НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із дисципліни «Дослідження операцій»
на тему:
«Завдання про вісім ферзів»

Виконав:
студент групи КМ-92
Майструк І.А.

Київ — 2021

Задача про вісім ферзів

Задача про вісім ферзів полягає в такому розміщенні восьми ферзів на шахівниці, що жодна з них не ставить під удар один одного. Тобто, вони не повинні стояти в одній вертикалі, горизонталі чи діагоналі. Задача про вісім ферзів є прикладом загальної задачі про n ферзів: задачі розміщення n ферзів на шахівниці розміром n × n, яка має розв'язки для n = 1 або n ≥ 4.

Задача має 92 розв'язки. Якщо відкинути схожі композиції з точністю до відбиття (віддзеркалення позиції на шахівниці) та обертання (обертання позиції на шахівниці), залишиться 12 унікальних розв'язків.

Історія появи задачі

Задачу показав у 1848 р. шахіст Макс Бецель, і вже через три роки математики, зокрема, Карл Гаус, працювали над пошуком її розв'язків та узагальненого варіанту. Перші розв'язки були знайдені Францом Науком 1850 р. Також Наук запропонував розширити задачу до *n* ферзів (на шахівниці розміром *n* × *n*). 1874 р. Ґюнтер запропонував метод пошуку розв'язків із використанням визначників, а Джеймс Глейшер його вдосконалив.

Едсгер Дейкстра використав цю задачу 1972 р. аби показати можливості того, що він назвав структурним програмуванням. Він надрукував докладне описання розробки алгоритму пошуку в глибину з поверненням.

Алгоритм пошуку розв'язків

Задача про ферзів є прикладом простої, але не тривіальної задачі, яку можна розв'язати із допомогою рекурсивного алгоритму, оскільки задача з n ферзями може бути представлена як задача розміщення ферзя для розв'язку з n − 1 ферзями. Нарешті, задачу можна звести до задачі з 8 ферзями, розв'язком

якої є порожня шахівниця. Наступна програма мовою Python знаходить всі розв'язки для задачі з n ферзями використовуючи рекурсивний алгоритм. Він рекурсивно досліджує шахівниці розміром n × n, n × n − 1, n × n − 2,

В програмі враховано те, що ферзі не можуть стояти на одному рядку. Також враховано те, що розв'язок з n − 1 ферзями на шахівниці n × n − 1 містить в собі розв'язок задачі з n − 2 ферзями на шахівниці розміром n × n − 2.

Тобто, алгоритм також отримує всі розв'язки, які входять до знайденого розв'язку на менших шахівницях. Для шахівниці 8×8 програма переглядає 15 720 позицій.

Класична задача

Всього задача про вісім ферзів має 92 розв'язки, але відкинувши розв'язки, які можна отримати відбиттям (віддзеркаленням позиції) та обертанням (обертанням позиції на шахівниці) лишиться 12 унікальних розв'язків. Оскільки кожен унікальний розв'язок має чотири відбиття (через діагональ, горизонталь, вертикаль, та середину шахівниці), та чотири повороти, можна отримати 8·12 = 96 розв'язки. Оскільки один з розв'язків (діаграма № 12), залишається незмінним при повороті на 180°, з нього можна отримати лише чотири розв'язки. Завдяки йому, загальна кількість розв'язків класичної задачі дорівнює 92.

Кількість розв'язків для довільних n

Верхня границя кількості розв'язків D(n) задачі розміру n дорівнює n!, що дорівнює кількості розв'язків для n тур. Кількість ферзів, які не ставлять під удар одна одну, набагато менша за це число.

Моя реалізація на мові Python:

```
global queen, counter, queens_amount

# Функция вывода шахматной доски
def chess_board_output():
    global counter, queens_amount
    x = y = 0
    counter += 1
    print('%d-ая разметка:\n' % counter)
    for x in range(queens_amount):
        for y in range(queens_amount):
            if x == queen[y]:
                print('[Q]', end='')
            else:
                print('[ ]', end='')
        print('')
    input('\n ...нажмите любую клавишу, чтобы увидеть следующую... \n')

# Функция проверки статуса ферзя(1 = в опасности, 0 = в безопасности)
def get_state(row, col):
    global queen
    i = 0
    queen_state = 0
    offset_row = offset_col = 0
    while (queen_state != 1) and i < col:
        offset_col = abs(i - col)
        offset_row = abs(queen[i] - row)
        # Проверка, находятся ли два ферзя на одной линии или на диагонали
        if queen[i] == row or offset_row == offset_col:
            queen_state = 1
        i = i + 1
    return queen_state

def position_process(value):
    global queen, queens_amount
    i = 0
    while i < queens_amount:
        if not get_state(i, value):
            queen[value] = i
            if value == 7:
                chess_board_output()
            else:
                position_process(value + 1)
        i += 1

# Основная программа
def main():
    global queen, counter, queens_amount
    queens_amount = 8    # Кол-во ферзей(для стандартной доски не больше 8)
    queen = [None] * 8   # Сохранить позицию в ряду из 8 ферзей
    counter = 0          # Общее количество решений
    print("Майстурук Илья\tКМ-92\nЗадача о 8-ми ферзях\n")
    position_process(0)

if __name__ == "__main__":
    main()
```

Скріншоти виконання програми

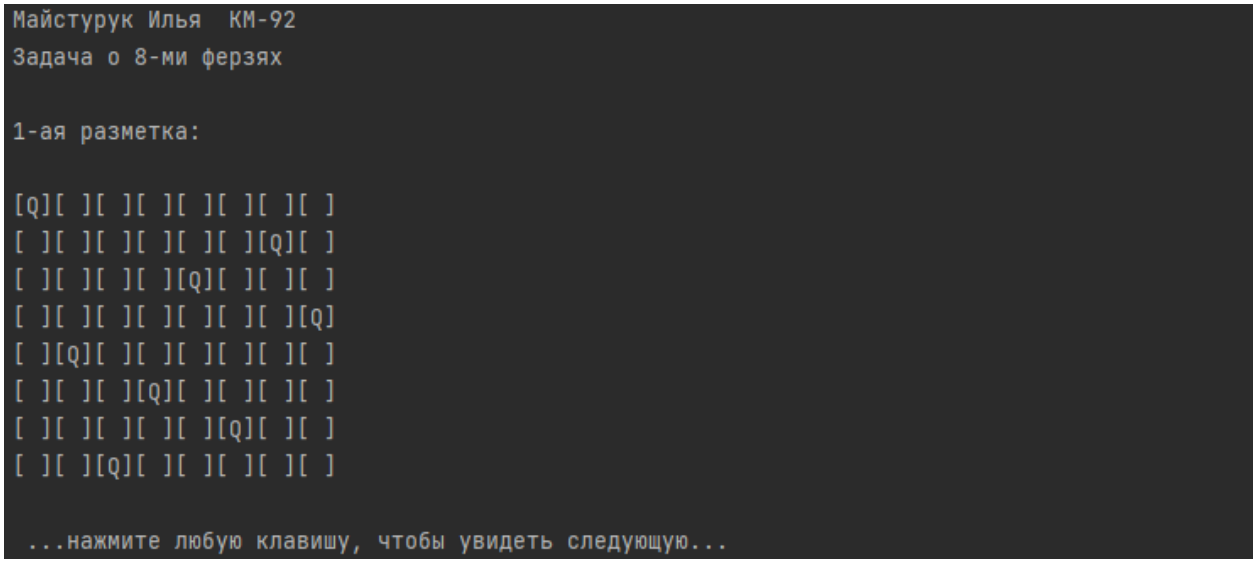

Рис. 1 – Початок виконання програми, перша ітерація

Рис. 2 – Кінець виконання програми, 92-га ітерація