

Задача А. Максимальное передаточное число в системе шестерен

Даны n шестеренок с количеством зубьев a_i . Необходимо найти подотрезок подряд идущих шестеренок $a_l, a_{l+1}, \dots, a_r$, такой что произведение отношений соседних шестеренок $\frac{a_l}{a_{l+1}} \cdot \frac{a_{l+1}}{a_{l+2}} \dots \frac{a_{r-1}}{a_r}$ максимально. После сокращений это произведение равно $\frac{a_l}{a_r}$. Таким образом, задача сводится к нахождению индексов $l \leq r$, максимизирующих $\frac{a_l}{a_r}$.

Идея решения

Заметим, что для фиксированного l максимальное значение $\frac{a_l}{a_r}$ достигается при минимальном a_r среди всех $r \geq l$. Поэтому для каждого индекса l достаточно рассмотреть минимальный элемент на суффиксе $[l, n-1]$. Остается перебрать все l и выбрать максимальное отношение $\frac{a_l}{\min_{r \geq l} a_r}$.

Для эффективного нахождения минимума на суффиксе можно предподсчитать суффиксные минимумы:

$$\text{sufMin}[i] = \min(a[i], \text{sufMin}[i+1]), \quad \text{sufMin}[n-1] = a[n-1]$$

После нахождения максимального отношения дробь сокращается на НОД числителя и знаменателя.

Задача В. Улучшение системы шестеренок

Даны n шестеренок с количеством зубьев a_i . Общее передаточное число от первой до последней шестеренки равно $\frac{a_1}{a_2} \cdot \frac{a_2}{a_3} \dots \frac{a_{n-1}}{a_n} = \frac{a_1}{a_n}$. Требуется с помощью минимального количества обменов любых двух шестеренок сделать это отношение максимальным. За один обмен можно поменять местами шестеренки на позициях i и j .

Идея решения

Поскольку передаточное число равно $\frac{a_1}{a_n}$, для его максимизации нужно:

1. Сделать числитель a_1 как можно больше
2. Сделать знаменатель a_n как можно меньше

Таким образом, нужно разместить максимальный элемент массива на первую позицию, а минимальный — на последнюю.

Рассмотрим случаи:

1. **Случай 1:** Если уже $a_1 = \max(a)$ и $a_n = \min(a)$ — ответ 0.
2. **Случай 2:** Если $a_1 = \max(a)$ но $a_n \neq \min(a)$ — достаточно 1 обмена: поменять минимальный элемент с последним.
3. **Случай 3:** Если $a_n = \min(a)$ но $a_1 \neq \max(a)$ — достаточно 1 обмена: поменять максимальный элемент с первым.
4. **Случай 4:** Если $a_1 = \min(a)$ и $a_n = \max(a)$ — достаточно 1 обмена: поменять первый и последний элементы.
5. **Случай 5:** Во всех остальных случаях нужно 2 обмена: сначала поставить максимальный на первое место, затем минимальный на последнее (или наоборот).

Задача С. Встреча в плацкартном поезде

Дан поезд из n вагонов с заполненностями a_i . Камиль начинает в тамбуре 0 и идет вправо, Вадим начинает в тамбуре n и идет влево. При встрече в тамбуре k :

- Путь Камия: сумма $a_1 + a_2 + \dots + a_k$ (обозначим P_k)
- Путь Вадима: сумма $a_n + a_{n-1} + \dots + a_{k+1}$ (это $S - P_k$, где $S = \sum_{i=1}^n a_i$)

Нужно найти k (от 0 до n), минимизирующий $|P_k - (S - P_k)| = |2P_k - S|$, а при равенстве выбрать минимальный k .

Идея решения

Обозначим:

- $S = \sum_{i=1}^n a_i$ — сумма всех заполненностей
- $P_k = \sum_{i=1}^k a_i$ — префиксная сумма до k -го вагона (при $k = 0$: $P_0 = 0$)

Тогда:

- Сложность пути Камиля до тамбура k : $C_{\text{кам}} = P_k$
- Сложность пути Вадима до тамбура k : $C_{\text{вад}} = S - P_k$
- Разность: $|C_{\text{кам}} - C_{\text{вад}}| = |P_k - (S - P_k)| = |2P_k - S|$

Таким образом, задача сводится к поиску $k \in [0, n]$, минимизирующего $|2P_k - S|$.

Ключевое наблюдение: При увеличении k на 1 значение P_k увеличивается на $a_{k+1} \geq 0$, поэтому $2P_k - S$ монотонно не убывает. Значит, минимум модуля будет достигаться в точке, где $2P_k - S$ меняет знак с отрицательного на положительный, или там, где оно равно нулю.

Оптимизация: Не нужно хранить все префиксные суммы, достаточно накапливать текущую сумму в переменной.

Задача D. Оптимальная поездка

Петя едет путь длиной S км. Его скорость v ограничена: $a \leq v \leq a + d$. Энергия на участке длины L со скоростью v : $E_{\text{seg}} = k \cdot v \cdot L$. Общий запас энергии: E , причём гарантируется $E \geq k \cdot a \cdot S$. Нужно минимизировать время пути, а при равном времени — минимизировать затраченную энергию.

Идея решения

Заметим, что при постоянной скорости v :

- Время прохождения: $T = \frac{S}{v}$
- Затраченная энергия: $En = k \cdot v \cdot S$

Время T убывает с ростом v , энергия En возрастает с ростом v .

Ключевое наблюдение: Оптимальная стратегия — ехать с постоянной скоростью, равной максимально возможной при ограничениях. Покажем это:

1. Чтобы минимизировать время, нужно максимизировать скорость.

2. Ограничения:

- По условию: $a \leq v \leq a + d$
- По энергии: $k \cdot v \cdot S \leq E$, то есть $v \leq \frac{E}{k \cdot S}$

3. Следовательно, максимальная допустимая скорость:

$$v_{\text{opt}} = \min \left(a + d, \frac{E}{k \cdot S} \right)$$

Если $v_{\text{opt}} = a + d$, то время минимально, энергия не превышает E . Если $v_{\text{opt}} = \frac{E}{k \cdot S}$, то время больше, но энергия в точности равна E (лимиту).

Алгоритм

1. Вычислить максимальную скорость по физическому ограничению: $v_{\max_speed} = a + d$
2. Вычислить максимальную скорость по энергетическому ограничению: $v_{\max_energy} = \frac{E}{k \cdot S}$
3. Оптимальная скорость: $v_{\text{opt}} = \min(v_{\max_speed}, v_{\max_energy})$
4. Время: $T = \frac{S}{v_{\text{opt}}}$
5. Энергия: $En = k \cdot v_{\text{opt}} \cdot S$

Задача Е. Оптимальные рейсы лифта

Имеется m сотрудников, находящихся на этажах $a_1, \dots, a_m$. Лифт:

- Начинает и заканчивает на 1 этаже
- За рейс может сделать не более k остановок (не считая 1 этажа)
- Может забирать всех сотрудников с этажа за одну остановку
- Порядок остановок произвольный
- Все этажи с сотрудниками должны быть посещены хотя бы раз

Требуется минимизировать количество рейсов.

Идея решения

Заметим ключевые моменты:

1. Поскольку на одном этаже может быть несколько сотрудников, достаточно посетить каждый этаж с сотрудниками хотя бы один раз. Порядок посещения внутри рейса не важен.
2. Лифт всегда возвращается на 1 этаж, поэтому каждый рейс независим.
3. Задача сводится к разбиению множества уникальных этажей на группы размера не более k (остановок за рейс).

Таким образом, если U — количество уникальных этажей, то минимальное количество рейсов равно:

$$r_{\min} = \left\lceil \frac{U}{k} \right\rceil$$

где $\lceil x \rceil$ — округление вверх.

Задача F. Испытание дрона

Дрон стартует из точки $S = (S_x, S_y)$, должен достичь $E = (E_x, E_y)$ за ровно k секунд. На каждом шаге:

- Дрон выбирает свою скорость (v_x, v_y) так, что $\sqrt{v_x^2 + v_y^2} \leq v_{\max}$
- Вектор поля в секунду i равен (q_{x_i}, q_{y_i})
- Фактическое смещение: $(v_x + q_{x_i}, v_y + q_{y_i})$

Поле задано n интервалами постоянного значения, сумма длительностей равна k .

Идея решения

Обозначим:

- $D_i = (q_{x_i}, q_{y_i})$ — вектор поля в секунду i
- $V_i = (v_{x_i}, v_{y_i})$ — скорость дрона в секунду i (относительно среды)

Тогда перемещение за секунду: $D_i + V_i$.

За k секунд общее перемещение:

$$\sum_{i=1}^k (D_i + V_i) = \sum_{i=1}^k D_i + \sum_{i=1}^k V_i = E - S$$

Обозначим $Q = \sum_{i=1}^k D_i$ — суммарное влияние поля. Тогда необходимо:

$$\sum_{i=1}^k V_i = (E - S) - Q = \Delta$$

Условие на каждую секунду: $\|V_i\| \leq v_{\max}$.

Ключевое наблюдение: Если $\|\Delta\| \leq k \cdot v_{\max}$, то можно выбрать все V_i одинаковыми:

$$V_i = \frac{\Delta}{k}$$

При этом $\|V_i\| = \frac{\|\Delta\|}{k} \leq v_{\max}$, что удовлетворяет ограничению.

Алгоритм

1. Вычислить суммарное влияние поля $Q = \sum_{i=1}^k D_i$
2. Вычислить необходимый суммарный вклад дрона: $\Delta = (E - S) - Q$
3. Если $\|\Delta\| > k \cdot v_{\max}$, вывести "No" (невозможно)
4. Иначе вывести "Yes" и траекторию:
 - Начало: $p_0 = S$
 - На каждом шаге: $p_i = p_{i-1} + D_i + \frac{\Delta}{k}$

Доказательство корректности

1. Если $\|\Delta\| \leq k v_{\max}$, то $\|\frac{\Delta}{k}\| \leq v_{\max}$
2. После k шагов:

$$p_k = S + \sum_{i=1}^k D_i + \sum_{i=1}^k \frac{\Delta}{k} = S + Q + \Delta = S + Q + ((E - S) - Q) = E$$

3. Все ограничения выполнены:
 - Скорость дрона: $V_i = \frac{\Delta}{k}$, $\|V_i\| \leq v_{\max}$
 - Время ровно k секунд
 - Конечная точка точно E

Задача G. Робот и доставка

Дан двумерный лабиринт $L \times C$ с базовой стоимостью $p_{i,j} \geq 1$ или -1 (недоступно). Робот стартует в (L_1, C_1) , должен добраться до (L_2, C_2) . Энергия для перехода из клетки (i, j) в соседнюю составляет $p_{i,j} \cdot W$, где W — масса посылки. Общий лимит энергии — E . Нужно найти максимальное целое W , при котором существует путь в пределах лимита E .

Идея решения

Обозначим:

- $P = \sum_{(i,j) \in \text{path}} p_{i,j}$ — сумма базовых стоимостей по выбранному пути
- Тогда общая энергия: $E_{\text{total}} = P \cdot W$

Для фиксированного W задача сводится к поиску кратчайшего пути в графе, где вес ребра (перехода из клетки (i, j)) равен $p_{i,j} \cdot W$. Так как W — константа для конкретной проверки, можно использовать алгоритм Дейкстры.

Ключевое наблюдение: при увеличении W стоимость любого пути возрастает пропорционально. Поэтому существует монотонность: если для W существует путь $\leq E$, то для любого $W' < W$ тоже существует (возможно, другой путь). Это позволяет использовать бинарный поиск по W .

Алгоритм

1. Прочитать данные, преобразовать координаты в 0-индексацию.
2. Бинарный поиск по W в пределах $[0, 2 \cdot 10^9]$ (верхняя граница завышена для надёжности).
3. Для каждого W_{mid} запустить Дейкстру:
 - Вес перехода из (x, y) в соседнюю клетку: $p_{x,y} \cdot W$
 - Найти минимальную энергию для достижения цели
4. Если минимальная энергия $\leq E$, пробуем увеличить W , иначе уменьшаем.
5. Вывести максимальное W , для которого путь существует.

Задача Н. Роботы-наблюдатели

Имеется N роботов, измеряющих нагрузку каждые Z минут в течение M дней. Данные за некоторые моменты отсутствуют. Нужно восстановить пропущенные значения по алгоритму:

Для робота b и времени t (кратного Z):

1. Для каждого из P предыдущих дней находим момент $t_i = t - i \cdot D$ (где $D = 24 \cdot 60$)
2. Для каждого t_i вычисляем среднюю нагрузку за предыдущий час: среднее арифметическое за моменты $t_i - Z, t_i - 2Z, \dots, t_i - 60$
3. Вычисляем относительную нагрузку: $r_i = \frac{\text{нагрузка в } t_i}{\text{средняя за предыдущий час}}$
4. Находим среднее относительной нагрузки по всем P дням: $R = \frac{1}{P} \sum_{i=1}^P r_i$
5. Для текущего дня вычисляем среднюю нагрузку за предыдущий час: A (по тем же правилам)
6. Восстановленное значение: $R \cdot A$

Гарантии: данные есть для первого часа каждого дня, для первых P дней все данные есть.

Идея решения

Ключевые моменты:

1. Время дискретно с шагом Z минут. Всего слотов: $\text{totalSlots} = M \cdot (24 \cdot 60 / Z)$
2. Восстановление можно проводить в хронологическом порядке, так как для вычисления средней за предыдущий час нужны только более ранние данные
3. После восстановления значения нужно сохранить, чтобы использовать при восстановлении последующих значений

Структура данных

- `val[slot][robot]` — значение нагрузки (известное или восстановленное)
- `has[slot][robot]` — флаг наличия исходных данных
- Слоты нумеруются от 0 до `totalSlots - 1`, время $t = \text{slot} \cdot Z$

Алгоритм

1. Инициализация: чтение всех известных значений, установка флагов
2. Проход по всем слотам в порядке возрастания времени
3. Для каждого робота в текущем слоте:
 - Если данные есть — пропускаем
 - Иначе вычисляем восстановленное значение по формуле
 - Сохраняем значение в массив `val` для использования в будущих вычислениях
 - Добавляем ответ в список
4. Вывод ответов в требуемом порядке

Задача I. Баланс на Оси

Даны коэффициенты A , B и ограничение L . Нужно определить, существуют ли целые x и y ($1 \leq x, y \leq L$), такие что:

$$A \cdot x = B \cdot y$$

Идея решения

Перепишем уравнение:

$$A \cdot x = B \cdot y$$

Поделим обе части на $\gcd(A, B)$:

$$\frac{A}{\gcd(A, B)} \cdot x = \frac{B}{\gcd(A, B)} \cdot y$$

Обозначим:

$$a' = \frac{A}{\gcd(A, B)}, \quad b' = \frac{B}{\gcd(A, B)}$$

Тогда a' и b' взаимно просты.

Уравнение принимает вид:

$$a' \cdot x = b' \cdot y$$

Поскольку a' и b' взаимно просты, из равенства следует:

- x должно делиться на b'
- y должно делиться на a'

Пусть $x = b' \cdot k$, $y = a' \cdot k$ для некоторого целого $k \geq 1$.

Тогда исходное уравнение выполняется для любого k .

Условия ограничений:

$$\begin{aligned} 1 \leq x = b' \cdot k \leq L \\ 1 \leq y = a' \cdot k \leq L \end{aligned}$$

Отсюда:

$$k \geq 1, \quad k \leq \frac{L}{b'}, \quad k \leq \frac{L}{a'}$$

Таким образом, решение существует тогда и только тогда, когда:

$$\max(a', b') \leq L$$

Задача J. Последовательность Арсения

Дана последовательность дробей:

$$p_1 = \frac{1}{1+1} = \frac{1}{2}$$

$$p_i = \frac{1}{1+p_{i-1}} \quad \text{для } i \geq 2$$

Нужно найти $p_n = \frac{a}{b}$, где a и b взаимно просты.

Идея решения

Вычислим первые несколько членов последовательности:

$$p_1 = \frac{1}{2}$$

$$p_2 = \frac{1}{1+\frac{1}{2}} = \frac{1}{\frac{3}{2}} = \frac{2}{3}$$

$$p_3 = \frac{1}{1+\frac{2}{3}} = \frac{1}{\frac{5}{3}} = \frac{3}{5}$$

$$p_4 = \frac{1}{1+\frac{3}{5}} = \frac{1}{\frac{8}{5}} = \frac{5}{8}$$

$$p_5 = \frac{1}{1+\frac{5}{8}} = \frac{1}{\frac{13}{8}} = \frac{8}{13}$$

$$p_6 = \frac{1}{1+\frac{8}{13}} = \frac{1}{\frac{21}{13}} = \frac{13}{21}$$

Заметим закономерность: дроби соответствуют последовательным отношениям чисел Фибоначчи:

$$p_n = \frac{F_n}{F_{n+1}}$$

где $F_1 = 1, F_2 = 1, F_3 = 2, F_4 = 3, F_5 = 5, F_6 = 8, \dots$

Докажем это по индукции:

1. База: $p_1 = \frac{1}{2} = \frac{F_1}{F_2}$

2. Предположение: $p_{n-1} = \frac{F_{n-1}}{F_n}$

3. Шаг:

$$p_n = \frac{1}{1+p_{n-1}} = \frac{1}{1+\frac{F_{n-1}}{F_n}} = \frac{1}{\frac{F_n+F_{n-1}}{F_n}} = \frac{F_n}{F_n+F_{n-1}} = \frac{F_n}{F_{n+1}}$$