



# Бот. Звіт — опис алгоритму роботи

**parse\_field\_info() → tuple[int, int]:**



Дістає розмір поля за допомогою input().

**parse\_field(size: tuple[int, int]) -> list[str]:**



На вхід бере розмір поля і зчитує, використовуючи input(), поле у вигляді списку рядків з символів ".", "X" та "O".

**parse\_figure() -> tuple[tuple[int, int], list[list[int]]]:**



Зчитує розмір фігури, та саму фігуру, використовуючи input(). Фігура обрізається з правого та нижнього країв.

**make\_move(field\_size: tuple[int, int], field: list[str],  
figure\_size: tuple[int, int], figure: list[str],  
player: int) -> tuple[int, int]:**



На вхід приймає інформацію про стан гри: поле, фігуру яку треба поставити та гравця. Повертає валідну позицію(координати) розміщення фігури, обраної програмою, на полі.

Ця функція є основною роботою програми.

Алгоритм використаний у кінцевому варіанті наступний:

1. Згенерувати усі можливі валідні ходи для заданого гравця:
  - a. виконується ітерація по кожній клітинці поля для розгляду можливості розміщення фігури
  - b. ітерація по кожному елементу фігури
  - c. перевірка кількості накладань зірочок фігури на символи даного гравця
  - d. якщо накладань рівно одне(overlap == 1), то ця позиція на полі записується в кінцевий список координат
2. Серед можливих ходів обрати одну позицію:  
позицію з найменшою відстанню до найближчої клітинки суперника  
відстанню до клітинки вважається мінімальна різниця координат від кожного з кутів фігури, розташованої на позиції, та координатою точки.

До остаточного варіанту були написані проміжні алгоритми:

1. Простий вибір з валідних ходів за індексом
2. Вибір ходу найближчого до центру
3. Відстань до центру + відстань до найближчої клітинки суперника