Дистанционные туры, осень 2016/17
Разбор тура #1

Собрано 3 ноября 2016 г. в 00:05

Содержание

A. Деляга Джек: динамика

Пусть изначально мы владели единицей исходного товара. Заметим, что выгодно всегда менять сразу всё, поэтому в любой момент времени у нас товар лишь одного типа.

Динамика. Состояние: сколько ходов мы сделали, каким товаром сейчас владеем. Функция: максимальное возможное количество товара.

Время работы $\mathcal{O}(mk)$, где m – число предложений, $k = 9$ – число транзакций.

B. Провода: 2D-дерево

Во-первых, заметим, что процесс происходит не на окружности, а на прямой. Дуга – отрезок. Теперь нам нужно научиться хранить множество пар (l_i, r_i) и по новой паре (l_*, r_*) говорить, количество $i\colon (l_* < l_i < r_* < r_i) \vee (l_i < l_* < r_i < r_*)$.

Это два $2D$-запроса, количество точек (l_i, r_i) в прямоугольнике. Решение за $\mathcal{O}(\log^2 n)$ времени на запрос: дерево отрезков декартовых деревьев. Если сделать сжатие координат, то можно также написать более быстрое “дерево Фенвика деревьев Фенвика”.

C. Демократия: динамика по дереву

Краткое условие: есть дерево и k цветов, выбрать по одной вершине каждого цвета, добавить и удалить минимальное число рёбер, чтобы выбранные k вершин образовывали компоненту связности. То есть, были связны, и с ними не было связно ничего лишнего.

Решение. Динамика по дереву. Состояние:

1. вершина дерева v,
2. взяли ли мы вершину v в ответ,
3. подмножества A выбранных в поддереве вершины v цветов.

Итого $1000 \cdot 2 \cdot 2^{10}$ состояний.
Функция – минимальное количество добавлений и удалений рёбер в поддереве v.

Пересчёт динамики. Пусть дети v уже посчитаны, чтобы посчитать слой динамики $f[v, 0]$, сделаем внутреннюю динамику по детям v. Суммарное количество переходов во всех внутренних динамиках равно числу рёбер дерева, то есть, $n - 1$. Внутренняя динамика: учли первых i детей, набрали множество цветов A, заплатили $g_v[i, A]$. Пересчёт. i-го ребёнка обозначим x_i. Переберём множество цветов $B \subseteq A$, которое покрыто именно в поддереве x_i.

$$g_v[i, A] = \min_{B \subseteq A}(g[v, i-1, A \setminus B] + \min(f[x_i, 0, B], f[x_i, 1, B] + 1) + 1$$

Плюс единицы обозначают удаление и добавление ребра соответственно. Слой динамики $f[v, 1]$ пересчитывается аналогичной внутренней динамикой. Количество пар $\langle A, B \rangle$ равно 3^k, поэтому суммарное время работы $n \cdot 3^k \leq 60\,000\,000$.

D. Peep Blue: Гранди + Жадность + Гаусс

Пусть после хода первого игрока остались кучки размера $a_1, a_2, \ldots, a_n$.

Второй игрок, чтобы выиграть должен оставить кучки, `XOR` которых равен нолю. Это будет невозможно сделать, если после хода первого из кучек нельзя будет выбрать непустое подмножество с нулевым `XOR`, то есть, двоичные вектора, соответствующие числам a_i линейно независимыми.

Как теперь первому игроку выбрать набор линейно независимых векторов максимального веса? Поскольку $\forall k \quad k$-мерные вектора образуют матроид, работает жадный алгоритм: отсортировать вектора по убыванию веса и в таком порядке пробовать жадно добавлять в ответ.

Уже добавленные в ответ вектора являются строками матрицы. Чтобы добавить очередной вектор, нужно поддерживать эту матрицу методом Гаусса в трапецевидном виде. Тогда добавление одного k-мерного вектора работает за $\mathcal{O}(k^2)$. Итого время работы решения $\mathcal{O}(nk^2 + n \log n)$, где $n \log n$ – время сортировки.чтоб В данной задаче $k = 30$.