Лекция 13. Динамическое программирование II.

#вшпи #аисд #дп

Автор конспекта: Гридчин Михаил

ДП по подотрезкам

> **Принцип ДП по подотрезкам**
>
> **Если имеется способ расчёта ответа на задачу на основе ответов на подотрезках, то выгодно применить ДП по подотрезкам.**

То есть для решения такой задачи надо сделать следующее:

- $\mathrm{dp}[l, r]$ - ответ на задачу на подотрезке $[l, r]$.
- Задать базу для отрезков длины $1, 2$.
- Определить функцию "слияния" ответов. То есть как получить из результата на подотрезках результат для всего отрезка.

Данная постановка предполагает, что ответ для отрезка $[l, r]$ можно извлечь из отрезков $[l+1, r], [l, r-1], [l+1, r-1]$.

Всего подотрезков $O(n^2)$ и пересчёт ведётся по увеличению длины подотрезка, то есть по увеличению $(r-l)$.

Пример: последовательность-палиндром

Дана строка S. Требуется найти длину максимальной подпоследовательности-палиндрома этой строки за $O(|S|^2)$.

Решение. Заведём следующую динамику:
Состояние динамики. $f(l, r)$ - ответ для строки $S[l : r]$
База. $dp[i][i] = 1$ и $dp[l][r] = 0$, где $r < l$.
Пересчёт. Возможны два случая:

- $S[l] = S[r]$, тогда $f(l, r) = f(l+1, r-1) + 2$.
- $S[l] \neq S[r]$, тогда $f(l, r) = \max\{f(l+1, r), f(l, r-1)\}$.
  Порядок пересчёта. Пересчёт по увеличению длины, то есть по увеличению $(r-l)$.
  Ответ. $f(1, n)$.

Кубическое ДП по подотрезкам

Данная постановка предполагает, что ответ для отрезка $[l, r]$ можно извлечь из отрезков $[l, m], [m+1, r]$, где $m \in [l, r]$.

Пример: длина максимальной скобочной подпоследовательности

Дана строка S из разных типов скобок. Требуется найти длину максимальной правильной скобочной подпоследовательности (ПСП) за $O(|S|^3)$.

Решение. Заведём следующую динамику:
Состояние динамики. $\mathrm{dp}[l][r]$ - минимальное число символов, которые надо удалить из $S[l : r+1]$, чтобы получилась ПСП.
База. $\mathrm{dp}[i][i] = 1, \mathrm{dp}[l][r] = 0$, где $r < l$.
Пересчёт. Возможны два случая, что может произойти со скобкой $S[l]$:

- Скобка $S[l]$ не войдёт в ответ, тогда она удалена, то есть $\mathrm{dp}[l][r] = \mathrm{dp}[l+1][r] + 1$.
- Скобка $S[l]$ войдёт в ответ, тогда найдётся парная ей скобка на позиции $m \in [l+1, r]$. Тогда

$$\mathrm{dp}[l][r] = \min_{m \in [l+1, r]} \mathrm{dp}[l+1][m-1] + \mathrm{dp}[m+1][r]$$

  Порядок пересчёта. Пересчёт по увеличению длины, то есть по увеличению $(r - l)$.
  Ответ. $dp[0][|S|-1]$.

Матричное ДП

> **Определение**
>
> **Линейным рекуррентным соотношением с постоянными коэффициентами (ЛРСПК) порядка k назовём соотношение вида**
>
> $$a_n = \sum_{i=1}^{k} \lambda_i a_{n-i} + f(n, a_{n-1}, \ldots, a_{n-k})$$
>
> **С заданными $a_1, \ldots, a_k$.**
> **ЛРСПК называется однородным, если $f \equiv 0$. То есть**
>
> $$a_n = \sum_{i=1}^{k} \lambda_i a_{n-1}$$

Пример. числа Фибоначчи. Для чисел Фибоначчи выполнено равенство

$$F_n = F_{n-1} + F_{n-2}$$

Это однородное ЛРСПК второго порядка.

Теоремы о решениях ЛРСПК (б/д)

Рассмотрим однородное ЛРСПК порядка k:

$$a_n = \sum_{i=1}^{k} \lambda_i a_{n-i}$$

Рассмотрим **характеристический многочлен**

$$x^n = \sum_{i=1}^{k} \lambda_i x^{n-i}$$

Пусть у него k различных корней $\gamma_1, \dots, \gamma_k$, тогда

$$\boxed{a_n = \sum_{i=1}^{k} C_i \gamma_i^n}$$

Где константы $C_1, \dots, C_k$ вычисляются из первых k значений рекурренты.

Пример. Рассмотрим однородное ЛРСПК второго порядка $a_n = \lambda_1 a_{n-1} + \lambda_2 a_{n-2}$. Рассмотрим корни γ_1, γ_2 характеристического многочлена $x^2 = \lambda_1 x + \lambda_2$. Допустим, что $\gamma_i \in \mathbb{R}$, тогда

$$a_n = \begin{cases} C_1\gamma_1^n + C_2\gamma_2^n & \gamma_1 \neq \gamma_2 \\ (C_1 + C_2 n)\gamma_1^n & \text{иначе} \end{cases}$$

Числа Фибоначчи как матрицы

Рассмотрим числа Фибоначчи $F_n = F_{n-1} + F_{n-2}$. В этот раз хотим быстро найти n-е число. Рассмотрим соотношение:

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix} = \dots = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \cdot \begin{pmatrix} F_1 \\ F_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Воспользуемся бинарным возведением в степень и получим n-е число Фибоначчи за $O(\log n)$

Однородное ЛРСПК

Рассмотрим однородное ЛРСПК порядка k:

$$a_n = \sum_{i=1}^{k} \lambda_i a_{n-i}$$

Запишем матрицу для решения такой рекурренты:

$$\begin{pmatrix} a_n \\ a_{n-1} \\ a_{n-2} \\ \vdots \\ a_{n-k+1} \end{pmatrix} = \begin{pmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_{k-1} & \lambda_k \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_{n-1} \\ a_{n-2} \\ a_{n-3} \\ \vdots \\ a_{n-k} \end{pmatrix} = \begin{pmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_{k-1} & \lambda_k \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}^{n-k+1} \cdot \begin{pmatrix} a_{k-1} \\ a_{k-2} \\ a_{k-3} \\ \vdots \\ a_0 \end{pmatrix}$$

Получим значение n члена рекурренты за $O(k^3 \log n)$ двоичным возведением матрицы в степень.

ДП по подмножествам

Рассмотрим работу с двоичными числами длины N. Доступны следующие операции:

- Побитовое ИЛИ. $\mathrm{a} \mid \mathrm{b}$
- Побитовое И. $\mathrm{a}\ \&\ \mathrm{b}$
- Побитовое ВЗАИМОИСКЛЮЧАЮЩЕЕ ИЛИ. $\mathrm{a} \wedge \mathrm{b}$
- Побитовая ИНВЕРСИЯ. $\sim \mathrm{a}$
- Побитовый СДВИГ влево или вправо. $\mathrm{a} << 1$ или $\mathrm{a} >> 1$

Рассмотрим множество $\{0, 1, \dots, n-1\}$. Тогда можно закодировать все возможные подмножества двоичной строки длины n или числами до 2^n. Если элемент i в подмножестве, то вместо i-го бита будем ставить единицу, иначе ноль.

Пусть A, B — подмножества $\{C_0, \dots, C_{n-1}\}$. Тогда

- $A \cup B = A \mid B$.
- $A \cap B = A\ \&\ B$
- $\bar{A} = \ \sim A$

Пример: задача о рюкзаке

Пусть $\mathrm{dp[mask]} = -\infty$, если суммарный вес предметов из mask больше W и $\sum_{i \in mask} c_i$ — стоимость предметов в mask. Будем расширять mask по одному предмету. Допустим $i \notin mask$, тогда

$$\mathrm{dp[mask} \mid (1 << i)] = \begin{cases} -\infty, & \text{если } w(\mathrm{mask}) + w_i > W \\ \mathrm{dp[mask]} + c_i \end{cases}$$

Пример: задача коммивояжёра

Дан полный граф на N вершинах. Для каждого ребра известен вес w_{ij}. Требуется найти путь минимального веса, проходящий по каждой вершине ровно один раз.

Решение. Построим следующую динамику:
Состояние динамики. $\mathrm{dp}[\mathrm{mask}][v]$ - минимальный вес обхода вершин из mask, при этом v — последняя вершина.
База. $\mathrm{dp}[1 << v][v] = 0$, остальные равны бесконечности.
Пересчёт.

$$\mathrm{dp}[\mathrm{mask} \mid (1 << u)][u] = \min\{\mathrm{dp}[\mathrm{mask} \mid (1 << u)][u], \min_{v \in \mathrm{mask}} \{\mathrm{dp}[\mathrm{mask}][v] + w(v, u)\}\}$$

Порядок пересчёта. По увеличению mask, перебирая u и v в цикле.
Ответ.

$$\min_{v} \mathrm{dp}[2^n - 1][v]$$

ДП по профилю

Задача "Симпатичные узоры"

Рассматривается прямоугольное поле размера $N \times M$. Каждая клетка может быть окрашена в один из двух цветов. Узор называется симпатичным, если он не содержит одноцветного квадрата размера 2×2. Требуется определить количество таких узоров.

Решение. Зафиксируем нумерацию столбцов от 1 до M. Представим один столбец в виде битовой маски длины N, где $0/1$ соответствует цвету клетки. Пусть $\mathrm{dp}[m][i]$ - это количество симпатичных узоров на первых i столбцах, если i-й столбец имеет маску m. Переход строится следующим образом: пусть m_1 - маска i-го столбца, а m_2 - маска $(i+1)$ -го. Маски m_1 и m_2 совместимы (обозначим это условие как $\mathrm{ok}(m_1, m_2) = 1$), если они не образуют одноцветного квадрата 2×2, то есть для любого $1 \le j < N$ не выполняется одновременно

$$m_1[j] = m_1[j+1] = m_2[j] = m_2[j+1]$$

Если совместимость выполнена, возможен переход:

$$\mathrm{dp}[m_2][i+1] += \mathrm{dp}[m_1][i]$$

Тогда переберём маску i столбца (2^N итераций). Для каждой маски проверим, образуют ли они одноцветный квадрат. Если маска new_mask не образует одноцветный квадрат с $mask$, то

$$\mathrm{dp}[\mathrm{new\_mask}][i+1] += dp[\mathrm{mask}][i]$$

Ответ можно найти, как

$$\sum_{i=0}^{2^N} dp[i][M]$$

Асимптотика решения $O(N^2 \times 4^N)$.
Можно улучшить, если изначально для каждой пары масок предпосчитать, образуют ли они квадрат 2×2.

$$\mathrm{dp}[\mathrm{mask}_2][i] = \sum_{\mathrm{mask}_1=0}^{2^N-1} dp[mask_1][i] \cdot I(ok(mask1, mask2))$$

Пусть $A_i[m]$ - ответ, если i столбцов и i-й имеет маску m. Ответ:

$$\sum_{m=0}^{2^N-1} A_M[m]$$

Тогда $A_i = BA_{i-1}$, где $B[m_1][m_2] = I(ok(m_1, m_2))$. Тогда $A_M = B^{M-1} \cdot A_0$. Время работы $O(8^n \log m)$.