

## Вештачка интелигенција-2021/2022/L

Dashboard / My courses / Ви-2021/2022/L / Самостојна проверка на знаење / Тест 3

## Quiz navigation



Finish attempt ...

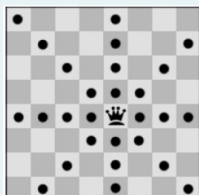
## Question 1

Correct

Mark 5.00 out of 5.00

Flag question

Дадена е  $n \times n$  табла за шах. Треба да се постават  $n$  кралици на таблата така што ниедна кралица да не се напаѓа. Кралиците може да се постават на било која позиција која сметаме дека е најсоодветна. Единственото ограничување е дека не треба да се напаѓаат. Може да се придвижуваат на кралицата ви се дадени на сликата подолу:



На влез се прима бројот на кралици и димензии на таблата  $n$ . На излез треба да се испечатат бројот на уникатни позиции на кои може да ги поставите кралиците со default Backtracking Solver ако бројот на кралици е помал или еднаков на 6. Во спротивно да се испечат само првото решение.

**Потсетник:** Во дадениот модул constraint веќе се имплементирани следните ограничувања како класи: *AllDifferentConstraint*, *AllEqualConstraint*, *MaxSumConstraint*, *ExactSumConstraint*, *MinSumConstraint*, *InSetConstraint*, *NotInSetConstraint*, *SomeInSetConstraint*, *SomeNotInSetConstraint*.

For example:

Input	Result
7	{1: (6, 6), 2: (5, 4), 3: (4, 2), 4: (3, 0), 5: (2, 5), 6: (1, 3), 7: (0, 1)}

Answer: (penalty regime: 0 %)

Reset answer

```
1 from constraint import *
2
3 if __name__ == '__main__':
4     problem = Problem(BacktrackingSolver())
5     num_of_queens = int(input())
6
7     variables = range(1, num_of_queens + 1)
8     domain = [(i, j) for i in range(num_of_queens) for j in range(num_of_queens)]
9
10    for variable in variables:
11        problem.addVariable(variable, domain)
12
13    # ---Tuka dodadete gi ogranichuvanjata-----
14
15    for queen1 in variables:
16        for queen2 in variables[queen1:]:
17            # No two queens are allowed to be on the same line, column or diagonal:
18            problem.addConstraint(
19                lambda q1, q2: q1[0] != q2[0] and q1[1] != q2[1] and abs(q1[0] - q2[0]) != abs(q1[1] - q2[1]),
20                (queen1, queen2))
21
22    # -----
23
24    if num_of_queens <= 6:
25        print(len(problem.getSolutions()))
26    else:
27        print(problem.getSolution())
28
```

Check

	Input	Expected	Got
✓	7	{1: (6, 6), 2: (5, 4), 3: (4, 2), 4: (3, 0), 5: (2, 5), 6: (1, 3), 7: (0, 1)}	{1: (6, 6), 2: (5, 4), 3: (4, 2), 4: (3, 0), 5: (2, 5), 6: (1, 3), 7: (0, 1)}

Passed all tests! ✓

Correct

Marks for this submission: 5.00/5.00.

← Тест 2

Jump to...

Класична лабораториска вежба 1 →

Next page

You are logged in as Велчевски Атанас (Log out)

Ви-2021/2022/L

Data retention summary

