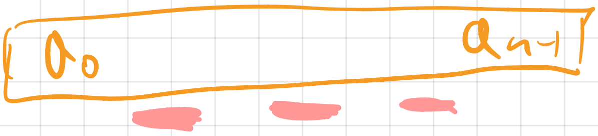
Динамическое программирование

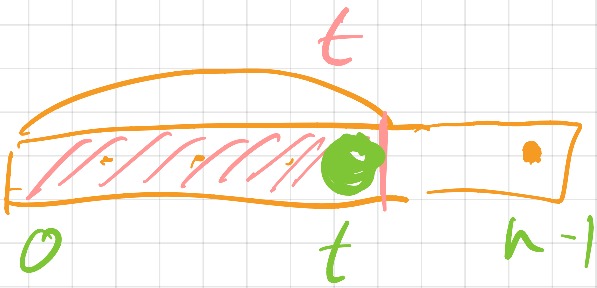
НВП



1 2 6 3 4 5

$\max k : \exists\, i_1, \ldots, i_k : 0 \le i_1 < i_2 < \ldots < i_k \le n-1$

$a_{i_1} < a_{i_2} < \ldots < a_{i_k}.$



Состояние dp_t = длина НВП, заканчивающейся в t

Переход $dp_t = \max\limits_{\substack{i<t \\ a_i < a_t}} dp_i + 1$

База $dp_t \leftarrow 1$

a = 1 2 6 3 4 5 2,5

dp = 1 2 3 3 4 5 3

Время работы: $\Theta(n^2)$

$$\sum_{i=0}^{n-1} i = \Theta(n^2)$$

НВП за $\Theta(n \log n)$

$\boxed{1}$ 2 5 6 4 ~~3 4 5~~

1 2 5 8

1 2 4 8

Нужна по 1 канд. от каждой длины.

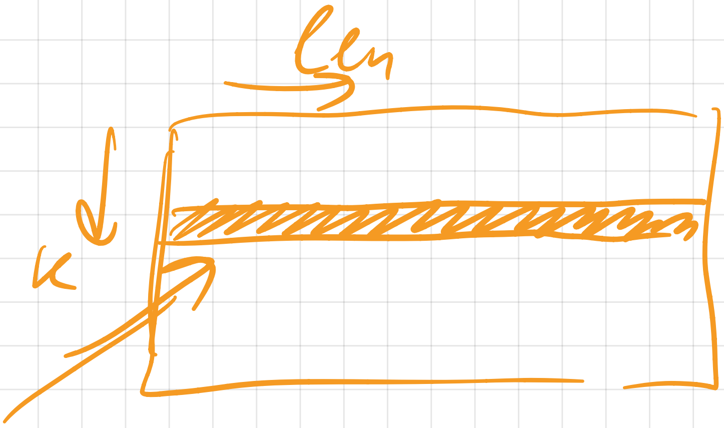
[1]

[1 2]

[1 2 4]

[1 2 5 6]

$dp_k[len]$ = {мин конец В.П. длины len среди эл-тов $a_0 \dots a_{k-1}$}.



Будем хранить только тек. строчку.

[1 2 5 6 4 | . . .

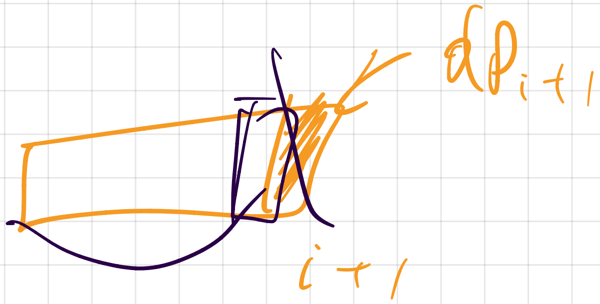
$dp = \{\infty, 1, 2, 4, 6, \infty, \infty, \dots \infty\}$

Обновление $a_0 \dots a_{k-1}$ с $+a_k$.
$a_0 \dots a_k$

1) Наблюдение:

Утв 1 $dp_i < dp_{i+1}$ или $dp_{i+1} = \infty$

Утв 1 ослабл: $dp_i \le dp_{i+1}$



Рассмотрим ответ соответствующий $i+1$

$$\underbrace{b_1, \ldots, b_{i+1}}_{dp_{i+1}}$$

$b_1, \ldots, b_i$ – кандидат на ответ в dp_i

$\Rightarrow dp_i \leq b_i$

и $b_i < b_{i+1}$, т.к. В.П.

$\Rightarrow dp_i < dp_{i+1}$. $\lightning$

2). $[1\ 2\ 5\ 6\ 4 \mid 3.5 \mid$

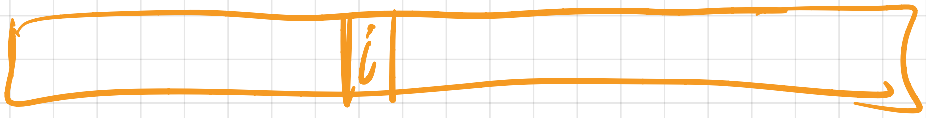
3.5

$dp = \{\infty, 1, 2, 4, 6, \infty, \infty, \ldots, \infty\}$

Можно дописать число 3.5

$\Rightarrow \{-\infty, 1, 2, 3.5, 6, \infty\}$

Утв: нужно пересчитать не более чем 1 эл-т.

dp: 

Пусть i: max, такой что $dp_i < x$

i.e. $(dp_0 \le dp_1 \le \dots \le dp_i < x)$

Тогда $dp_{i+1} \ge x$

и dp_{i+1} релаксируется x

С другой стороны, если

произвести релакс

$\boxed{dp_j}$ $j<i$

x

$dp_{j+1} = \min(dp_{j+1}, x)$

$\wedge$

x

$O(n \log n)$

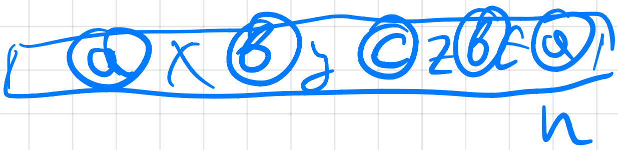
└ n раз добавляем эл-т

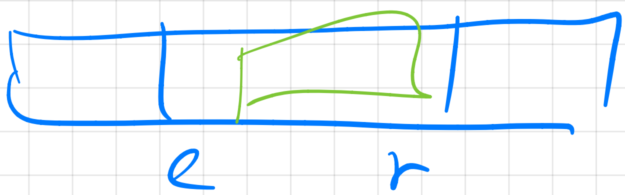
└ n раз бинпоиск

└ n раз релакс.

Наибольший подпалиндром

abcba



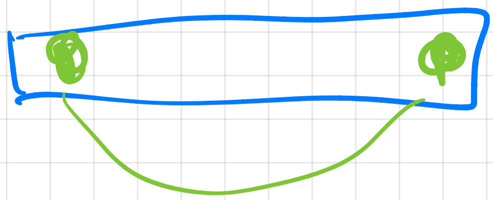


Состояние $dp_{l,r} = \max$ подпалиндром среди $s_l .. s_r$.

Переход $dp_{l,r} \leftarrow dp_{l+1,r}$

$\leftarrow dp_{l,r-1}$

$\leftarrow dp_{l+1,r-1} + 2$ если $s_l = s_r$



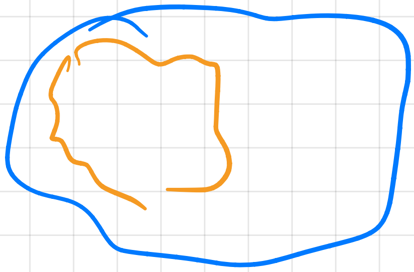
Базы

$dp_{i,i} \leftarrow 1$

$O(n^2)$ (состояний и переходов)

```
for len = 1 ... n
    for l = 0 .. n - len:
        r = l + len - 1
        calc(l, r)
```

Динамика по подмн-вам



$U = \{0, 1, \dots, n-1\}$

$S \subseteq U$

$f(S)$

$S = \{2, 3, 5\}$

$= 101100_2$

5 4 3 2 1 0

$[0 \dots 2^n - 1]$

$S_1, S_2 \subseteq U, \; x \in U$

$S_1 \cup S_2$ $\qquad$ $S_1 | S_2$

$S_1 \cap S_2$ $\qquad$ $S_1 \& S_2$

$S_1 \setminus S_2$ $\qquad$ $S_1 \& (\sim S_2)$

$= S_1 \cap (U \setminus S_2)$

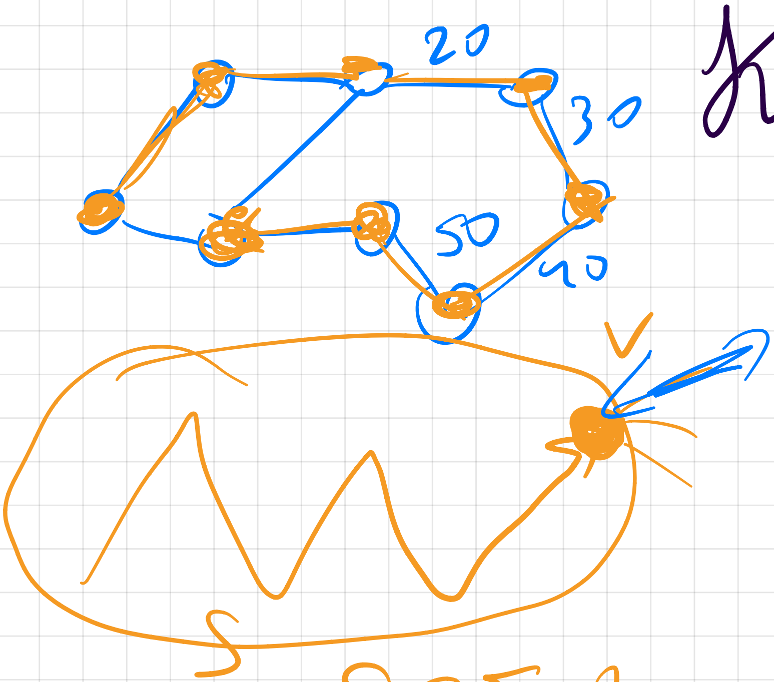
$\{x\}$ $\qquad$ $1 << x$

$x \in S$?

$S \mathrel{!}= S \mid (1 \ll x)$

$(S \,\&\, (1 \ll x)) \mathrel{!}= 0$.

Коммивояжёр



$S \subseteq [n]$, $v \in S$ — посл. вершина

текущий вес

$2^n \cdot n$ состояний

Состояние

$dp[v][S]$: min по весу путь посещ S и заканчивающийся в v.

Переход

$2^n \cdot n^2$ переходов. for $u \in S$, $u \neq v$

$dp[v][S] \leftarrow dp[u][S \setminus (1 \ll v)] + w[u][v]$.

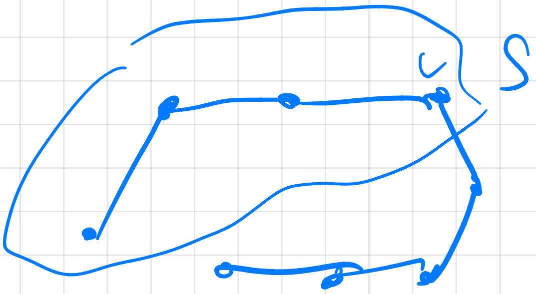
База $\forall v \quad dp[v][1 \ll v] = 0.$

$dp[*][*] = +\infty$

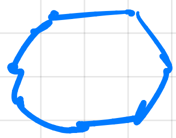
Время: $O(2^n \cdot n^2)$

```
for S = 0..(1<<n)-1:
    for v = 0..n-1
        if v ∉ S
            continue
        dp[v][S] = вычислить
```

Гамильтонов путь



= посетить каждую вершину 1 раз

(цикл )

$dp[v][S]$ = можно или не можно посетить S и закончить в v.

Состояние

$$dp[v][S] \Rightarrow dp[u][S \cup (1 \ll u)]$$

$(v, u): G[v][u] = 1$

$u \notin S$

переходы

$$dp[v][S] \xleftarrow{\text{for } u} dp[u][S \setminus (1 \ll v)]$$

$(v, u): G[v][u] = 1$

$u \in S$

$u \neq v$

база

$$dp[v][1 \ll v] = 1.$$

$$O(2^n \cdot n^2).$$

Новая ДП.

сост

$$dp'[S] = \{v : dp[v][S] = 1\}$$

$2^n \cdot n$

переходы

$$dp'[S] = \{v : dp[v][S] = 1\}$$

база

$$dp'[1 \ll v] = 1 \ll v \quad \forall v.$$

Calc(S):

for $v = 0..n-1$

if $v \in S$:

if $\exists u : G[v][u]=1$ & $u \in dp'[S\setminus\{v\}]$

$dp'[S] \mathrel{|}= (1 \ll v)$

$2^n \cdot n^2$

Вычисление cost за n^2.

$\Rightarrow 2^n \cdot n^2.$

$adj[v] = \{u \mid G[v][u]=1\}$

Calc(S):

for $v = 0..n-1$

if $v \in S$:

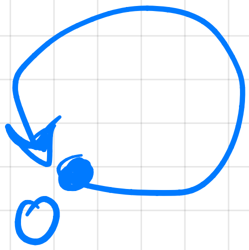
if $(adj[v]$ & $dp'[S\setminus\{v\}]) \neq 0$:

$dp'[S] \mathrel{|}= (1 \ll v)$

$2^n \cdot n$

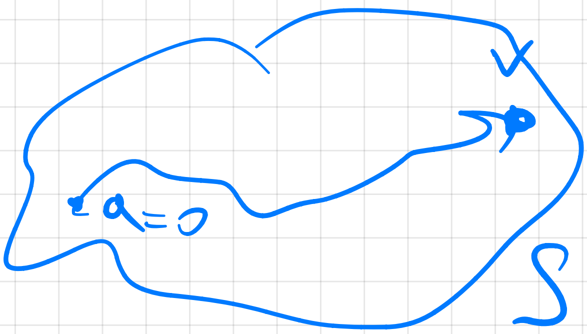
Гам. Цикл.

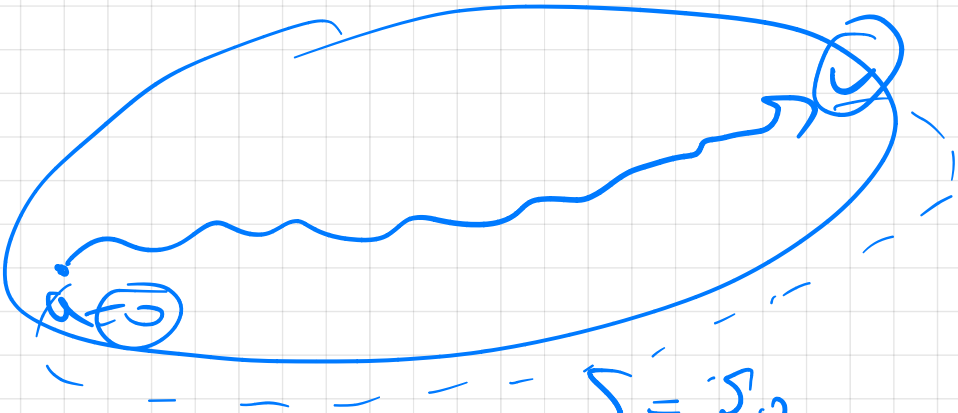
dp[u][v][S] = ∃? путь из $a \to v$ посещ. в точн. S.



dp[v][S] =

= ∃? путь из $0 \to v$ посещающий в точности S.

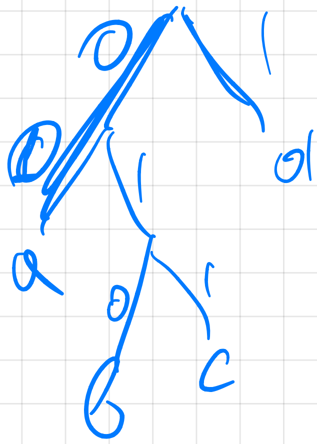




$S = \{0, \dots, n-1\}$

Беспрефиксные коды.

a b c d



a : 00	00
b : 010	01
c : 011	10
d : 1	11

abc → $\boxed{a}\boxed{b}\boxed{c}$

abc ← $\boxed{\underbrace{00}_{a}\ \underbrace{010}_{b}\ \underbrace{011}_{c}}$

Задача: дан текст: придумать коды так что

$$\sum_{c \in \Sigma} \text{частота}(c) \cdot \text{длина кода}(c) \to \min$$

Хемминга.

Задача про упоряд. коды

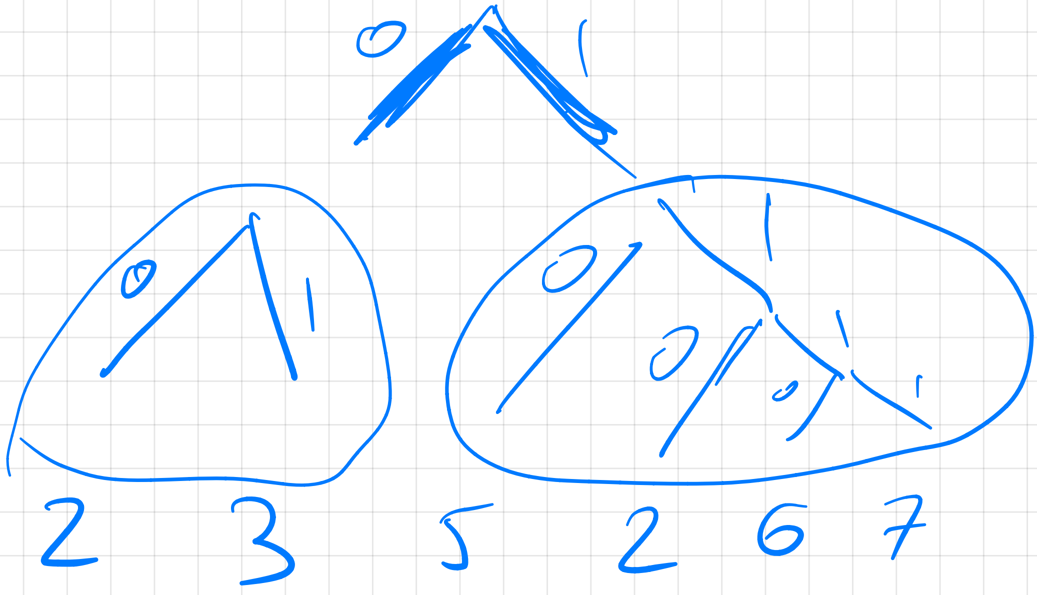
$a_0 \; \dots \; a_{n-1}$ ← буквы

$s_0 \; \dots \; s_{n-1}$ ← коды.

$q_0 \; \dots \; q_{n-1}$ — частоты

$a_i < a_{i+1}$, $s_i < s_{i+1}$

$$\sum_{i=0}^{n-1} q_i \cdot |s_i| \to \min$$



e r

Состояние $dp[l][r]$ = кратч. код если нужно кодировать только буквы от l до r

<u>Переход</u>

$$dp[l][r] = \min_{m=l..r-1} dp[l][m] + dp[m+1][r] + \sum_{i=l..r} q_i$$

baze : $dp[i][i] = 0$.