



Разбор задачи «Набор воды домой»

1 Постановка задачи

Определите, сколько всего раз Ильнуру придётся заплатить, чтобы набрать n бутылок воды по k литров каждая.

Даны четыре тестовых набора, для которых требуется найти ответ. Если вы не знаете ответ для теста, запишите -1 .

Номер теста	n	k	a	Ответ
1	121	5	0	121
2	198	173	0.5	132
3	139	3737	0.23	114
4	203881	61329	0.7319	117722

2 Разбор решения

2.1 Ключевые моменты

- **Ошибка автомата:** После каждой оплаты автомат наливает $k \cdot (1 + a)$ литров вместо k .
- **Управление наливом:** Ильнур может использовать "Паузу" и "Старт" для остановки и возобновления налива, что позволяет использовать излишки $(a \cdot k)$ для заполнения следующих бутылок.
- **Минимизация оплат:** Нужно найти минимальное количество оплат, чтобы суммарно налить $n \cdot k$ литров.

2.2 Формула для расчета

Общий объем воды, который нужно набрать:

$$\text{total} = n \cdot k$$

За одну оплату автомат наливает:

$$\text{за одну оплату} = k \cdot (1 + a)$$

Минимальное количество оплат:

$$\text{payments} = \left\lceil \frac{\text{total}}{k \cdot (1 + a)} \right\rceil = \left\lceil \frac{n}{1 + a} \right\rceil$$

2.3 Обоснование

- Если $a = 0$, то автомат наливает ровно k литров за оплату, поэтому $\text{payments} = n$.
- Если $a > 0$, каждая оплата дает больше воды, поэтому количество оплат уменьшается.

3 Ответы для тестовых случаев

1. Тест 1: $n = 121$, $k = 5$, $a = 0$.

$$\text{payments} = \left\lceil \frac{121}{1 + 0} \right\rceil = 121$$

Ответ: 121



2. Тест 2: $n = 198$, $k = 173$, $a = 0.5$.

$$\text{payments} = \left\lceil \frac{198}{1 + 0.5} \right\rceil = \lceil 132 \rceil = 132$$

Ответ: 132

3. Тест 3: $n = 139$, $k = 3737$, $a = 0.23$.

$$\text{payments} = \left\lceil \frac{139}{1 + 0.23} \right\rceil \approx \lceil 113.008 \rceil = 114$$

Ответ: 114

4. Тест 4: $n = 203881$, $k = 61329$, $a = 0.7319$.

$$\text{payments} = \left\lceil \frac{203881}{1 + 0.7319} \right\rceil \approx \lceil 117721.000058 \rceil = 117722$$

Ответ: 117722

Разбор задачи «Оптимизация магической энергии»

Условие задачи

Дано:

- Массив кристаллов с энергиями $a_1, a_2, \dots, a_n$
- 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:

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

Упростим выражение:

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

Таким образом:

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



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 «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.

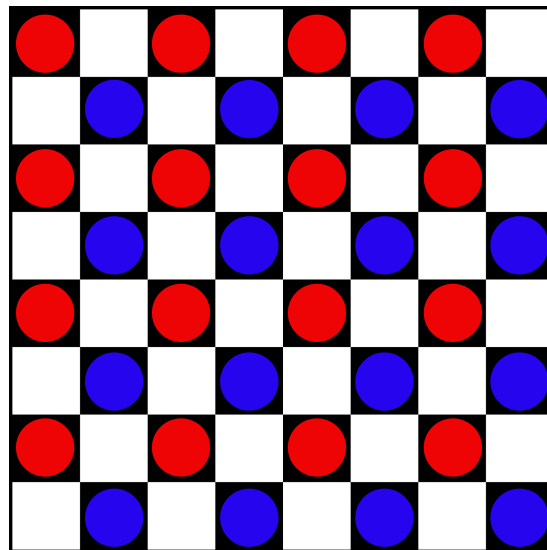
Ключевые наблюдения

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:

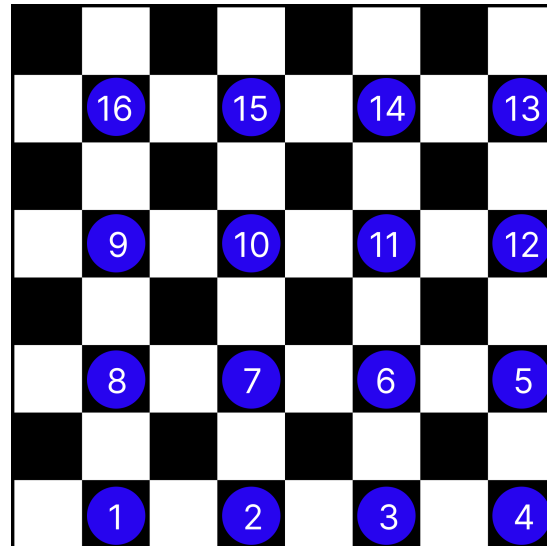
Для шахматной доски $n \times m$:

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

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

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:

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

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 доставки роботами»

Условие задачи

Дано n пунктов назначения с координатами $x_1, \dots, x_n$ 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 ($y \leq x_i$) 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 $i \rightarrow i+1$ 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 $L = \text{lcm}(p_1, \dots, p_{n-1}) \leq 840$. 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: $f(q_i) = f(q_i \bmod L)$. 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.