ДП по отрезкам

Задача. Пусть дана строка S. Найдите длину макс. подпосл-ти палиндрома за $O(|S|^2)$

□ Заведем $dp[l][r]$ - ответ для задачи на $[l, r]$. База: $\forall l \in \{1 \dots n\}$ $dp[l][l] = 1$, т.к. 1 символ сам по себе палиндром длины 1. $\forall r < l$ положим $dp[l][r] = 0$, т.к. некорректные границы не должны влиять на ответ.

Расширяем динамику: dp для отрезка $[l, r]$

- $S[l] = S[r]$? мы можем расширить слева и справа палиндром полученный в подотрезке $[l+1, r-1]$

$$dp[l][r] = dp[l+1][r-1] + 2$$

- $S[l] \neq S[r]$? Тогда мы не можем расширить палиндром подстроки $[l+1, r-1]$. Возьмем наиб. из уже сущ-х. Их можно взять из $[l, r-1]$ и $[l+1, r]$.

$$dp[l][r] = \max\{dp[l][r-1], dp[l+1][r]\}$$

Core Idea: Чтобы пересчитать отрезки длины L надо знать ответы для отрезков длины меньше L.

Example

S = „ACBBA", n = 5

Step 1. Initialize base cases.

- dp[1][1] = dp[2][2] ... = dp[5][5] = 1
- ∀r ∃l: r<l ⟹ dp[l][r] = 0

Step 2. Substrings of length 2

l = 1, r = 2 S[1] = „A", S[2] = „C"

=> dp[1][2] = max(dp[1][1], dp[1,1]) = 1.

The same for (l=2, r=3) dp = 1, (l=3, r=4) dp = 2, (l=4, r=5). dp[3][4] = 2

Step 3. Substrings of lenght 3.

l = 1, r = 3 → dp[1][3] = max{ dp[1][2], dp[2][3] } = 1

l = 2, r = 4 → dp[2][4] = max{ dp[2,3], dp[3,4] } = 2

- dp[3][5] = 2

Step 4. (L = 4)

- dp[1][4] = max(dp[1][3], dp[2,4]} = 2
- dp[2, 5] = 2

Step 5. (L = 5)

dp[1][5] = max{dp[2,5], dp[1,4]} = 2.

Step 6

The answer:

dp[1][n]

```
def PALINDROM(S):
    n = len(S)
    dp = new int[n][n]

    for l in range (1, n+1):        # make dp ready
        dp[l][l] = 1
        r = 1
        while r<l:
            dp[l][r] = 0            # invalid ranges
            r++

    for L in range(2, n):
        for l in range(1, n-L):
            r = l+L-1
            if S[l] == S[r]:
                dp[l][r] = dp[l+1][r-1] + 2
            else:
                dp[l][r] = max(dp[l+1][r],
                               dp[l][r-1])

    return dp[1][n]
```

Маски

кодирование мн-в.

Пусть U - мн-во, $|U|=n$. Тогда $A \subset U$ можно закодировать с пом-ю битов: если i-ый элемент есть в мн-ве, то он вставляется в 1, иначе - в 0.

Def $1 << n := \underbrace{100\ldots0}_{n}$

Задача Дан полный граф на n вершинах. Для $\forall (u,v)$ известна стоимость $c(u,v)$. Надо посетить все вершины 1 раз за мин. стоимость.

□ Зафиксируем мн-во вершин V. Будем строить двухмерную динамику, $dp[V'][v]$ - мин. стоимость обхода вершин у V', а v - посл. вершина в обходе. В д. случае V' - число длины n, которое представляет собой маску, постр. на основе рассуждений ↑.

База: $\forall v \in V \hookrightarrow dp[\{v\}][v]=0$.

Для остальных ячеек $+\infty$.

$$\boxed{dp[1<<v][v]=0}$$

$$dp[V' \cup \{v\}][v] = \min\Big\{dp[V' \cup \{v\}][v],\ \min_{v' \in V'}\{dp[V'][v'] + c(v',v)\}\Big\}$$

$\{v\} = 1 << v$ (установить v-й бит в 1). А как объединить? В ответе проставить i-ый бит в 1, если в V или $\{v\}$ он 1. Но это же побитовое ИЛИ!

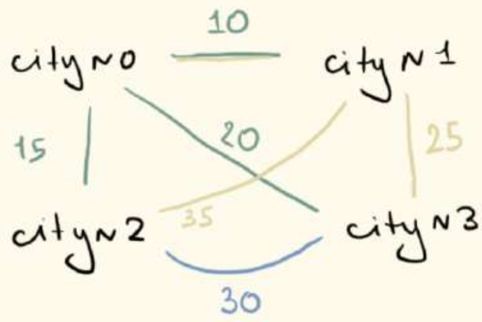
Итого:

$$dp[V' | (1<<v)][v] = \min \begin{cases} dp[V'|(1<<v)][v] \\ \min\limits_{v' \in V'} \{dp[V'][v'] + c(v', v)\} \end{cases}$$

Core Idea: перебираем всевозм. маски для V. А внутри - вершины.

```
def COMM(V):
    n = |V|
    dp = int[2^n][n]
    + БАЗА dp + ...
    for V' ⊆ V:
        for v ∈ 1 ... n:
            if V & (1<<v) = 0:
                for v' ∈ V':
                    if V'&(1<<v) ≠ 0:
                        dp[V'|(1<<v)][v]
                              ||
                        min { dp[V'|(1<<v)][v]
                            { min_{v'∈V'} {dp[V'][v'] + c(v',v)}
    return dp[V][0]
```

Example



mask	visited	0	1	2	3
0001	0	0	·	·	·
0011	0,1	·	10	·	·
0101	0,2	·	·	15	·
1000	0,3	·	·	·	20
0111	0,1,2	·	50	45	·
1011	0,1,3	·	45	·	35
1101	0,2,3	·	·	50	45
1111	0,1,2,3	·	70	65	75

□ - infinity

MEMO

Step 0

dp[0001][0] = 0 (we must start just somewhere)

all other dp[mask][v] = +∞ ∀mask ∀v∈V.

Step 1 Processing mask 0001 (only №0)

From mask = 0001, v = 0:

1) try to go to u = 1: new_mask = 0011

dp[0011][1] = min(∞, 0 + C[0][1] = 10) = 10

2) try to go to u = 2: new_mask = 0101

dp[0101][2] = 15

3) try to go to u = 3: new_mask = 1001

dp[1001][3] = min(∞, 0 + 20) = 20.

Step 3 Processing mask 0011

1) u = 2, 0111

dp[0111][2] = min[∞; 10 + C[1][2]] = 45

2) dp[1011][3] = min(∞; 10 + C[1][3]) = 35

mask	0	1	2	3
0001	0	∞	∞	∞
0011	∞	10	∞	∞
0101	∞	∞	15	∞
1001	∞	∞	∞	20
0111	∞	50	45	∞
1011	∞	45	∞	35
1101	∞	∞	50	45

Step 4 Processing mask 0101 ...

Step 5 Processing mask 1001

Time complexity $O(2^n \cdot n)$

Решение рекуррентных соотношений

Def Всюду и далее РС – рекур. соотн-е.

Def Линейным однородным РС с постоянными коэф-ми k-го порядка будем называть отн-е вида:

$$a_0 y_n + a_1 y_{n-1} + \dots a_k y_{n-k} = 0, \quad a_i \in \mathbb{R}, \ a_0 \neq 0.$$

· Начальное усл-е на ЛОРСПК т.к. $y_0, y_1 \dots y_{k-1} \in \mathbb{R}$.
ЛОРСПК в привычном (???) виде:

$$y_n = \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + \dots + \alpha_k y_{n-k}$$

Заметим, что числа Фибоначчи:

$F_{n+2} - F_{n+1} - F_n = 0$ это ЛОРСПК с

$\alpha_0 = 1, \ \alpha_1 = -1, \ \alpha_2 = -1.$

Опять теория... 

Def Характеристическим многочленом будем называть $x^n = \alpha_1 x^{n-1} + \alpha_2 x^{n-2} + \dots \alpha_k x^{n-k}$.

Theorem Пусть хар многочлен ЛОРСПК 2-го порядка им. 2 реал. корня $\lambda_1 \neq \lambda_2 \in \mathbb{R}$. Тогда $\exists c_1, c_2 \in \mathbb{R}: \forall n \in \mathbb{N} \hookrightarrow y_n = c_1 \lambda_1^n + c_2 \lambda_2^n$

Применим к числам Фибоначчи: характер-е ур-е выглядит след. образом: $x^2 - x - 1 = 0$

$$x = \frac{1 \pm \sqrt{5}}{2}$$ – золотое сеч-е!

$\varphi = \frac{1+\sqrt{5}}{2} \qquad \Phi = \frac{1-\sqrt{5}}{2}$

Statement. $\phi = -\varphi^{-1}$

$\square$ $\phi \cdot (-\varphi^{-1}) = \frac{1-\sqrt{5}}{2} \cdot \frac{(-1-\sqrt{5})}{2} = \frac{-1-\sqrt{5}+\sqrt{5}+5}{4} = \frac{4}{4} = 1$ $\blacksquare$

По утв-ю теоремы $\exists c_1, c_2 \in \mathbb{R}$:

$F_n = c_1\left(\frac{1+\sqrt{5}}{2}\right)^n + c_2\left(\frac{1-\sqrt{5}}{2}\right)^n$. Найдем c_1, c_2:

$$\begin{cases} 0 = F_0 = c_1\left(\frac{1+\sqrt{5}}{2}\right)^0 + c_2\left(\frac{1-\sqrt{5}}{2}\right)^0 = c_1 + c_2 \\ 1 = F_1 = c_1 \frac{1+\sqrt{5}}{2} + c_2 \frac{1-\sqrt{5}}{2} \end{cases}$$

$$\begin{cases} c_1 = -c_2 \\ c_1 \cdot \frac{1+\sqrt{5}}{2} - c_1 \frac{1-\sqrt{5}}{2} = 1 \end{cases} \quad \begin{cases} c_1 = -c_2 \\ c_1 = \frac{1}{\sqrt{5}} \end{cases}$$

$$\begin{cases} c_1 = \frac{1}{\sqrt{5}} \\ c_2 = -\frac{1}{\sqrt{5}} \end{cases}$$

Запишем решение:

$$F_n = \frac{1}{\sqrt{5}}\left(\underbrace{\left(\frac{1+\sqrt{5}}{2}\right)}_{\varphi}^n - \underbrace{\left(\frac{1-\sqrt{5}}{2}\right)}_{-\varphi^{-1}}^n\right)$$

Имеем формулу Бине:

$$F_n = \frac{\varphi - (-\varphi)^{-n}}{\sqrt{5}}$$

$\blacksquare$

Матричный переход к РС

St РС можно представлять в виде матриц.

$$\begin{pmatrix}F_n\\F_{n-1}\end{pmatrix} = \begin{pmatrix}F_{n-1}+F_{n-2}\\F_{n-1}\end{pmatrix}$$

Например:

$$\begin{pmatrix}F_n\\F_{n-1}\end{pmatrix} = \underbrace{\begin{pmatrix}1&1\\1&0\end{pmatrix}}_{2\times 2}\underbrace{\begin{pmatrix}F_{n-1}\\F_{n-2}\end{pmatrix}}_{2\times 1}$$

Тогда итоговая формула n-го члена есть просто возведение в степень матрицы:

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

Можно просто перемножать n раз за $O(n)$.

А можно определить РС: $(O(\log n))$

$$P(A,n) = \begin{cases} E, & n=0\\ A, & n=1\\ P(A,\frac{n}{2})\cdot P(A,\frac{n}{2}), & n\%2=0\\ P(A,\frac{n-1}{2})\cdot P(A,\frac{n-1}{2})\cdot A, & n\%2=1\end{cases}$$

$$A^n = P(A,n)$$

Задача Число гадкое, если модуль разности между $\forall$ двумя соседними цифрами лежит в отрезке $[l, r]$. Сколько сущ-т гадк. чисел длиной n? Время: $O(\log n)$

$$\underset{\text{length}}{dp[m]}\underset{\text{the last digit}}{[k]} = \sum_{k'=0}^{9} dp[m-1][k'] \cdot I[l \le |k'-k| \le r]$$

Сведем к матричному ДП

$$dp[m][k] = \underbrace{\begin{pmatrix} I[l \le |0-k| \le r] & I[l \le |1-k| \le r] & \ldots \end{pmatrix}}_{1\times 9} \times \underbrace{\begin{pmatrix} dp[m-1][0] \\ dp[m-1][1] \\ \vdots \\ dp[m-1][9] \end{pmatrix}}_{9\times 1}$$

Более того:

$$\begin{pmatrix} dp[m][0] \\ dp[m][1] \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} I[l \le |0-0| \le r] & I[l \le |1-0| \le r] & \ldots \\ I[l \le |0-1| \le r] & I[l \le |1-1| \le r] & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix}}_{A} \times \begin{pmatrix} dp[m-1][0] \\ dp[m-1][1] \\ dp[m-1][2] \\ \vdots \end{pmatrix}$$

Заметим, что A зависит только от усл. задачи. Б. считать ее фикс.

$$dp[m] = A\, dp[m-1]$$

$dp[m] = A^{m-1} dp[1]$

$dp[1][k] = 1 \quad \forall k \in [1, 9]$

$dp[1][0] = 0$. $\Rightarrow$ осталось возводить A за log.

How to compute A^{m-1} in $O(\log n)$ time?

$$P(A, 0) = E$$
$$P(A, 1) = A$$

If n is even:

$$P(A, n) = P(A, \tfrac{n}{2}) \cdot P(A, \tfrac{n}{2})$$

Example: To compute A^8, compute A^4 once, then square it.

Else:

$$P(A, n) = P(A, \tfrac{n-1}{2}) \cdot P(A, \tfrac{n-1}{2}) \cdot A$$

Example: To compute A^7, compute A^3 (via recursion), square it to get A^6, then multiply by A once more!

For each computing recursion depth is $\log_2 n$. So total $O(\log n)$