

# گزارش کار الگوریتم تپه نوردی برای مسئله 8 وزیر

---

نجمه نظری 99156006

امیرحسین حسنی 99155004

محمد اسماعیلی 99155001

- 
- ابتدا یک objective function تعریف میکنیم که وظیفه آن محاسبه تعداد وزیر هایی است که در یک جواب با هم برخورد ندارند
  - برای مثال در حالتی که ما 8 وزیر داشته باشیم این تابع یک لیست همچون  $a=[1,3,6,2,0,4,5,7]$  دریافت میکند که هر یک از اعداد شماره ستونی هستند که وزیر در آن قرار دارد همینطور index این اعداد نشان دهنده سطری که وزیر در آن است میباشد
  - حال تابع hill\_climbing\_for\_nqueen را تعریف میکنیم که از ما  $n$  که تعداد وزیر ها و همینطور نشان دهنده صفحه بازی است را دریافت میکند



---

- در تابع `hill_climbing_for_nqueen` یک جواب رندوم همچون لیست `a` تولید میشود و سپس یک ماتریکس به شکل صفحه بازی به همراه وزیر های درون بازی ساخته و این حالت ابتدایی به همراه وزیر هایی که یکدیگر را تحدید نمیکنند پرینت میشود

- سپس وارد حلقه ی مسئله میشویم که در آن 3 همسایه با تغییراتی کم در جواب ابتدایی به شکل زیر ساخته میشوند

```
rand_num1 = np.random.randint(0,high=n)
rand_num2 = np.random.randint(0,high=n)
rand_num3 = np.random.randint(0,high=n)
```

```
s1[np.random.randint(0,high=n)] = np.random.randint(0,high=n)
```

```
s2[np.random.randint(0,high=n)] = np.random.randint(0,high=n)
s2[np.random.randint(0,high=n)] = np.random.randint(0,high=n)
```

```
s3[np.random.randint(0,high=n)] = np.random.randint(0,high=n)
s3[np.random.randint(0,high=n)] = np.random.randint(0,high=n)
s3[np.random.randint(0,high=n)] = np.random.randint(0,high=n)
```

- میبینیم که برای همسایه اول 1 برای دوم 2 و برای

- همسایه سوم 3 وزیر نسبت به جواب اولیه به

- صورت رندوم جابجا میشوند

- 
- حال با استفاده از شرط حلقه و تابع objective و ماکزیمم گیری بین جواب ها best جدید ما به دست می آید و این حلقه تا زمانی که جواب ما بهتر یا مثل همسایه های خود شود ادامه پیدا میکند. در آخر حالت نهایی به همراه وزیر هایی که یکدیگر را تحدید نمیکند چاپ میشوند.



```
hill_climbing_for_nqueen(8)
```

```
starting game state=
```

```
[[0. 0. 0. 0. 0. 8. 0. 0.]  
 [0. 0. 0. 0. 8. 0. 0. 0.]  
 [0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 8.]  
 [0. 0. 0. 0. 8. 0. 0. 0.]  
 [0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 8. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 8. 0. 0.]]  
queens in peace = 1
```

```
Final game state =
```

```
[[0. 0. 0. 0. 8. 0. 0. 0.]  
 [0. 0. 8. 0. 0. 0. 0. 0.]  
 [0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 8.]  
 [0. 0. 0. 0. 0. 8. 0. 0.]  
 [0. 0. 8. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 8. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 8. 0. 0.]]  
queens in peace = 2
```

```
hill_climbing_for_nqueen(9)
```

```
starting game state=
```

```
[[0. 0. 0. 0. 8. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 8. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 0. 8.]  
 [0. 0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 8. 0. 0.]  
 [0. 0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 8. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 8. 0.]]  
queens in peace = 2
```

```
Final game state =
```

```
[[0. 0. 0. 0. 0. 0. 8. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 8. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 0. 8.]  
 [0. 0. 0. 0. 0. 0. 0. 8. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 0. 8.]  
 [0. 0. 0. 0. 0. 0. 8. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 8. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 0. 8.]  
 [0. 0. 0. 0. 0. 8. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 0. 8.]]  
queens in peace = 2
```

```
hill_climbing_for_nqueen(9)
```

```
starting game state=
```

```
[[0. 8. 0. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 8. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 8. 0. 0. 0.]  
 [0. 0. 0. 0. 8. 0. 0. 0. 0.]  
 [0. 0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 8. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 8. 0. 0. 0.]]  
queens in peace = 0
```

```
Final game state =
```

```
[[8. 0. 0. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 8. 0. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 8. 0. 0.]  
 [0. 0. 0. 0. 0. 8. 0. 0. 0.]  
 [0. 0. 0. 0. 8. 0. 0. 0. 0.]  
 [0. 0. 8. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 8. 0. 0. 0. 0.]  
 [0. 8. 0. 0. 0. 0. 0. 0. 0.]]  
queens in peace = 3
```