مبینا نوری – ۹۶۱۵۰۹۱۰۶۳

در این مساله حالت کلی ۸ وزیر یعنی n وزیر را پیاده سازی کردیم.

ابتدا یک کلاس کروموزوم با توابع زیر درست کردیم.

```
class Chromosome:
    def __init__(self, gens):…

    def calculate_fitness(self):…

    def print_chromosome(self):…

    def show(self):…
```

در هنگام ساخت کروموزم fitness آن محاسبه میگردد.

```
def __init__(self, gens):
    self.n = len(gens)
    self.gens = gens
    self.fitness = self.calculate_fitness()
```

در تابع calculate_fitness تعداد جفت وزیر هایی که هم را تهدید می کنند را میشماریم (جواب مسئله fitness برابر با ۰ است)

برای اینکار دو به دو مکان وزیر ها را چک میکنیم. شرط تهدید نکردن این است که شیب خط بینشان 1 یا -۱ یا ۰ نباشد.

```
9       def calculate_fitness(self):
10          fitness = 0
11
12          collision_shibs = [-1, 0, 1]
13
14          for i in range(self.n):
15              for j in range(i + 1, self.n):
16                  shib = (self.gens[i] - self.gens[j]) / (i - j)
17                  if shib in collision_shibs:
18                      fitness += 1
19
20          return fitness
21
```

حال برای شروع الگوریتم یک جمعیت اولیه درست میکنیم. به تعداد ۱۰۰۰ کروموزوم. سپس آنها را بر حسب عدد fitness مرتب میکنیم.

```
73      population = [create_random_chromosome(n) for _ in range(1000)]
74      population.sort(key=lambda chrom : chrom.fitness)
75
```

در تابع create_random_chromosome به ازای هر ستون یک عدد رندم بین ۱ تا n قرار میدهیم. و با آن یک کروموزوم ساخته و برمیگردانیم.

```
39  def create_random_chromosome(n) -> Chromosome:
40      return Chromosome([random.randint(1, n) for _ in range(n)])
```

حال در main پس از ساخت جمعیت اولیه، یک حلقه اجرا میکنیم که تا زمان پیدا نشدن جواب مسئله نسل را عوض کند.

```
78      while not 0 in [chrom.fitness for chrom in population]:
79          print("=== Generation {} ===".format(generation))
80          population = genetic_queen(population)
81          print("")
82          print("Minimum Fitness:", min([chrom.fitness for chrom in population]))
83          print("Maximum Fitness:", max([chrom.fitness for chrom in population]))
84          generation += 1
85
```

تابع genetic_queen یک نسل گرفته (آرایه ای از کروموزوم) و نسل بعد را بر روی همان تولید میکند.

```
53  def genetic_queen(population):
54      mutation_probability = 0.3
55
56      for _ in range(len(population)):
57          x = random.choice(population)
58          y = random.choice(population)
59          child = cross_over(x, y)
60          if random.random() < mutation_probability:
61              child = mutate(child)
62          population.append(child)
63          if child.fitness == 0:
64              break
65
66      random.shuffle(population)
67      population.sort(key=lambda chrom : chrom.fitness)
68      population = population[0 : 1000]
69      return population
```

در این تابع به تعداد جمعیت(که همیشه ۱۰۰۰ است) فرزند تولید میشود. به این صورت که از دو والد رندوم یک فرزند تولید کرده و به جمعیت اضافه میکنیم. و به احتمال ۰.۳ فرزند را جهش می دهیم.

در انتها اعضای جمعیت را بر حسب fitness مرتب کرده و ۱۰۰۰ عضو برتر را به عنوان جمعیت جدید قرار میدهیم. با این کار به مرور زمان کروموزوم های برتر بیشتر میشوند تا به جواب برسیم.

توابع فرزند آوری و جهش :

```python
def cross_over(first_chrom : Chromosome, second_chrom : Chromosome) -> Chromosome:
    n = first_chrom.n
    ind = random.randint(0, n - 1)
    return Chromosome(first_chrom.gens[0 : ind] + second_chrom.gens[ind : n])

def mutate(chrom : Chromosome) -> Chromosome:
    n = chrom.n
    ind = random.randint(0, n - 1)
    new_val = random.randint(1, n)
    return Chromosome(chrom.gens[0 : ind] + [new_val] + chrom.gens[ind + 1 : n])

```

در تابع cross_over یک اندیس رندوم گرفته و برای فرزند تکه سمت راست کروموزوم را از والد اول و تکه سمت راست را از والد دوم تولید میکنیم.

در تابع mutate ژن مربوط به یک اندیس رندوم را برابر با یک عدد رندوم بین ۱ تا n قرار میدهیم.

نمونه اجرا :

حالت اول برای 8 وزیر. زمان اجرا : کمتر از ۱ ثانیه. یافت جواب در نسل ۱۵

```
Enter Number of Queens: 8
=== Generation 1 ===

Minimum Fitness: 2
Maximum Fitness: 8
=== Generation 2 ===

Minimum Fitness: 2
Maximum Fitness: 6
=== Generation 3 ===

Minimum Fitness: 2
Maximum Fitness: 6
=== Generation 4 ===

Minimum Fitness: 2
Maximum Fitness: 5
=== Generation 5 ===

Minimum Fitness: 1
Maximum Fitness: 5
=== Generation 6 ===

Minimum Fitness: 1
Maximum Fitness: 4
=== Generation 7 ===

Minimum Fitness: 1
Maximum Fitness: 4
=== Generation 8 ===

Minimum Fitness: 1
```

```
Maximum Fitness: 4
=== Generation 9 ===

Minimum Fitness: 1
Maximum Fitness: 4
=== Generation 10 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 11 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 12 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 13 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 14 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 15 ===

Minimum Fitness: 0
Maximum Fitness: 3
!Solved in Generation 15
Chromosome = [7, 4, 2, 8, 6, 1, 3, 5],  Fitness = 0


_____________________________
XXXXXXQX
XXXQXXXX
XQXXXXXX
XXXXXXXQ
XXXXXQXX
QXXXXXXX
XXQXXXXX
XXXXQXXX
----------------------------
```

حالت دوم : ۱۴ وزیر. زمان اجرا : 5 ثانیه. یافت جواب در نسل 94

```
=== Enter Number of Queens: === Generation 1
```

```
Minimum Fitness: 5
Maximum Fitness: 15
=== Generation 2 ===

Minimum Fitness: 5
Maximum Fitness: 13
=== Generation 3 ===

Minimum Fitness: 5
Maximum Fitness: 11
=== Generation 4 ===

Minimum Fitness: 4
Maximum Fitness: 11
=== Generation 5 ===

Minimum Fitness: 4
Maximum Fitness: 10
=== Generation 6 ===

Minimum Fitness: 4
Maximum Fitness: 9
=== Generation 7 ===

Minimum Fitness: 4
Maximum Fitness: 9
=== Generation 8 ===

Minimum Fitness: 3
Maximum Fitness: 9
=== Generation 9 ===

Minimum Fitness: 2
Maximum Fitness: 8
=== Generation 10 ===

Minimum Fitness: 2
Maximum Fitness: 8
=== Generation 11 ===

Minimum Fitness: 2
```

```
Maximum Fitness: 8
=== Generation 12 ===

Minimum Fitness: 2
Maximum Fitness: 7
=== Generation 13 ===

Minimum Fitness: 2
Maximum Fitness: 7
=== Generation 14 ===

Minimum Fitness: 2
Maximum Fitness: 7
=== Generation 15 ===

Minimum Fitness: 2
Maximum Fitness: 7
=== Generation 16 ===

Minimum Fitness: 2
Maximum Fitness: 7
=== Generation 17 ===

Minimum Fitness: 2
Maximum Fitness: 6
=== Generation 18 ===

Minimum Fitness: 2
Maximum Fitness: 6
=== Generation 19 ===

Minimum Fitness: 2
Maximum Fitness: 6
=== Generation 20 ===

Minimum Fitness: 2
Maximum Fitness: 6
=== Generation 21 ===

Minimum Fitness: 2
Maximum Fitness: 6
=== Generation 22 ===
```

```
Minimum Fitness: 2
Maximum Fitness: 6
=== Generation 23 ===

Minimum Fitness: 2
Maximum Fitness: 6
=== Generation 24 ===

Minimum Fitness: 2
Maximum Fitness: 6
=== Generation 25 ===

Minimum Fitness: 2
Maximum Fitness: 5
=== Generation 26 ===

Minimum Fitness: 2
Maximum Fitness: 5
=== Generation 27 ===

Minimum Fitness: 2
Maximum Fitness: 5
=== Generation 28 ===

Minimum Fitness: 2
Maximum Fitness: 5
=== Generation 29 ===

Minimum Fitness: 2
Maximum Fitness: 5
=== Generation 30 ===

Minimum Fitness: 2
Maximum Fitness: 5
=== Generation 31 ===

Minimum Fitness: 2
Maximum Fitness: 5
=== Generation 32 ===

Minimum Fitness: 2
```

```
Maximum Fitness: 5
=== Generation 33 ===

Minimum Fitness: 2
Maximum Fitness: 5
=== Generation 34 ===

Minimum Fitness: 2
Maximum Fitness: 5
=== Generation 35 ===

Minimum Fitness: 2
Maximum Fitness: 4
=== Generation 36 ===

Minimum Fitness: 2
Maximum Fitness: 4
=== Generation 37 ===

Minimum Fitness: 2
Maximum Fitness: 4
=== Generation 38 ===

Minimum Fitness: 2
Maximum Fitness: 4
=== Generation 39 ===

Minimum Fitness: 2
Maximum Fitness: 4
=== Generation 40 ===

Minimum Fitness: 1
Maximum Fitness: 4
=== Generation 41 ===

Minimum Fitness: 1
Maximum Fitness: 4
=== Generation 42 ===

Minimum Fitness: 1
Maximum Fitness: 4
=== Generation 43 ===
```

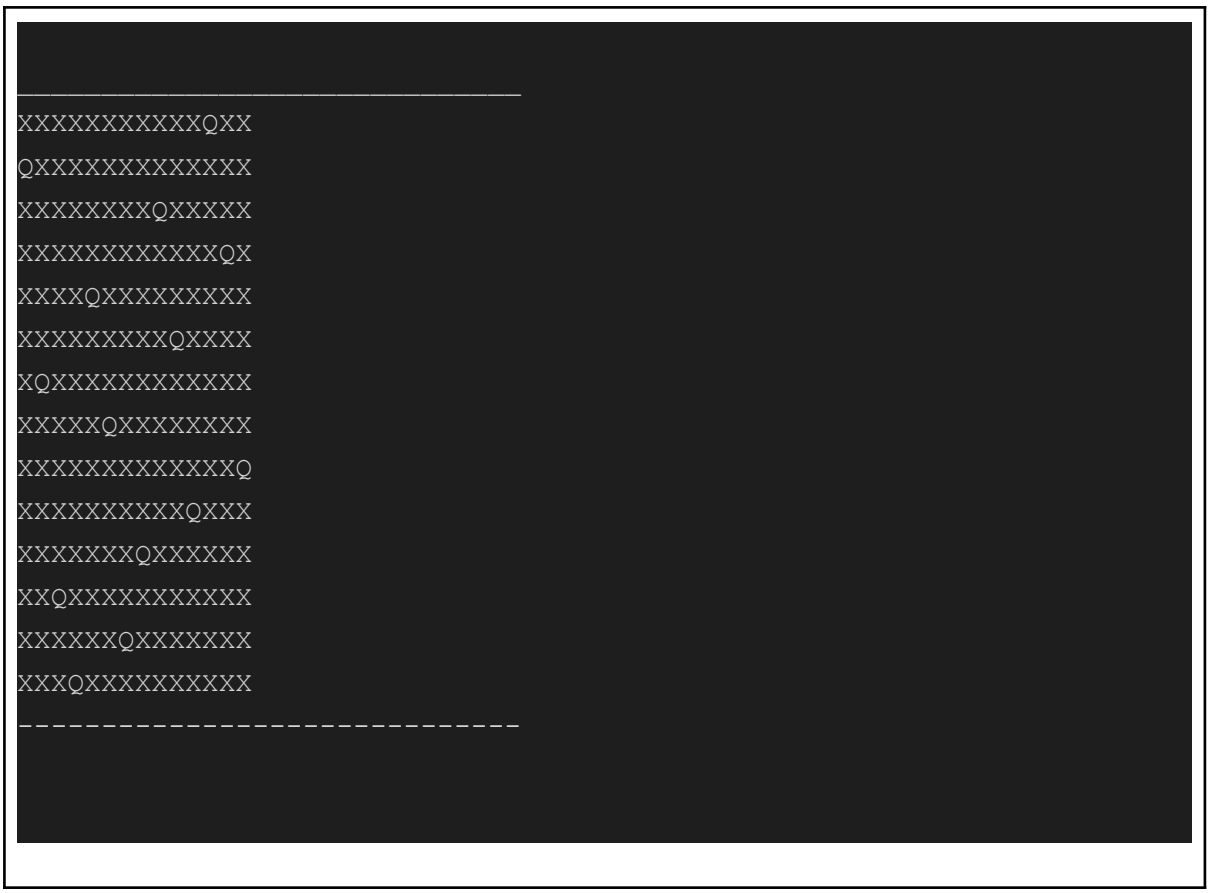
```
Minimum Fitness: 1
Maximum Fitness: 4
=== Generation 44 ===

Minimum Fitness: 1
Maximum Fitness: 4
=== Generation 45 ===

Minimum Fitness: 1
Maximum Fitness: 4
=== Generation 46 ===

Minimum Fitness: 1
Maximum Fitness: 4
=== Generation 47 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 48 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 49 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 50 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 51 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 52 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 53 ===

Minimum Fitness: 1
```

```
Maximum Fitness: 3
=== Generation 54 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 55 ===

Minimum Fitness: 1
Maximum Fitness: 3
=== Generation 56 ===

Minimum Fitness: 1
Maximum Fitness: 2
=== Generation 57 ===

Minimum Fitness: 1
Maximum Fitness: 2
=== Generation 58 ===

Minimum Fitness: 1
Maximum Fitness: 2
=== Generation 59 ===

Minimum Fitness: 1
Maximum Fitness: 2
=== Generation 60 ===

Minimum Fitness: 1
Maximum Fitness: 2
=== Generation 61 ===

Minimum Fitness: 1
Maximum Fitness: 2
=== Generation 62 ===

Minimum Fitness: 1
Maximum Fitness: 2
=== Generation 63 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 64 ===
```

```
Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 65 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 66 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 67 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 68 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 69 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 70 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 71 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 72 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 73 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 74 ===

Minimum Fitness: 1
```

```
Maximum Fitness: 1
=== Generation 75 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 76 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 77 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 78 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 79 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 80 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 81 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 82 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 83 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 84 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 85 ===
```

```
Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 86 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 87 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 88 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 89 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 90 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 91 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 92 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 93 ===

Minimum Fitness: 1
Maximum Fitness: 1
=== Generation 94 ===

Minimum Fitness: 0
Maximum Fitness: 1
!Solved in Generation 94
Chromosome = [12, 1, 9, 13, 5, 10, 2, 6, 14, 11, 8, 3, 7, 4],
Fitness = 0
```


_____
xxxxxxxxxxxQxx
Qxxxxxxxxxxxxx
xxxxxxxxQxxxxx
xxxxxxxxxxxxQx
xxxxQxxxxxxxxx
xxxxxxxxxQxxxx
xQxxxxxxxxxxxx
xxxxxQxxxxxxxx
xxxxxxxxxxxxxQ
xxxxxxxxxxQxxx
xxxxxxxQxxxxxx
xxQxxxxxxxxxxx
xxxxxxQxxxxxxx
xxxQxxxxxxxxxx
_____