Q  
Q . . . . .  
. . Q . . .  
. . . . Q .
```

Cozum 2:

```
. . Q . . .  
. . . . . Q  
. Q . . . .  
. . . . Q .  
Q . . . . .  
. . . Q . .
```

Cozum 3:

```
. . . Q . .  
Q . . . . .  
. . . . Q .  
. Q . . . .  
. . . . . Q  
. . Q . . .
```

Cozum 4:

```
. . . . Q .  
. . Q . . .  
Q . . . . .  
. . . . . Q  
. . . Q . .  
. Q . . . .
```

OPTIMIZED 2

Toplam cozum: 4

Calisma suresi: 0.023000 saniye



```
Brute Force:  
Toplam cozum: 4  
Calisma suresi: 0.294000 saniye
```

```
-----  
Optimized 1:  
Toplam cozum: 4  
Calisma s³resi: 0.018000 saniye
```

```
-----  
Optimized 2:  
Toplam cozum: 4  
Calisma suresi: 0.009000 saniye
```

```
-----  
Backtracking:  
Toplam cozum: 4  
Calisma suresi: 0.009000 saniye  
-----
```

N=8 değeri için tüm modlar

```
Brute Force:  
Toplam cozum: 92  
Calisma suresi: 974.498000 saniye
```

```
-----  
Optimized 1:  
Toplam cozum: 92  
Calisma s³resi: 0.585000 saniye
```

```
-----  
Optimized 2:  
Toplam cozum: 92  
Calisma suresi: 0.305000 saniye
```

```
-----  
Backtracking:  
Toplam cozum: 92  
Calisma suresi: 0.376000 saniye
```