Условие задачи

- Задано клетчатой поле $w \times h$
- На нем выделены s клеток — базы
- Задана последовательность роботов
  - Каждая группа роботов принадлежит одной из s баз
  - Каждый робот может дойти до любой базы в квадрате $(2 \cdot m_i + 1) \times (2 \cdot m_i + 1)$ с центром в базе
    - За исключением, что роботы не ходят вне поля
- В конце в каждой клетке могут находится не более q роботов
- Найти максимальный префикс последовательности, что роботов можно так разместить в клетках

Подзадача 1

- $w, h \leq 20$, одна база и $q = 1$
- Это значит, что всего wh роботов может стоять на поле
- Переберем ответ: число роботов, которых собираемся разметить
  - Упорядочим роботов по мобильности
  - Начиная с менее мобильного размещаем роботов жадно
- Итого время работы $\mathcal{O}(w^3h^3)$ или немного быстрее, в зависимости от реализации

Подзадача 5

- $w, h \leq 10^5$, одна база и $q = 1$
- Применим двоичный поиск по числу размещаемых роботов
- Упорядочим группы роботов по мобильности
- Начиная с менее мобильной группы пытаемся размещать группы роботов жадно
  - Поддерживаем сколько клеток свободно на текущем расстоянии
  - Если число роботов, которых нужно разместить, превышает число свободных клеток, то нельзя
- Итого время работы $\mathcal{O}(\log(wht) \cdot t \log t)$

Идеи

1. Задача о максимальном паросочетании в двудольном графе
   - Если поле маленькое, то можно роботов и места в клетках представить как двудольный граф
     - Левая доля — роботы
     - Правая — места в клетках
   - Тогда размещение роботов в этих клетках — это паросочетание в этом графе, и наоборот
   - Нужно уметь определять, существует ли паросочетание, которое покрывает левую долю
2. Двоичный поиск по ответу
   - Если можно расположить множество роботов, то, убрав одного робота, их все еще можно расположить

Подзадачи 2 и 3

- $w, h \leq 20$, две или три базы, $q = 1$
- Сделаем двоичный поиск по ответу
- Для проверки построим двудольный граф и будем находить паросочетание в нем
- Одни из известных алгоритмов для поиска максимального паросочетания в двудольных графах:
  - Алгоритм Куна
  - Алгоритм Форда-Фалкерсона
- Число вершин в графе $\mathcal{O}(wh)$, число ребер $\mathcal{O}((wh)^2)$
- Общее время работы: $\mathcal{O}((wh)^3 \log wht)$

Подзадача 4

- $w, h \leq 20$, три базы, $q \leq 100$
- Объединим места в клетках в одну вершину
- Паросочетание можно представить в виде потока в сети
- Применим такой же алгоритм, что и в прошлых подзадачах, только будем находить максимальный поток

Полное решение

- $w, h \leq 10^5$, число баз не больше 4, $q \leq 100$
- Давайте научимся проверять без паросочетания
- Для одной базы мы выяснили, что если отсортировать роботов по мобильности и для каждого префикса этой последовательности выполняется:
  - (число мест, до которых могут дойти роботы) $\geq$ (число роботов в этом префиксе)
- Тогда роботов можно разместить по клеткам
- Обобщим этот критерий:
  - Для каждой базы возьмем своих роботов и отсортируем по мобильности
  - Если мы попробуем всеми способами выбрать все различные наборы префиксов
  - И для каждого набора префиксов выполняется:
    - (число мест) $\geq$ (число роботов)
    - места — до которых может дойти хотя бы один робот из всех префиксов
    - роботы — все роботы выбранных префиксов

Доказательство критерия

- Этот критерий следует из Леммы Холла о паросочетании в двудольном графе
- Ее можно сформулировать так
  - Есть n девушек и m парней
  - Девушки хотят пойти на бал
    - Девушка пойдет на бал только со знакомым парнем
  - Если выполняется следующее:
    - любые k выбранных девушек в объединении знают не менее k парней
  - То можно сделать пары так, чтобы каждая девушка нашла себе пару на бал
- Это доказывает наш критерий
  - Наш алгоритм не рассматривает все подмножества
    - Но если подмножество не рассмотрено, в него всегда можно добавить робота с меньшей мобильностью
    - Число мест от этого не поменяется
  - Критерии эквивалентны

Полное решение, реализация

- Переберем все различные наборы префиксов рекурсивно
- Для каждого из наборов посчитаем, сколько мест достижимо
  - Для этого нужно найти площадь объединения прямоугольников
  - Это делается с помощью формулы включений-исключений
  - За время $\mathcal{O}(2^s)$
- Сделаем двоичный поиск
- Снова переберем все различные наборы префиксов
  - Мы знаем, сколько есть роботов в префиксах
  - Сравниваем с соответствующим числом мест
- Итого, решение за время $\mathcal{O}\left(2^s \cdot \left(\frac{t}{s}\right)^s + \log\left(whqt\right) \cdot \left(\frac{t}{s}\right)^s\right)$.