Открытая Олимпиада Университета Иннополис по программированию

21 февраля 2016 года

Задача А. Зонд на Нибиру

Задача А. Зонд на Нибиру

- Автор задачи — Нияз Нигматуллин
- Подготовка задачи — Нияз Нигматуллин

Постановка задачи

- Задан многоугольник
- Нужно провести вертикальную полосу ширины L
    - Максимизировать площадь пересечения полосы и многоугольника

Решение задачи

- Выпуклый многоугольник
- Функция: $f(x)$ — площадь пересечения, если левый конец полосы в точке x
  - $f(x)$ выпукла
- Троичный поиск или другой метод поиска локального максимума
- $s(a)$ — площадь пересечения полуплоскости $x \leqslant a$ и многоугольника
  - $f(x) = s(x + L) - s(x)$
- Пересечь вертикальную прямую с многоугольником и посчитать площадь нового многоугольника

Решение задачи

- Стороны параллельны осям координат
- Ответ в целой точке
- Перебрать точку, посчитать ответ

Решение задачи

- Сканирующая прямая по x
- Поддерживаем $f(x)$
- Событие начала отрезка и конца
- Каждый отрезок вносит вклад в площадь, как площадь под этим отрезком
    - Трапеция
- Каждый со своим знаком
    - Отрезок слева направо — отрицательный
    - Справа налево — положительный

Решение задачи

- Поддерживаем сумму высот трапеций
- Также поддерживаем сумму изменений высот
- Площадь трапеции между двумя событиями x_1 и x_2:
  - $\left(\frac{(x_2-x_1)\cdot\Delta h}{2}+h\right)(x_2-x_1)$
- Делаем для каждого события добавления отрезка в x удаление отрезка в $x+L$
- Максимизируем функцию площади между двумя событиями
  - $\left(\frac{(x-x_1)\cdot\Delta h}{2}+h\right)(x-x_1)$
  - Квадратичная функция от x
  - Максимум в $\frac{-h}{\Delta h}$
- Время работы: $O(n\log n)$

Альтернативное решение

- Есть большие проблемы с точностью
- Поэтому координаты и требуемая точность ответа малы
- Между целым x и $x+1$ функция квадратична
- Можно воспользоваться численным методом для поиска
- Время работы: $O(\max|x_i| \cdot \mathtt{SEARCH})$, где $\mathtt{SEARCH}$ — это время поиска максимума

Вопросы?

Задача B. «Забавные числа»

Задача B. «Забавные числа»

- Автор задачи — Айдар Гизатуллин
- Подготовка задачи — Архипов Денис

Постановка задачи

- Назовем число забавным, если оно само составное, а количество его делителей простое.
- Цель — посчитать количество таких числе на отрезке $[L \dots R]$.

Решение задачи

- Рассмотрим число N. Как посчитать его количество делителей?
- Представим N в виде разложения на простые делители: $p_1^{\alpha_1} \times p_2^{\alpha_2} \times \ldots \times p_k^{\alpha_k}$.
- Теперь количество делителей — это $(\alpha_1 + 1) \times (\alpha_2 + 1) \times \ldots \times (\alpha_k + 1)$.
- Нас интересуют только числа вида: p^{q-1}. Где p и q — простые числа.
- Переберем все p, возведем его в степень $q - 1$ и проверим, что получившееся число лежит в $[L \ldots R]$.

Задача C. «Гоша и праздники»

Задача C. «Гоша и праздники»

- Автор задачи — Нияз Нигматуллин, Ильнар Сабирзянов
- Подготовка задачи — Дмитрий Якутов

Постановка задачи

- Есть взвешенное дерево и события в вершинах.
- Нужно найти максимальное возможное число посещенных событий.

Решение задачи

- Разобьем решение задачи на две части.
- Первая часть — нахождение расстояний между вершинами.
- Вторая — поиск ответа.

Решение первой части за $O(n^2)$

- Заданный граф — дерево, между каждой парой вершин есть ровно один простой путь.
- Этот простой путь и будет кратчайшим.
- Запустим обход в глубину или в ширину от каждой вершины.

Вторая часть за $O(m!)$

- Переберем все последовательности посещения событий.
- Проверим, что возможно перейти от каждого события до следующего.
- Обновим ответ длиной последовательности.

Вторая часть за $O(n \cdot MaxTime)$

- Храним `dp[ver][time]` — максимальное число посещенных событий.
- Делаем переход через всех соседей вершины.

Вторая часть за $O(m^2)$

- Храним dp[event] — максимальное число посещенных событий.
- Делаем переход через все события с меньшим временем.

Решение первой части за $O(n \cdot log(n))$

- Выбираем любую вершину дерева и называем её корнем.
- Храним длину пути от корня до каждой вершины.
- Разбиваем путь от вершины a до вершины b их `lca`.

Вторая часть за $O(m \cdot log(m) \cdot log(n))$

- Воспользуемся той же идеей, что в решении за $O(m^2)$.
- Применим идею «разделяй-и-властвуй».

Вторая часть за $O(m \cdot log(m) \cdot log(n))$

- Хотим обновить `dp[ev]` через `prevEv`.
- Пусть v — вершина события `ev`, u — события `prevEv`, x — вершина, разделяющая v и u.
- Можем обновить `ev` через `prevEv`, если $time(prevEv) + dist(u, v) \leq time(ev)$

Вторая часть за $O(m \cdot log(m) \cdot log(n))$

- $time(prevEv) + dist(u, v) \leq time(ev)$
- $time(prevEv) + dist(u, x) + dist(x, v) \leq time(ev)$
- $time(prevEv) + dist(u, x) \leq time(ev) - dist(x, v)$

Вторая часть за $O(m \cdot log(m) \cdot log(n))$

- $time(prevEv) + dist(u, x) \leq time(ev) - dist(x, v)$
- Для каждой вершины x сохраним в массиве `undex[x]` все события e, которые были в дереве в момент, когда x была разделяющей, в порядке увеличения величины `time(e) + dist(x, y)`, где y — вершина, в которой проходит событие e.
- Переберем событие `ev`, переберем разделяющую вершину x, надо обновить ответ на некотором префиксе массива `under[x]`

Вторая часть за O($m \cdot log(m) \cdot log(n)$)

- Длину префикса массива `under[x]` находим двоичным поиском
- На массивах `under[x]` храним структуру данных, поддерживающую операции `RMQ`
- Обновим значения элементов массивов `under[x]` числом `dp[ev]`

Задача D. «Д.Р.У.З.Ь.Я.»

Задача D. «Д.Р.У.З.Ь.Я.»

- Автор задачи — Максим Корчагин
- Подготовка задачи — Айдар Гизатуллин

Постановка задачи

- 2 отсортированных массива моментов времени, в которые звонят 2 друга.
- Узнать какое минимальное и максимальное количество гостей из n друзей может придти на каждую из вечеринок.

Решение задачи

- Все друзья равнозначны.
- В любой момент времени для каждого друга: никто не звонил, звонил Амир, Булат или оба.
- Если Амир пригласил максимальное количество, то Булат — минимальное. Аналогично наоборот.

Решение задачи

- Находим максимум для Амира и минимум для Булата.
- Рассматриваем моменты времени в порядке возрастания.
- Амир всегда звонит тому, кого еще никто не пригласил.
- Булат должен звонить тому, кого уже пригласил Амир, иначе неприглашенному.

Решение задачи

- Находим максимум для Амира и минимум для Булата.
- Рассматриваем моменты времени в порядке возрастания.
- Амир всегда звонит тому, кого еще никто не пригласил.
- Булат должен звонить тому, кого уже пригласил Амир, иначе неприглашенному.

Вопросы?

Вопросы?

Камень, ножницы, бумага – 2

Постановка задачи

Два игрока играют в камень, ножницы, бумагу. Необходимо определить минимальное количество раз, когда выигрывает первый, если известно, что первый поставил камень r_1 раз, ножницы – s_1 раз, бумагу – p_1 раз.

Второй – r_2, s_2 и p_2 раз соответственно.

Идея решения

Представим результаты игры в виде такой таблицы.

	Камень	Ножницы	Бумага
Камень	a_1	a_2	a_3
Ножницы	a_4	a_5	a_6
Бумага	a_7	a_8	a_9

Идея решения

Тогда на сумму значений в строках и столбцах накладываются следующие ограничения:

	Камень	Ножницы	Бумага	Сумма
Камень	a_1	a_2	a_3	r_2
Ножницы	a_4	a_5	a_6	s_2
Бумага	a_7	a_8	a_9	p_2
Сумма	r_1	s_1	p_1	

Идея решения

Итак, надо минимизировать сумму $a_3 + a_4 + a_8$

Для удобства переобозначим столбцы.

	Бумага	Камень	Ножницы	Сумма
Камень	a_3	a_1	a_2	r_2
Ножницы	a_6	a_4	a_5	s_2
Бумага	a_9	a_7	a_8	p_2
Сумма	p_1	r_1	s_1	

Идея решения

Заметим, что можно взять любую пару столбцов и строк и сделать следующее преобразование:

	Бумага	Камень	Ножницы	Сумма
Камень	a_3^{+1}	a_1^{-1}	a_2	r_2
Ножницы	a_6^{-1}	a_4^{+1}	a_5	s_2
Бумага	a_9	a_7	a_8	p_2
Сумма	p_1	r_1	s_1	

Идея решения

Так можно делать, если все значения остаются неотрицательными. Обнулим все значения на диагонали, кроме одного.

	Бумага	Камень	Ножницы	Сумма
Камень	a_3 -1	a_1 +1	a_2	r_2
Ножницы	a_6 +1	a_4 -1	a_5	s_2
Бумага	a_9	a_7	a_8	p_2
Сумма	p_1	r_1	s_1	

	Бумага	Камень	Ножницы	Сумма
Камень	a_3 -1	a_1	a_2 +1	r_2
Ножницы	a_6	a_4	a_5	s_2
Бумага	a_9 +1	a_7	a_8 -1	p_2
Сумма	p_1	r_1	s_1	

	Бумага	Камень	Ножницы	Сумма
Камень	a_3	a_1	a_2	r_2
Ножницы	a_6	a_4 -1	a_5 +1	s_2
Бумага	a_9	a_7 +1	a_8 -1	p_2
Сумма	p_1	r_1	s_1	

Идея решения

Пусть мы решили оставить ненулевым a_3

Попробуем сократить и его.

	Бумага	Камень	Ножницы	Сумма
Камень	b_3 -1	b_1	b_2 +1	r_2
Ножницы	b_6 +1	0	b_5 -1	s_2
Бумага	b_9	b_7	0	p_2
Сумма	p_1	r_1	s_1	

	Бумага	Камень	Ножницы	Сумма
Камень	b_3 -1	b_1 +1	b_2	r_2
Ножницы	b_6	0	b_5	s_2
Бумага	b_9 +1	b_7 -1	0	p_2
Сумма	p_1	r_1	s_1	

Идея решения

$c_3 = p_1 - s_2 - p_2$. Это минимальное количество раз, когда бумага первого побеждает камень второго.

	Бумага	Камень	Ножницы	Сумма
Камень	c_3	c_1	c_2	r_2
Ножницы	c_6	0	0	s_2
Бумага	c_9	0	0	p_2
Сумма	p_1	r_1	s_1	

Решение

Заметим, что среди всех диагональных элементов в ходе таких преобразований мы могли выбрать, какой из элементов оставить ненулевым.

Тогда ответом к задаче будет

$$max\begin{Bmatrix} p_1 - p_2 - s_2 \\ r_1 - r_2 - p_2 \\ s_1 - s_2 - r_2 \\ 0 \end{Bmatrix}$$

Альтернативное решение

Чтобы совсем не думать над задачей, можно написать алгоритм нахождения максимального потока минимальной стоимости на следующем графе: