

Лабораторна робота №8.2

Для довільної шахівниці $n \times m$ підрахуйте кількість різних способів обійти всі клітини шаховим конем із заданої клітини. Окремо виведіть кількість замкнених обходів.

Код

```
def is_valid_move(board, n, m, row, col):
    if row >= 0 and row < n and col >= 0 and col < m and board[row][col] == -1:
        return True
    return False

def knight_tour(n, m):
    board = [[-1 for _ in range(m)] for _ in range(n)]
    move_count = 0
    start_row = 0
    start_col = 0
    board[start_row][start_col] = move_count

    count = [0]
    closed_tours = [0]

    find_tour(board, n, m, start_row, start_col, move_count + 1, count, closed_tours)
    print(f"І ще {count[0]-4} способів обходу")
    print(f"Кількість замкнених обходів: {closed_tours[0]}")

def find_tour(board, n, m, row, col, move_count, count, closed_tours):
    moves = [
        (2, 1), (1, 2), (-1, 2), (-2, 1),
        (-2, -1), (-1, -2), (1, -2), (2, -1)
    ]

    if count[0] < 4 and move_count == n * m:
        print_board(board, n, m)
        count[0] += 1
        return
    elif move_count == n * m:
        count[0] += 1
        return

    # Перевірка, чи останній хід коня зводить його на початкову позицію
    last_move = moves.index((row - board[row][col], col - board[row][col]))
    if (row + moves[last_move][0], col + moves[last_move][1]) == (0, 0):
        closed_tours[0] += 1

    return

    for move in moves:
        next_row = row + move[0]
        next_col = col + move[1]

        if is_valid_move(board, n, m, next_row, next_col):
            board[next_row][next_col] = move_count
            find_tour(board, n, m, next_row, next_col, move_count + 1, count,
closed_tours)
            board[next_row][next_col] = -1

    # Додатковий код для завершення підрахунку, якщо було знайдено достатньо
способів
```

```

        if count[0] >= 1000:
            return

def print_board(board, n, m):
    """
    Виводить шахівницю з номерами ходів коня.
    """
    for row in range(n):
        for col in range(m):
            print(board[row][col], end="\t")
        print()
    print()

knight_tour(5, 5)

```

Результат

"H:\University\2 курс\2 семестр\Дискретн.

```

0  5  14  9  20
13 8  19  4  15
18 1  6  21 10
7  12 23 16  3
24 17  2  11 22

```

```

0  5  10 17 20
11 16 19  4  9
6  1  14 21 18
15 12 23  8  3
24 7  2  13 22

```

```

0  5  10 15 20
11 14 19  4  9
6  1  12 21 16
13 18 23  8  3
24 7  2  17 22

```

```

0  5  16 11 20
15 10 19  4 17
6  1  8  21 12
9  14 23 18  3
24 7  2  13 22

```

І ще 300 способів обходу
Кількість замкнених обходів: 0

"H:\University\2 курс\2 семестр\Дискретні Ст

0	19	10	15	4	17
11	22	5	18	9	14
20	1	12	7	16	3
23	6	21	2	13	8

0	13	10	21	4	17
11	22	5	18	9	20
14	1	12	7	16	3
23	6	15	2	19	8

0	17	10	21	4	23
11	14	5	18	9	20
16	1	12	7	22	3
13	6	15	2	19	8

0	17	10	21	4	19
11	14	5	18	9	22
16	1	12	7	20	3
13	6	15	2	23	8

І ще 216 способів обходу

Кількість замкнених обходів: 0

Process finished with exit code 0