

( / )



Curriculum

**Short Specializations** ^

Average: 97.3% v

# 0x05. N Queens

Algorithm

Python

By: Alexa Orrico, Software Engineer at Holberton School

Weight: 1

Project over - took place from Jan 8, 2024 6:00 AM to Jan 12, 2024 6:00 AM

☒ An auto review will be launched at the deadline

## In a nutshell...

- **Auto QA review:** 0.0/15 mandatory
- **Altogether: 0.0%**
  - Mandatory: 0.0%
  - Optional: no optional tasks

The "0x05. N queens" project is a classic problem in computer science and mathematics, known for its application of the backtracking algorithm to place N non-attacking queens on an N×N chessboard. To successfully complete this project, you will need to understand several key concepts and have access to resources that will help you grasp the necessary algorithms and techniques.

## Concepts Needed:

### 1. Backtracking Algorithms:

- Understanding how backtracking algorithms work to explore all potential solutions to a problem and backtrack when a solution cannot be completed.
- Backtracking Introduction (/rltoken/Pj4a2hp-rsGS6h83joZw7w)

### 2. Recursion:

- Using recursive functions to implement backtracking algorithms.
- Recursion in Python (/rltoken/X1vaNXgy\_pPyvKfOJm90XQ)

### 3. List Manipulations in Python:

- Creating and manipulating lists, especially to store the positions of queens on the board.



- Python Lists (/rltoken/P3KbYxmdtSeoJvVfr9lv0w)

#### (/)4. Python Command Line Arguments:

- Handling command-line arguments with the `sys` module.
- Command Line Arguments in Python (/rltoken/2IF4V6xsY\_Nq-xcGDK3Bhw)

By studying these concepts and utilizing the resources provided, you will be equipped with the knowledge required to implement an efficient solution to the N queens problem using Python. This project not only tests programming and problem-solving skills but also offers an excellent opportunity to learn about algorithmic thinking and optimization techniques.

## Additional Resources

- Mock Interview (/rltoken/0eUZmAkrTsQUmOh7\_KOuBg)

## Requirements

### General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted/compiled on Ubuntu 20.04 LTS using `python3` (version 3.4.3)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/python3`
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should use the `PEP 8` style (version 1.7.\*)
- All your files must be executable

## Tasks

### 0. N queens

**mandatory**

Score: 0.0% (Checks completed: 0.0%)



( / )



Chess grandmaster Judit Polgár (/rltoken/fZ1ecpPEmVL9nvkBn8WQGg), the strongest female chess player of all time

The N queens puzzle is the challenge of placing N non-attacking queens on an  $N \times N$  chessboard. Write a program that solves the N queens problem.

- Usage: nqueens N
  - If the user called the program with the wrong number of arguments, print Usage: nqueens N , followed by a new line, and exit with the status 1
- where N must be an integer greater or equal to 4
  - If N is not an integer, print N must be a number , followed by a new line, and exit with the status 1
  - If N is smaller than 4 , print N must be at least 4 , followed by a new line, and exit with the status 1
- The program should print every possible solution to the problem
  - One solution per line
  - Format: see example
  - You don't have to print the solutions in a specific order
- You are only allowed to import the sys module

Read: Queen (/rltoken/ghWql1wvx6g-UI7nrufMKA), Backtracking (/rltoken/-hgZbgRFkwmxaKnLnCluEQ)

```
julien@ubuntu:~/0x08. N Queens$ ./0-nqueens.py 4
[[0, 1], [1, 3], [2, 0], [3, 2]]
[[0, 2], [1, 0], [2, 3], [3, 1]]
julien@ubuntu:~/0x08. N Queens$ ./0-nqueens.py 6
[[0, 1], [1, 3], [2, 5], [3, 0], [4, 2], [5, 4]]
[[0, 2], [1, 5], [2, 1], [3, 4], [4, 0], [5, 3]]
[[0, 3], [1, 0], [2, 4], [3, 1], [4, 5], [5, 2]]
[[0, 4], [1, 2], [2, 0], [3, 5], [4, 3], [5, 1]]
julien@ubuntu:~/0x08. N Queens$
```



(/)  
**Repo:**

- GitHub repository: alx-interview
- Directory: 0x05-nqueens
- File: 0-nqueens.py

☐ Done?

Help

Check your code

Ask for a new correction

QA Review

Copyright © 2024 ALX, All rights reserved.

