



СЕМИНАР ДЛЯ
УЧИТЕЛЕЙ
ИНФОРМАТИКИ

Декабря 01
2018

Ведущий:
В. Карабанов А.
методист И

РЕКУРСИВНЫЙ
ПЕРЕБОР
С ВОЗВРАТОМ

ЗАДАЧА № 93. МИРНЫЕ ФЕРЗИ

Дано число N . Определите, сколькими способами можно расставить на доске $N \times N$ N ферзей, не бьющих друг друга.

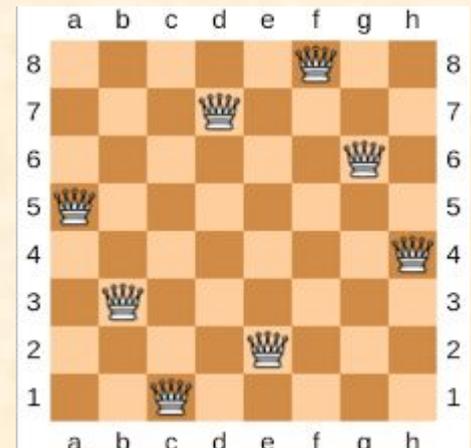
Входные данные: задано единственное число ($N \leq 10$)

Выходные данные: вывести количество способов, которыми можно расставить на доске $N \times N$ N ферзей, не бьющих друг друга.

Пример:

входные данные: 8

выходные данные: 92

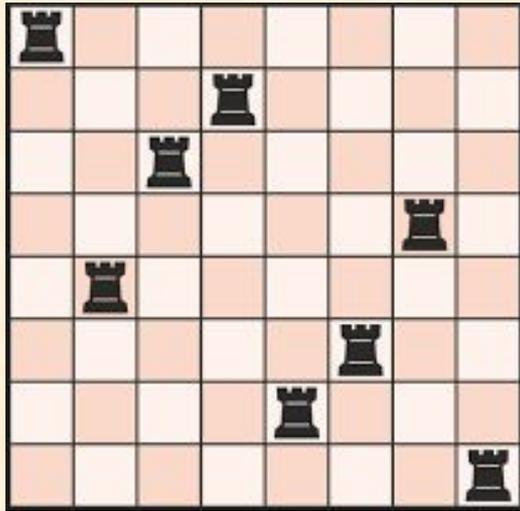


КЛАССИЧЕСКИЙ СЛУЧАЙ: 8×8

Общее число возможных расположений 8 ферзей на 64-клеточной доске равно $4\,426\,165\,368 = (64!/(8!(64-8)!))$.

Очевидно, что на одной горизонтали или вертикали доски не может находиться больше одного ферзя, поэтому алгоритм решения изначально не должен включать в перебор позиции, где два ферзя стоят на одной горизонтали или вертикали. Даже такое простое правило способно существенно уменьшить число возможных расположений: $16\,777\,216$ (то есть 8^8).

КЛАССИЧЕСКИЙ СЛУЧАЙ: 8×8



Генерируя перестановки, которые являются решениями задачи о восьми ладьях и затем проверяя атаки по диагоналям, можно сократить число возможных расположений всего до 40 320 (то есть 8!).

КЛАССИЧЕСКИЙ СЛУЧАЙ: 8×8

Один из типовых алгоритмов решения задачи — использование поиска с возвратом: первый ферзь ставится на первую горизонталь, затем каждый следующий пытаются поставить на следующую так, чтобы его не били ранее установленные ферзи. Если на очередном этапе постановки свободных полей не оказывается, происходит возврат на шаг назад — переставляется ранее установленный ферзь.

ПОДЗАДАЧА:

Даны координаты двух ферзей.
Определите, бьют ли они друг дру

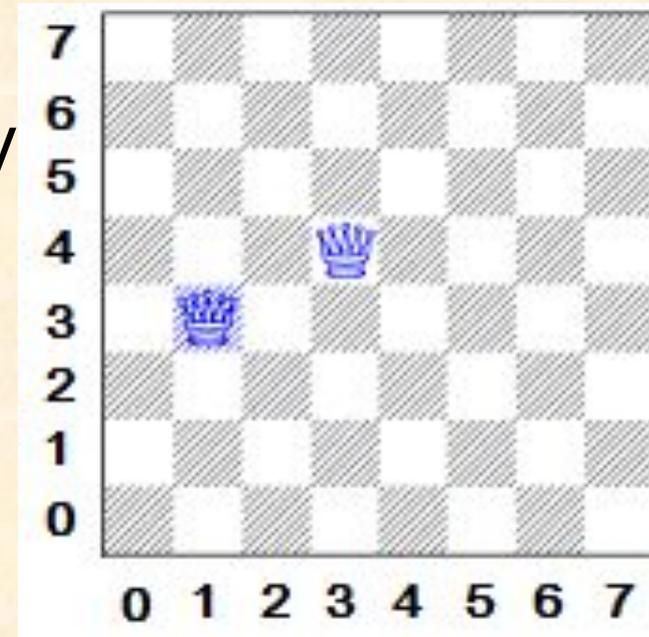
Входные данные: четыре числа.

Выходные данные: Yes или No.

Пример:

входные данные: 1 3 3 4

выходные данные: No



МАТЕМАТИЧЕСКАЯ МОДЕЛЬ:

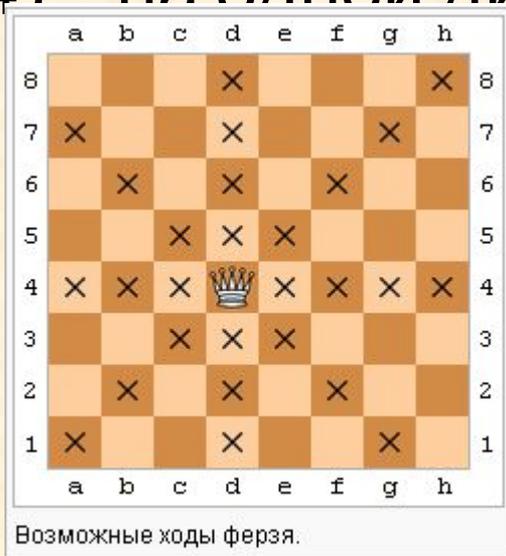
Ферзи бьют друг друга, если они находятся:

- 1) на одной горизонтали;
- 2) на одной вертикали;
- 3) на одной диагонали «юго-запад – северо-восток»;

- 4) на одной диагонали «юго-восток – северо-запад»;

Проверка:

- 1) $y_1 = y_2$;
- 2) $x_1 = x_2$;
- 3) $y_1 - x_1 = y_2 - x_2$;
- 4) $y_1 + x_1 = y_2 + x_2$.



РЕАЛИЗАЦИЯ (PYTHON):

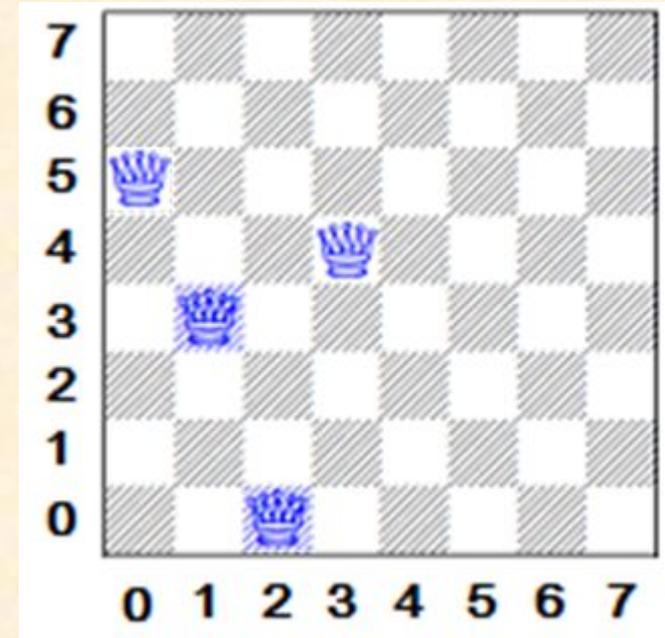
```
def check(x1, y1, x2, y2):  
    if y1 == y2 or x1 == x2 or  
       y1 - x1 == y2 - x2 or  
       y1 + x1 == y2 + x2:  
        return True  
    return False
```

РЕКУРСИВНАЯ ФУНКЦИЯ.

Необходимо хранить текущую (промежуточную) «хорошую» расстановку части ферзей.

Как это сделать?

- 1) завести двумерный массив;
- 2) достаточно одномерного массива!



[5, 3, 0, 4]

РЕКУРСИВНАЯ ФУНКЦИЯ.

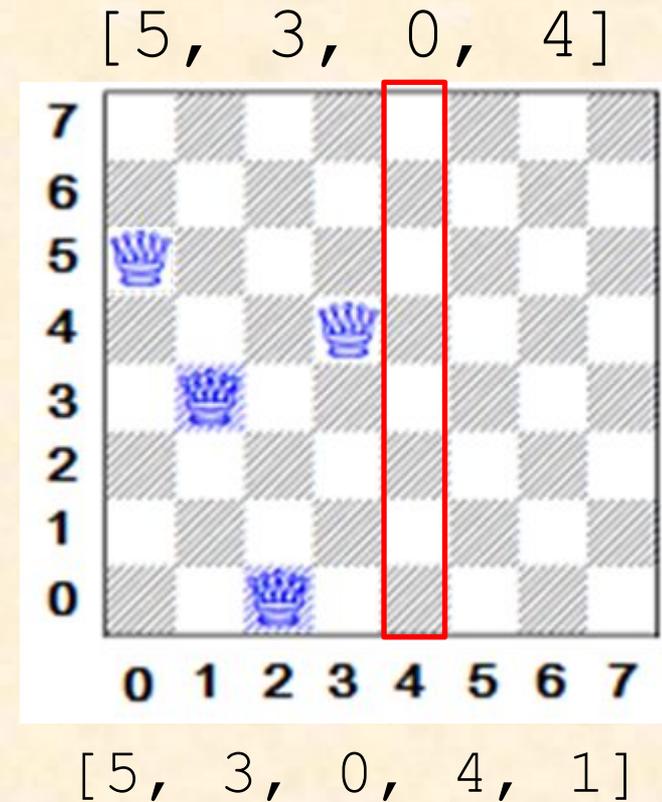
Будем рекурсивно передавать в функцию текущее положение ферзей.

Крайний случай?

Длина массива равна N.

Значит найдено еще одно решение (нужно увеличить счетчик на 1).

В противном случае – будем перебирать все возможные положения «нового» ферзя, и, если он не бьет всех «старых» ферзей – добавлять его положение в массив и передавать снова в



РЕАЛИЗАЦИЯ РЕКУРСИВНОЙ ФУНКЦИИ:

```
def rec(prefix):  
    global count  
    if len(prefix) == n: } # Крайний случай  
        count += 1  
        return  
    for i in range(n):  
        f = True # флажок  
        for j in range(len(prefix)):  
            if check(len(prefix), i, j, prefix[j]):  
                f = False  
                break  
        if f:  
            rec(prefix + [i])
```

ОСНОВНАЯ ПРОГРАММА:

```
n = int(input())
```

```
count = 0
```

```
rec([])
```

```
print(count)
```

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ:

- 1) добавить визуализацию решений;
- 2) заработать миллион долларов:

Компьютерные программы перестают справляться с решением задачи, когда размер доски доходит до 1000 на 1000 клеток. Тот, кто сможет написать программу, способную решить задачу быстро, сможет адаптировать ее и для решения других важных проблем: «К ним относятся поиск самой большой группы друзей в Facebook, которые не знакомы друг с другом, или взлом кода, который защищает все наши операции в Интернете». Решение задачи о восьми ферзях эквивалентно решению одной из так называемых задач тысячелетия, а именно задачи о тождестве классов сложности P и NP. За решение именно этой задачи Институт Клэя объявил награду в миллион долларов.

ЗАДАЧА № 94. МИРНЫЕ ФЕРЗИ (БЕЗ ПОВОРОТОВ И ОТРАЖЕНИЙ)

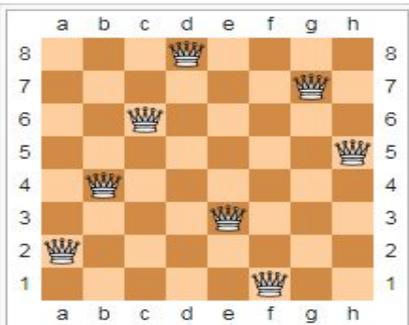
Дано число N . Определите, сколькими способами можно расставить на доске $N \times N$ N ферзей, не бьющих друг друга. Расстановки ферзей, которые можно получить друг из друга поворотами и отражениями доски, нужно считать за одно.

Входные данные: задано единственное число ($N \leq 10$)

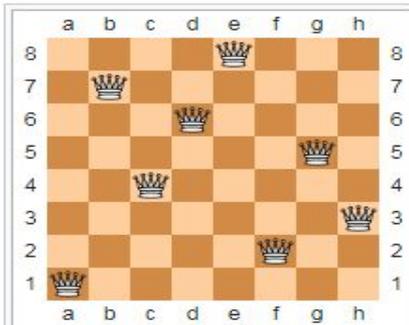
Выходные данные: вывести количество способов, которыми можно расставить на доске $N \times N$ N ферзей, не бьющих друг друга.

Пример:

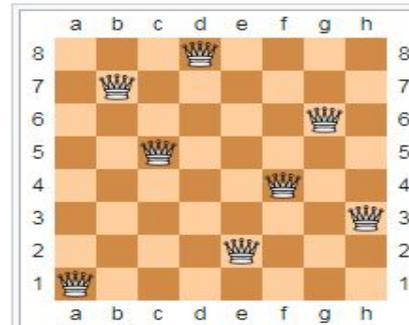
входные данные: 8



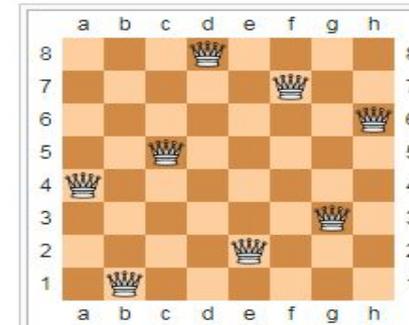
Solution 1



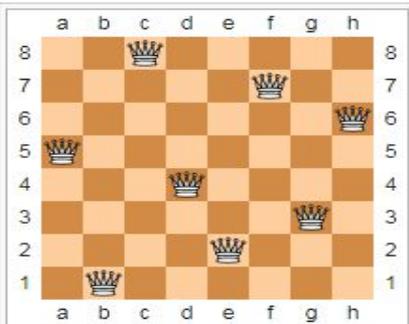
Solution 2



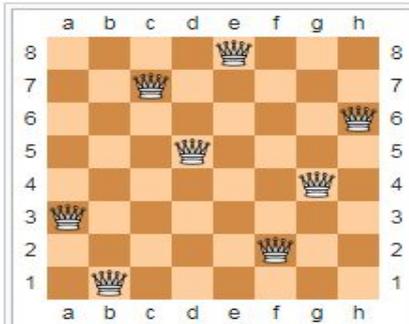
Solution 3



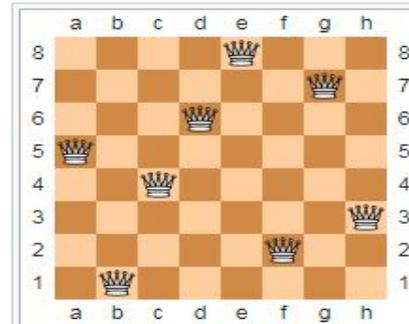
Solution 4



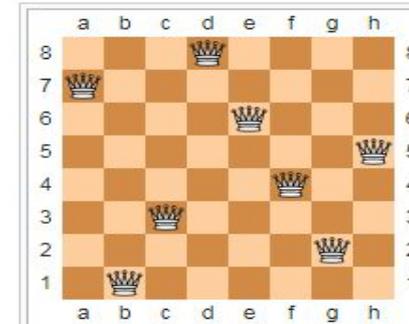
Solution 5



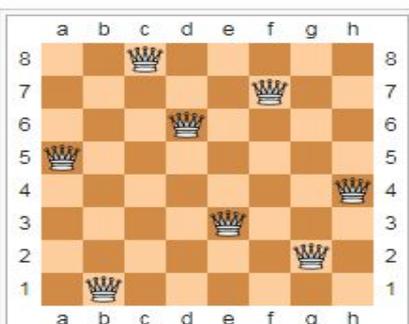
Solution 6



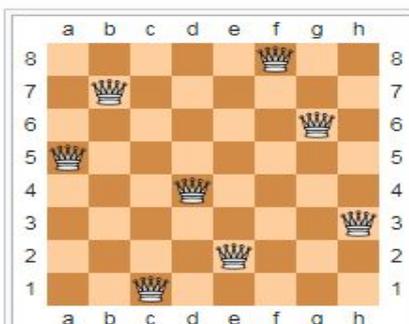
Solution 7



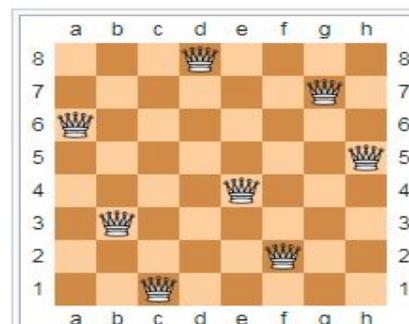
Solution 8



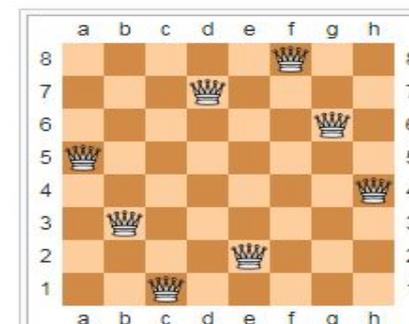
Solution 9



Solution 10



Solution 11



Solution 12

12 УНИКАЛЬНЫХ (БЕЗ ПОВОРОТОВ И ОТРАЖЕНИЙ) РЕШЕНИЙ ЗАДАЧИ «МИРНЫЕ ФЕРЗИ» НА ДОСКЕ 8 x 8

ЗАДАЧА №157. МОНЕТКИ

В стране используются монетки достоинством A_1, A_2, \dots, A_M . Человек пришел в магазин и обнаружил, что у него есть ровно по две монетки каждого достоинства. Ему нужно заплатить сумму N . Напишите программу, определяющую, сможет ли он расплатиться без сдачи.

Входные данные: На вход программы сначала поступает число N ($1 \leq N \leq 10^9$), затем - число M ($1 \leq M \leq 15$) и далее M попарно различных чисел A_1, A_2, \dots, A_M ($1 \leq A_i \leq 10^9$).

Выходные данные: Сначала выведите K - количество монет, которое придется отдать. Далее выведите K чисел, задающих достоинства монет. Если решений несколько, выведите вариант, в котором человек отдаст наименьшее возможное количество монет. Если таких вариантов несколько, выведите любой из них. Если без сдачи не обойтись, то выведите одно число 0. Если же у человека не хватит денег, чтобы заплатить указанную сумму, выведите число -1.

Входные данные

Выходные данные

100 6 3

11 20 30 40 11 99

40 30 30