

Московский физико-технический институт
Физтех-школа прикладной математики и информатики

БОЛЬШОЕ НАЗВАНИЕ КУРСА
V СЕМЕСТР

Лектор: *Иван Иванович Иванов*

h/ν

Автор: *Павел Дуров*
Репозиторий на Github

осень 2022

Содержание

1	Продолжение ДП	2
1.1	Рюкзак с оптимизацией	2
1.1.1	Альтернативный вариант	2
1.2	Динамическое программирование с помощью матриц	2
1.3	Задача	3
1.4	Задача	3
1.5	Задача	4

1 Продолжение ДП

1.1 Рюкзак с оптимизацией

→ w предметов

→ w_i - вес

→ c_i - стоимость

Решение мы помним с прошлой лекции, а сейчас займемся оптимизацией памяти:

$dp[i][o]$ зависят только от $dp[i-1][o]$, поэтому нам достаточно хранить не всю таблицу целиком, а всего 2 слоя, с которыми мы работаем.

Итог - память $O(W)$

1.1.1 Альтернативный вариант

Будем действовать от стоимости предметов:

1. Заводим массив $dp[0 \dots n-1][0 \dots C-1]$, где $C = \sum_{i=0}^{n-1} c_i$
2. $dp'[i][b]$ - min суммарный вес предметов, имеющих номера $\leq i$, и общую стоимость b
3. $dp'[i][b] = \min(dp[i-1][b], dp'[i-1][b-c_i] + w_i)$

Это используется, если суммарная стоимость значительно меньше суммарного веса.

1.2 Динамическое программирование с помощью матриц

Попробуем найти $F_n = F_{n-1} + F_{n-2}$ с методом матриц.

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} F_{n-2} \\ F_{n-3} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \cdot \begin{pmatrix} F_1 \\ F_0 \end{pmatrix}$$

БИНАРНОЕ ВОЗВЕДЕНИЕ МАТРИЦЫ В СТЕПЕНЬ

Проводится так же, как и для натуральных чисел:

$$a^n = \begin{cases} 1, & \text{если } n = 0 \\ \left(a^{\frac{n}{2}}\right)^2, & \text{если } n \text{ четно} \\ a \cdot a^{n-1}, & \text{если } n \text{ нечетно} \end{cases} \quad (1.1)$$

Если две матрицы имеют размеры $k \times k$, то их произведение можно найти за $O(k^3)$

Тогда A^n описанным алгоритмом находится за $O(k^3 \log n)$

Задача $a_n = \lambda a_{n-1} + \mu a_{n-2} + 1$

$$\begin{pmatrix} a_n \\ a_{n-1} \\ 1 \end{pmatrix} = \begin{pmatrix} \lambda & \mu & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_{n-1} \\ a_{n-2} \\ 1 \end{pmatrix}$$

Взяв произведение этих матриц, получим ответ за $O(3^3 \log n)$

Задача Пусть G – невзвешенный ориентированный граф. Найти количество путей длины ровно k из вершины x в вершину y

На ввод дается матрица смежности M , где $m_{ij} = 1 \Leftrightarrow$ есть ребро $i \rightarrow j$, а 0 иначе.

Пусть $dp[v][l]$ – количество путей длины l от x до v .

Тогда $dp[v][l] = \sum_{u \in v: M_{uv}=1} dp[u][l-1]$

$$\begin{pmatrix} dp[1][l] \\ dp[2][l] \\ \vdots \\ dp[v][l] \\ \vdots \\ dp[n][l] \end{pmatrix} = M^T \cdot \begin{pmatrix} dp[1][l-1] \\ dp[2][l-1] \\ \vdots \\ dp[v][l-1] \\ \vdots \\ dp[n][l-1] \end{pmatrix}$$

Комментарий от эксперта:

Пересчет динамики получается домножением столбца $dp[v][i-1]$ на транспонированную матрицу смежности слева

Утверждение 1.1. M^k – количество путей из u в v длины ровно k .

1.3 Задача

Найти количество путей длины $\leq k$ из x в y .

Можно найти ответ из суммы $(M^0 + M^1 + \dots + M^k)_{xy}$, но как ее посчитать быстро?

Введем $f(M, k) = (M^k, M^0 + M^1 + \dots + M^{k-1})$.

1. $k = 0 \rightarrow f(M, k) = (E, E)$
2. $k \neq 2 \rightarrow f(M, k-1) = (M^{k-1}, M^0 + M^1 + \dots + M^{k-2})$, откуда $f(M, k) = f(M, k-1)$, в котором умножили первый элемент на M , предварительно прибавив его ко второму.
3. $k \geq 2$, $f(M, k)$ получается из $f(M, \frac{k}{2})$ умножением первой части, увеличенной на 1, на вторую и возведением первой части в квадрат.

ВТОРОЙ КОММЕНТАРИЙ ОТ ЭКСПЕРТА:

По формуле геометрической прогрессии $\sum_{i=0}^k M^i = (M^{k+1} - E) \cdot (M - E)^{-1}$. Если $M - E$ необратима, подкрутим её коэффициент на 0.00001.

1.4 Задача

Пусть G – граф. Надо проверить, есть ли хотя бы 1 путь из x в y длины ровно k ?

$$d[v][l] = \bigvee_u (dp[u][l-1] \wedge M_{uv})$$

Обозначим $A * B = C$, где $*$ – булево умножение, такое выражение:

$$c_{ij} = \bigvee_k (A_{ik} \wedge B_{kj})$$

Утверждение 1.2. $M_{uv}^{*k} = 1$, если есть путь $u \rightarrow v$, а иначе 0

Такое тоже работает за $O(n^3 \log k)$

1.5 Задача

G - взвешенный граф. Хотим найти \min стоимость пути длины ровно k из x в y .

Пусть $dp[v][l]$ - минимальная стоимость пути $x \rightarrow v$ за l ребер. Тогда его можно найти по формуле $\min(dp[u][l-1] + cost(u, v))$

Обозначим: $A \circ B = C$, где

$$c_{ij} = \min_k (a_{ik} + b_{kj})$$

$$(A \circ B) \circ C = A \circ (B \text{ circ } C)$$

Утверждение 1.3. $M^{\circ k}$ - минимальная стоимость пути из u в v , используя ровно k ребер.