МФТИ, ФПМИ
Алгоритмы и структуры данных, 2-й семестр, весна 2022
Семинар №2. Динамическое программирование (2)

Во всех задачах этого листка, при необходимости, можно считать, что все арифметические операции выполняются за $O(1)$.

1. Решите задачу о рюкзаке в следующих модификациях:
а) i-й предмет можно брать от 0 до cnt_i раз (разрешается добавить в асимптотику зависимость от значений cnt_i);
б) каждый предмет можно брать неограниченное число раз (асимптотика: $O(n \cdot W)$, где W — вместимость рюкзака);
в) от каждого предмета можно отпилить произвольную часть (то есть увеличить общий вес на $\alpha \cdot w_i$, а к стоимости добавить $\alpha \cdot c_i$, где $\alpha \in [0, 1]$). Асимптотика: $O(n \log n)$.

2. Дан тетраэдр и муравей, находящийся в одной из его вершин. За один ход нужно переместиться вдоль любого ребра. Для заданного n за $O(\log n)$ определить количество путей длины n, возвращающих муравья в исходную вершину.

3. Есть слоистый граф из l слоёв, в каждом по n вершин. Из i-го слоя есть все рёбра в $(i+1)$-й, причём вес ребра в j-ю вершину большего слоя не зависит от истока этого ребра, и этот вес не меняется от слоя к слою (этот вес задаётся явным образом). Нужно найти количество путей из первого слоя в последний, сумма весов рёбер в которых кратна M. Асимптотика: $O(nM + M^3 \log l)$.

4. Задана двумерная целочисленная сетка с неотрицательными координатами. Из $(0, 0)$ нужно попасть в $(k, 0)$. Ходить из точки (x, y) можно только в точки $(x + 1, y - 1), (x + 1, y), (x + 1, y + 1)$. Есть n горизонтальных отрезков с ординатой $\leqslant Y$, выше которых нельзя подниматься. Их концы (a_i, b_i) по оси Ox таковы, что $a_1 = 0, b_n = k, a_{i+1} = b_i$. Найти количество валидных путей за $O(n \cdot Y^3 \log k)$.

5. Последовательность $\{a_n\}_{n=0}^{+\infty}$ задана следующими соотношениями: $a_0 = 13$, $a_1 = 8$, а также $a_n = 5a_{n-1} + 2a_{n-2} + n^2$ для всех $n \geqslant 2$. По заданному k найдите a_k за $O(\log k)$.

6. Назовём число *гладким*, если в его десятичной записи абсолютная разность любых двух рядом стоящих цифр не меньше l и не больше r. Дано число n. Сколько существует гладких натуральных чисел, состоящих из n цифр? Асимптотика: $O(\log n)$.

7. Дано подвешенное дерево на n вершинах со взвешенными рёбрами (у каждого ребра есть стоимость). Для каждой вершины v найти самую удалённую вершину в её поддереве. То же для наддерева. Асимптотика: $O(n)$.

8. Задан массив чисел $a_1, \ldots, a_{nt}$ длины $n \cdot t$. Известно, что для любого $i > n$ верно, что $a_i = a_{i-n}$. Найдите длину самой длинной неубывающей подпоследовательности заданного массива за $O(n^3 \log t)$.

1.

а) Для состояния $dp[i][w]$ надо рассмотреть $\min\left(cnt_i, \left\lfloor \frac{w}{w_i} \right\rfloor\right)$ значений из $dp[i-1][\cdot]$. Какие?

б) В этом пункте можно обратить внимание на значения не из $dp[i-1][\cdot]$, а из $dp[i][\cdot]$.

в) Для этого пункта не нужно динамическое программирование.

2. Сколько существует способов за k шагов переместиться из исходной вершины в j-ю?

3. Сколькими способами можно добраться до i-го слоя, набрав остаток r по модулю M? Примените идею матричного умножения.

4. Нужно n раз использование бинарное возведение в степень.

5. В столбец пересчёта добавьте $1, n, n^2$.

6. Вспомните решение задачи с помощью обычного ДП. Примените идею матричного умножения.

7. Пусть известен ответ в поддереве каждого из сыновей вершины v. Куда выгоднее всего пойти из v? Задача для наддерева решается через задачу о поддереве.

8. Используйте идею max-plus умножения. Введите операцию умножения матриц:

$$(A \odot B)_{ij} = \max_k (a_{ik} + b_{kj})$$

Массив имеет блочный вид: его элементы равны $a_1, \ldots, a_n, a_1, \ldots, a_n, a_1, \ldots, a_n$ и так далее.
Определим $dp(i \to j)$ как длину ННП (наибольшей неубывающей подпоследовательности), которая начинается в a_i, не содержит ни одного из чисел $a_{i+1}, \ldots, a_n$ и заканчивается в a_{n+j}. При этом начальное a_i в длине не учитывается. Иными словами, если есть всего два блока, рассматривается ННП, которая содержит единственный элемент a_i из первого блока и заканчивается элементом a_j из второго блока. Эту динамику можно найти за $O(n^3)$:

$$dp(i \to j) = \begin{cases} 0, & \text{если } a_i > a_j \\ \max\left(1, \max\limits_{\substack{k \\ k<j \\ a_k \leqslant a_j}} dp(i \to k) + 1\right), & \text{иначе} \end{cases}$$

Введём теперь dp_i^k – длина ННП, которая заканчивается в элементе $a_{n(k-1)+i}$, то есть в k-м блоке в i-м числе. Набор dp_i^1 по всем i легко найти за $O(n^2)$ (просто найти ННП в массиве длины n).
А значения $(k+1)$-го слоя выражаются через значения k-го слоя следующим образом:

$$\begin{pmatrix} dp_1^{k+1} \\ dp_2^{k+1} \\ \ldots \\ dp_n^{k+1} \end{pmatrix} = \begin{pmatrix} dp(1 \to 1) & dp(2 \to 1) & \ldots & dp(n \to 1) \\ dp(1 \to 2) & dp(2 \to 2) & \ldots & dp(n \to 2) \\ & & \ldots & \\ dp(1 \to n) & dp(2 \to n) & \ldots & dp(n \to n) \end{pmatrix} \odot \begin{pmatrix} dp_1^k \\ dp_2^k \\ \ldots \\ dp_n^k \end{pmatrix}$$

Остаётся возвести матрицу $n \times n$ в степень t.