Расстановка тыкв

Автор разбора: Даниил Орешников

Будем решать задачу методом динамического программирования. Пусть $\texttt{dp}[i]$ — «оптимальный», то есть дающий максимальное значение удовлетворенности, способ расставить тыквы на первых i местах, при котором на месте номер i **обязательно стоит тыква**. Переберем предыдущее место j, на котором стоит тыква. Для фиксированного j формула пересчета удовлетворенности будет иметь вид

$$\texttt{dp}[i] = \texttt{dp}[j] - c_i + \sum_{t=1}^{n} |x_i - x_j - d_t|.$$

Заметим, что максимизация этой величины равносильна минимизации $\texttt{dp}[j] + \sum\limits_{t=1}^{n} |x_i - x_j - d_t|$. Определим $f_j(x)$ как $\texttt{dp}[j] + \sum\limits_{t=1}^{n} |x - x_j - d_t|$. Тогда для фиксированного x_i мы ищем $\max\limits_{j<i} f_j(x_i)$. Заметим, что все f_j имеют не более одной точки пересечения каждый с каждым, так как

- каждая из них является суммой модулей линейных функций;
- модуль линейной функции выпуклый вверх, поэтому сумма таких модулей тоже выпукла вверх;
- график f_j является параллельным переносом f_1 на $x_j - x_1$ по оси X и на $\texttt{dp}[j] - \texttt{dp}[1]$ по оси Y.

Выпуклые вверх функции, получающиеся друг из друга параллельным переносом, имеют не более одной точки пересечения, таким образом $\max\limits_{j} f_j(x)$ имеет не более n участков, на которых максимальной является конкретная f_j. Учитывая все эти условия, можно было применить дерево Ли Чао, чтобы при пересчете $\texttt{dp}[i]$ найти $\max\limits_{j} f_j(x_i)$, а затем добавить f_i в рассмотрение.

Чтобы быстро вычислять $\sum\limits_{t=1}^{n} |x - x_j - d_t|$, достаточно в самом начале отсортировать все d_t по возрастанию и посчитать их префиксные суммы. Найдем бинпоиском первое d_t, большее, чем $x - x_j$, после чего раскроем все модули для меньших d со знаком плюс, а все остальные — со знаком минус. Получившееся выражение можно будет легко посчитать с помощью префиксных сумм d_t.

Поскольку в задаче требуется найти оптимальную расстановку, в которой первое и последнее место заняты тыквами, достаточно начинать с $\texttt{dp}[1] = -c_1$ и взять в качестве ответа $\texttt{dp}[n]$.