Разбор задачи «Роботы на шахматной доске»

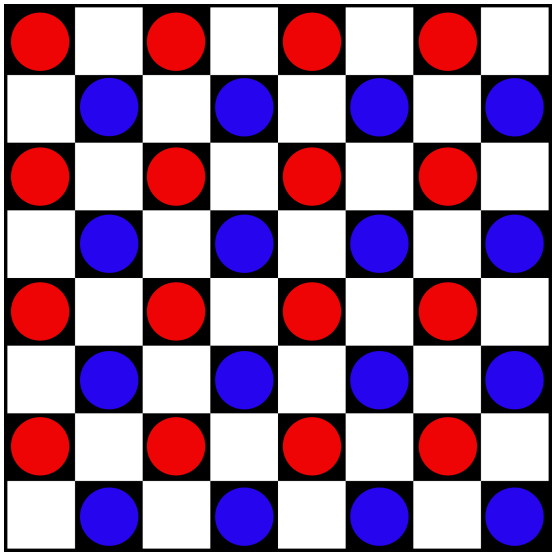
Необходимо найти наименьшее количество роботов, чтобы обойти все черные клетки доски размером $n \times m$ за время не более k минут. Каждый робот может двигаться со скоростью 1 клетка в минуту.

Ключевые наблюдения

1. **Разбиение на компоненты:**
Черные клетки шахматной доски можно разбить на две независимые компоненты:

- Первая компонента (клетки с синим кругом)
- Вторая компонента (клетки с красным кругом)

Робот из одной компоненты не может попасть в другую из-за инварианта по модулю 2.

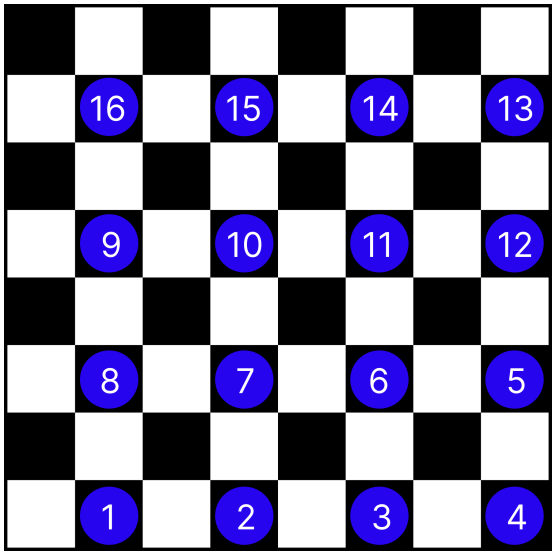


2. **Количество клеток в компонентах:**
Для шахматной доски $n \times m$:

$$cnt_1 = \left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lfloor \frac{m+1}{2} \right\rfloor$$

$$cnt_2 = \left\lfloor \frac{n+1}{2} \right\rfloor \cdot \left\lfloor \frac{m}{2} \right\rfloor$$

3. **Нумерация клеток для обхода:**
Для каждой компоненты пронумеруем клетки в порядке обхода:



Решение для одной компоненты

Для компоненты с cnt_{cell} клетками минимальное количество роботов:

$$cnt_{robot} \geqslant \frac{cnt_{cell}}{k}$$

$$cnt_{robot} = \left\lceil \frac{cnt_{cell}}{k} \right\rceil$$

Итоговое решение

Общее количество роботов:

$$ans = \left\lceil \frac{cnt_1}{k} \right\rceil + \left\lceil \frac{cnt_2}{k} \right\rceil$$

Разбор задачи «Оптимизация магической энергии»

Условие задачи

Дано:

- Массив кристаллов с энергиями $a_1, a_2, \ldots, a_n$
- Суммарная энергия вычисляется по формуле:

$$E = \sum_{i=1}^{n-1} \frac{a_i + a_{i+1}}{2}$$

- Можно выполнить не более двух обменов элементов массива
- Необходимо минимизировать E

Анализ формулы суммарной энергии

Раскроем сумму:

$$E = \frac{a_1 + a_2}{2} + \frac{a_2 + a_3}{2} + \cdots + \frac{a_{n-1} + a_n}{2}$$

Упростим выражение:

$$E = \frac{1}{2}\left((a_1 + a_2) + (a_2 + a_3) + \cdots + (a_{n-1} + a_n)\right) = \frac{1}{2}\left(a_1 + 2a_2 + 2a_3 + \cdots + 2a_{n-1} + a_n\right)$$

Таким образом:

$$E = \frac{a_1}{2} + a_2 + a_3 + \cdots + a_{n-1} + \frac{a_n}{2}$$

Минимизация E

Чтобы минимизировать E, нужно: поместить **наибольшие** элементы в позиции a_1 и a_n, так как они входят в сумму с меньшим коэффициентом $\frac{1}{2}$

Оптимальная стратегия:

- Отсортировать массив по убыванию
- Поместить два наибольших элемента в начало и конец массива (позиции a_1 и a_n)
- Остальные элементы могут быть в любом порядке (их коэффициенты одинаковы)

Разбор задачи «Оптимизация доставки роботами»

Условие задачи

Дано n пунктов назначения с координатами $x_1, \ldots, x_n$ на числовой прямой. Требуется выбрать точку y так, чтобы минимизировать суммарные потери энергии при доставке грузов роботами. Потери определяются как $b_i \cdot (x_i - y)$ при движении вправо ($y \leqslant x_i$) и $a_i \cdot (y - x_i)$ при движении влево ($y > x_i$). Необходимо обработать Q запросов с разными (a_i, b_i).

Ключевые наблюдения

Первым важным наблюдением является то, что оптимальное положение точки y всегда совпадает с одной из точек x_i. Это следует из анализа поведения функции суммарных потерь: при перемещении y между двумя соседними точками x_{i-1} и x_i функция ведет себя линейно, а значит, минимум достигается на границах интервала.

Функция суммарных потерь $F(y)$ обладает свойством выпуклости, так как представляет собой сумму выпуклых кусочно-линейных функций. Каждое слагаемое $f_i(y)$ имеет "излом"в точке x_i и является выпуклым, следовательно, их сумма также выпукла. Это важное свойство позволяет применить для поиска минимума метод бинарного или тернарного поиска, так как выпуклая функция унимодальна — сначала строго убывает до точки минимума, затем строго возрастает.

Предварительная обработка данных

Для эффективного решения необходимо предварительно отсортировать массив координат x_i по возрастанию. Это позволяет в дальнейшем быстро разделять точки на лежащие слева и справа от выбранной позиции y. Далее вычисляется массив префиксных сумм $pref$, который будет использоваться для быстрого вычисления сумм элементов на любом подотрезке.

Особое внимание следует уделить обработке дубликатов координат. Поскольку при одинаковых x_i значения функции в этих точках будут идентичны, достаточно рассматривать только уникальные координаты. Это значительно сокращает пространство поиска без потери точности.

Разбор задачи «Почта»

Формулировка задачи и ключевые идеи

Робот-почтальон должен доставить письмо из центрального отделения в дом получателя, проходя через цепочку из n почтовых станций. Начальный и конечный участки пути преодолеваются пешком за фиксированное время x и y соответственно. Основная сложность заключается в оптимальном использовании автобусов между станциями: для каждого перехода $i \to i+1$ автобусы отправляются с интервалами, кратными p_i, а сама поездка занимает t_i времени. Необходимо для Q моментов выхода q_i определить минимальное время доставки.

Анализ временных характеристик

Время $f(p)$, необходимое роботу, чтобы добраться до пункта назначения, если он выходит из своего дома в момент времени p. Функция строится последовательно. Сначала робот тратит x времени на пеший переход до первой станции, прибывая в момент $q + x$. Далее для каждой станции i вычисляется ближайший момент отправления автобуса: $\lceil \frac{x_i}{p_i} \rceil \cdot p_i$, после чего добавляется время поездки t_i. Критически важно, что если робот прибывает точно в момент отправления, он успевает на автобус без задержки. Финальный пеший переход добавляет y к общему времени.

Оптимизация через периодичность

Ключевое наблюдение: поскольку все p_i ограничены значением 8, система демонстрирует периодическое поведение с периодом $L = \mathrm{lcm}(p_1, ..., p_{n-1}) \leqslant 840$. Это позволяет предвычислить значения $f(k)$ для всех $k \in [0, L-1]$ за $O(nL)$ операций. Для произвольного запроса q_i ответ определяется через остаток от деления: $f(q_i) = f(q_i \mod L)$. Такой подход сокращает сложность обработки Q запросов с $O(nQ)$ до оптимальных $O(nL + Q)$.

Обоснование корректности

Периодичность сохраняется благодаря свойствам LCM: для любого q и станции i выполняется $\lceil \frac{x_i+L}{p_i} \rceil \cdot p_i = \lceil \frac{x_i}{p_i} \rceil \cdot p_i + L$, так как L кратно p_i. Это гарантирует, что времена прибытия на станции для моментов выхода q и $q + L$ отличаются ровно на L, но итоговое время доставки $f(q)$ остается инвариантным относительно сдвигов на период.

Практическая реализация

Алгоритм состоит из двух этапов:

1. Предварительный расчет $f(k)$ для всех остатков k от 0 до $L - 1$ путем последовательного моделирования маршрута.

2. Мгновенный ответ на запросы через таблицу предвычисленных значений.

Разбор задачи «Система подчинения роботов»

1 Формализация задачи

Имеется ориентированный граф $G = (V, E)$, где $V = \{1, \ldots, n\}$ — множество роботов, а E — множество пар делегирования. Требуется построить ориентированное дерево T с корнем r (лидером), удовлетворяющее следующим условиям:

- Для любого робота $v \in V$ множество подчинённых v в дереве T совпадает с множеством вершин, достижимых из v в графе G.
- В дереве T у каждой вершины, кроме корня, ровно один предок.

2 Ключевые наблюдения

1. **Ацикличность.** Если граф G содержит циклы, то ответа не существует. Действительно, в дереве T для любой вершины v её подчинённые образуют поддерево, и если в G есть цикл, то все вершины цикла должны быть подчинёнными друг друга, что невозможно в дереве.
2. **Топологическая сортировка.** Если граф ациклический, то его можно топологически отсортировать. Пусть $\mathrm{ord}[v]$ — порядковый номер вершины v в топологической сортировке.
3. **Построение дерева.** Для каждой вершины $v \neq r$ выберем её предка p_v как вершину u с максимальным $\mathrm{ord}[u]$ среди всех вершин, из которых есть ребро в v в графе G.
4. **Проверка корректности.** Для всех рёбер $(x, y) \in E$ необходимо убедиться, что x является предком y в дереве T. Это можно проверить с помощью эйлерова обхода дерева и подсчёта времени входа $\mathrm{in}[v]$ и выхода $\mathrm{out}[v]$ для каждой вершины. Тогда x — предок y, если $\mathrm{in}[x] \leqslant \mathrm{in}[y] \leqslant \mathrm{out}[x]$.