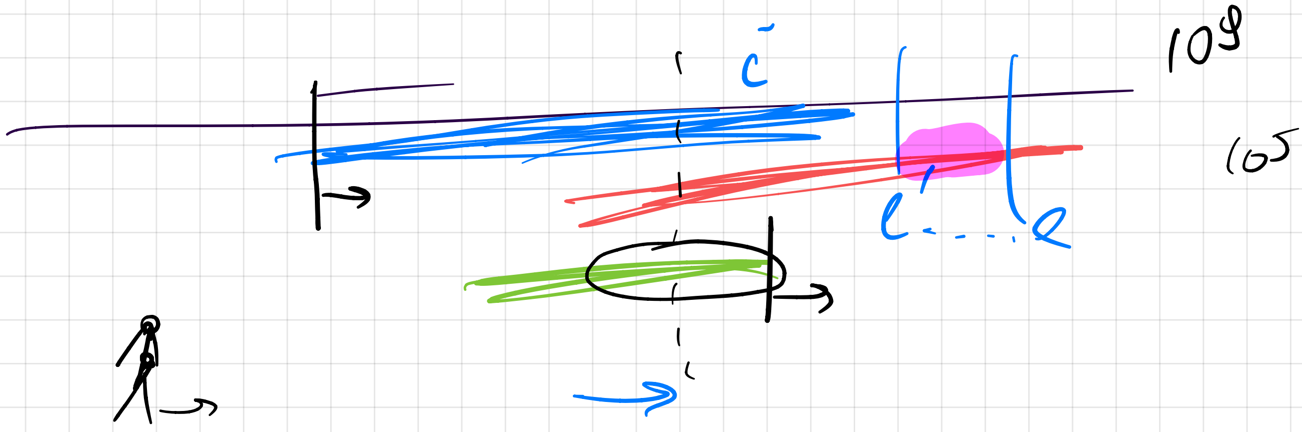
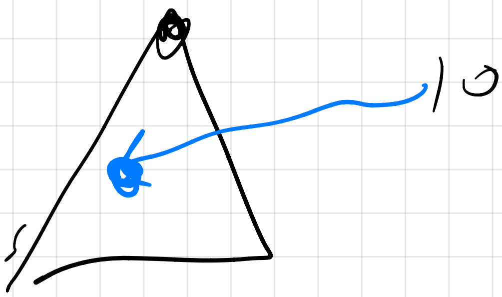
Том Сойера

Сканлайн



Способ 1:

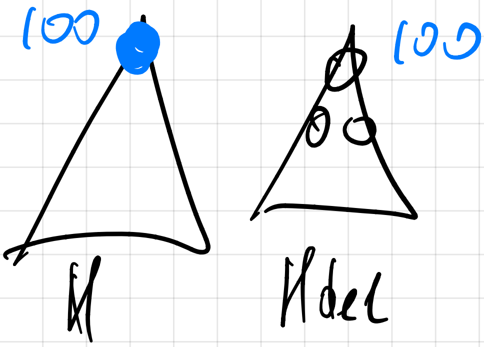
хранить ссылку



dict()

$O(1)$

Способ 2:



last = 0

for e in events:

обработать.

ans[Тек красно] += (last − e.t)

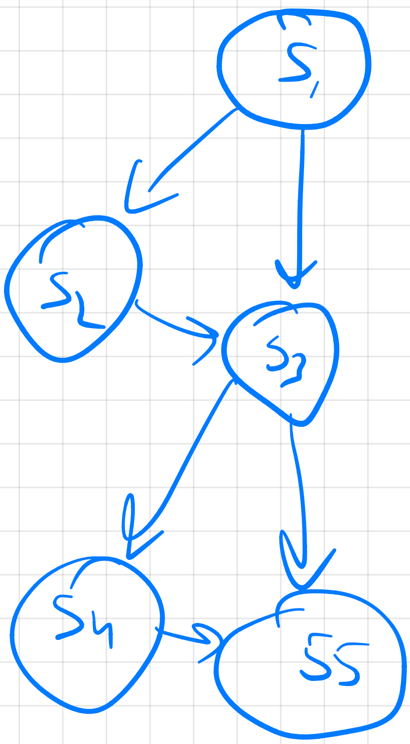
last = e.t

10^9

$10^6 - 10^8$

Python

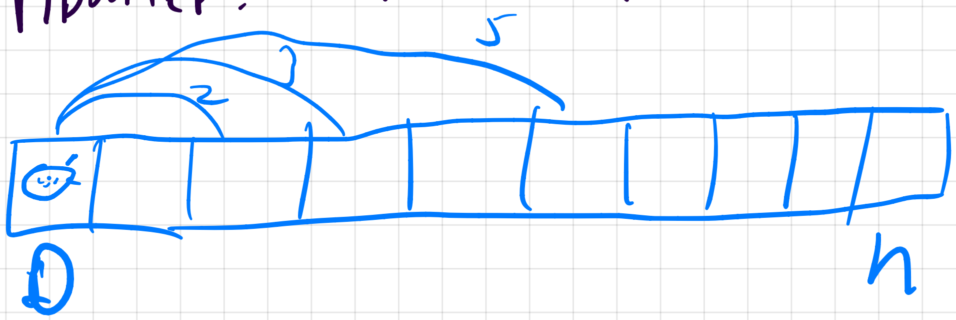
Динамическое Программирование



$f(S_1)$

$can(S)$

Пример: полоска:

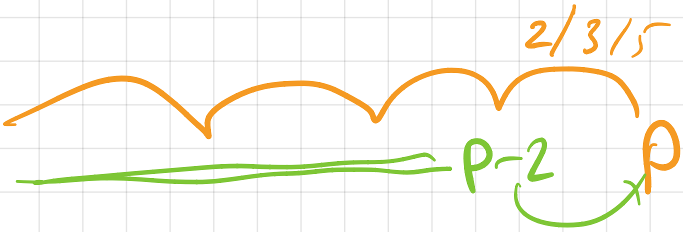


Состояние $Ways(p)$ – сколько способов допрыгать из 0 в p.

$Ways(n) =$ ответ

База $ways(0) = 1$

Переход $ways(p) = ways(p-2) + ways(p-3) +$
$+ ways(p-5)$



Динамика назад


```
ways = [0 for _ in range(n+1)]
ways[0] = 1

for i = 1..n:
    for k ∈ [2, 3, 5]:
        if i >= k:
            ways[i] += ways[i-k]
```

Вычисляется ways[i]

уже посчитано.

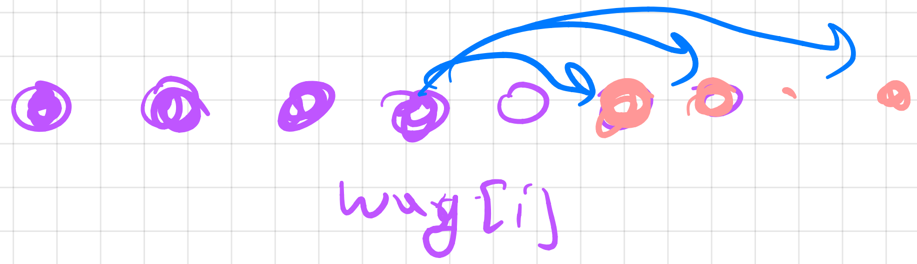
Динамика вперёд

```
ways[0] = 1

for i = 0..n:
    for k ∈ [2, 3, 5]:
        if i+k ≤ n:
            ways[i+k] += ways[i]
```

Возьмем будущее сост.

ways[i] уже посчитан



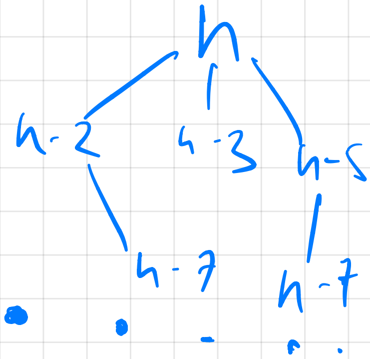
$$O(n \cdot \text{Тарифм. д-бит})$$

<u>Утв.</u> $ways[n] = \Theta(c^n)$

ways[n] $\underbrace{\boxed{}}_{\Theta(n)\ \text{бит}}$

$$\text{Тарифм. д-бит} = \Theta(n).$$

Поэтому нельзя делать рек.




```
def ways(n):
    if n==0: return 1
    if n<0: return 0
    return ways(n-2)+..+ways(n-5)
```

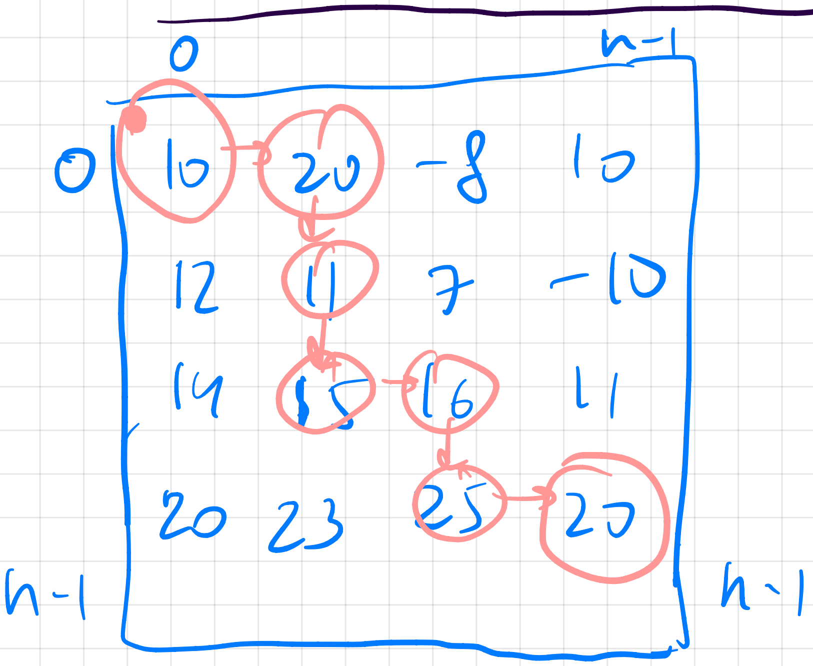
<u>Динамика лениво</u>

```
Ans = [-1 ... -1 .. -1]

def ways(n):
    if n < 0:
        return 0
    if Ans[n] ≠ -1:
        return ans[n]
    result = 0
    if n == 0:
        result = 1
    else
        result = ways(n-2) + ...
                 ... + ways(n-5)
    Ans[n] = result
    return result
```

Мемоизация.

Путь по матрице

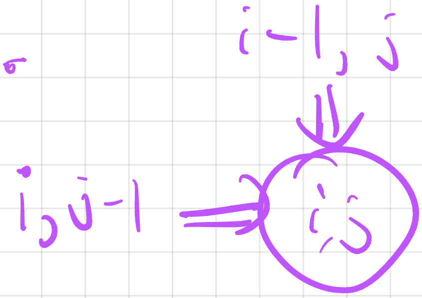


$O(nm)$

Состояние

величина

$d_{i,j} = \min$ по весу пути из $(0,0)$ в (i,j)



$O(nm)$

Переход

$$d_{ij} = \min(d_{i-1,j},\ d_{i,j-1}) + w_{ij}$$

База $d_{00} = w_{00}$

$\Rightarrow$ Время работы: $O(nm)$

$d_{00} = w_{00}$

Динамика между.

```
for i = 0..n-1
    for j = 0..m-1
        if i ≠ 0:
            dij ← di-1,j + wij
        if j ≠ 0:
            dij ← di,j-1 + wij
```

переход (релакс)

$a \Leftarrow b$

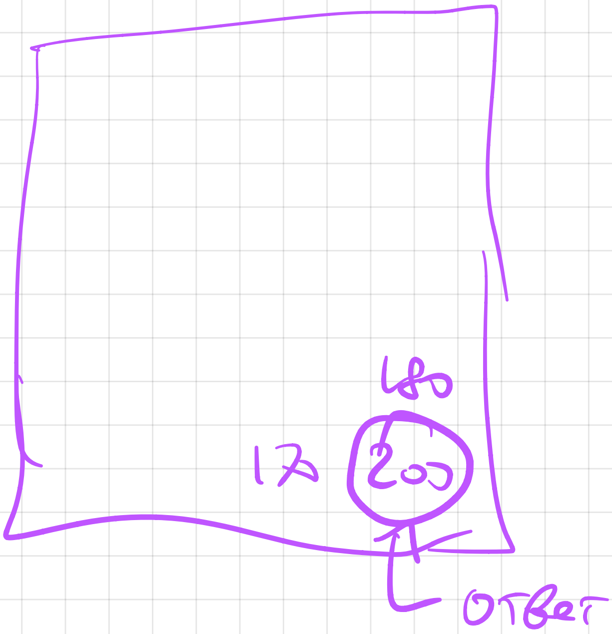
$a = \min(a, b)$

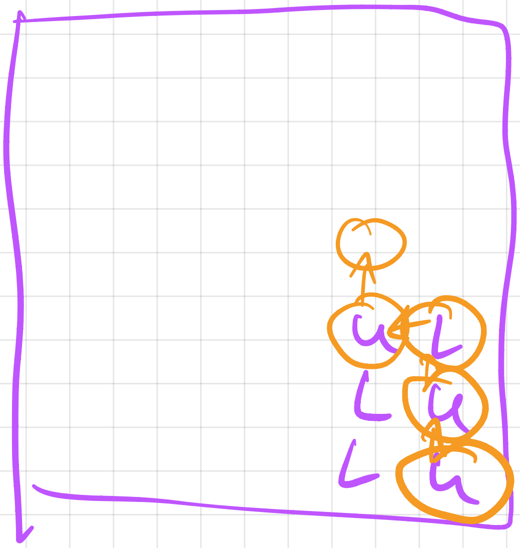
Восстановление ответа.

1) Массив предков.

$$d_{ij} = \min(d_{i-1,j}, d_{i,j-1}) + w_{ij}$$

par_{ij} = U или L в зависимости от того где мин.





$i = n-1$

$j = m-1$

while $i \neq 0$ && $j \neq 0$:

ans.push(P_{ij})

перейти в $(i-1, j)$ или $(i, j-1)$

2) Без массива предков.

$d_{00} = w_{00}$

for $i = 0..n-1$

for $j = 0..m-1$

if $i \neq 0$:

$d_{ij} \leftarrow d_{i-1,j} + w_{ij}$

if $j \neq 0$:

$d_{ij} \leftarrow d_{i,j-1} + w_{ij}$

$i = n-1$

$j = m-1$

while $i \neq 0$ && $j \neq 0$

if $i \neq 0$ && $d_{ij} = d_{i-1,j} + w_{ij}$:

переход ~~на~~ вверх

else

переход влево

Задача о Рюкзаке

(Subset Sum)

$a_0 \quad a_1 \quad \ldots \quad a_{n-1} \in \mathbb{Z}_{\geq 0}$

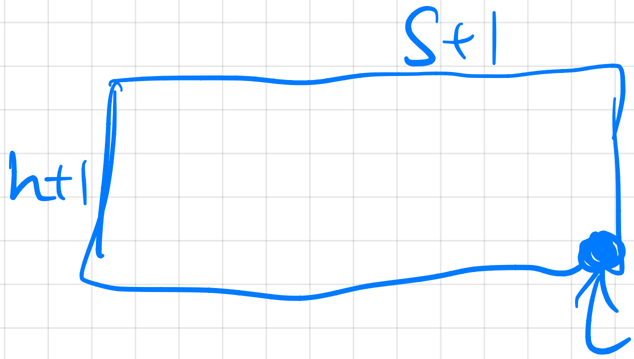
и есть $S \in \mathbb{Z}_{\geq 0}$.

$a_{[0,k)}$ и $W \Rightarrow$

$Can_{k,w} \Rightarrow$ можем ли набрать вес w

Состояние

используя $a_0 \ldots a_{n-1}$



База

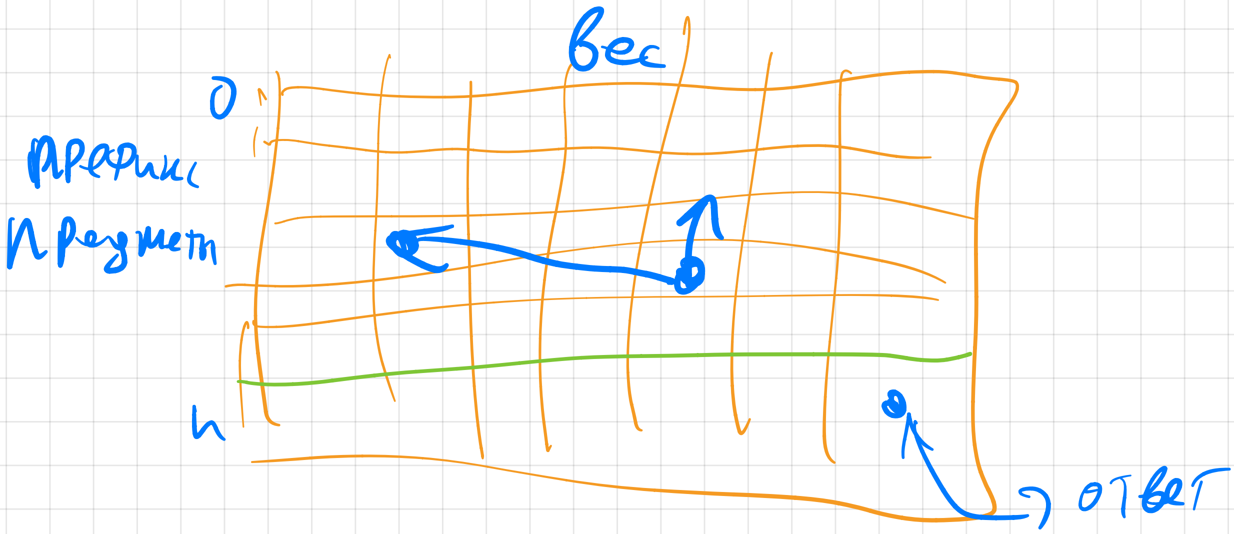
$Can_{0,0} = 1$

$Can_{0,t} = 0 \quad \forall t \geqslant 1$

Переход

$$\underbrace{Can_{k,w}}_{a_0 \ldots a_{k-1}} = \underbrace{Can_{k-1,w}}_{a_0 \ldots a_{k-2}} \vee \ldots$$

$$\vee \underbrace{Can_{k-1,\, w - a_{k-1}}}_{a_0 \ldots a_{k-}}$$

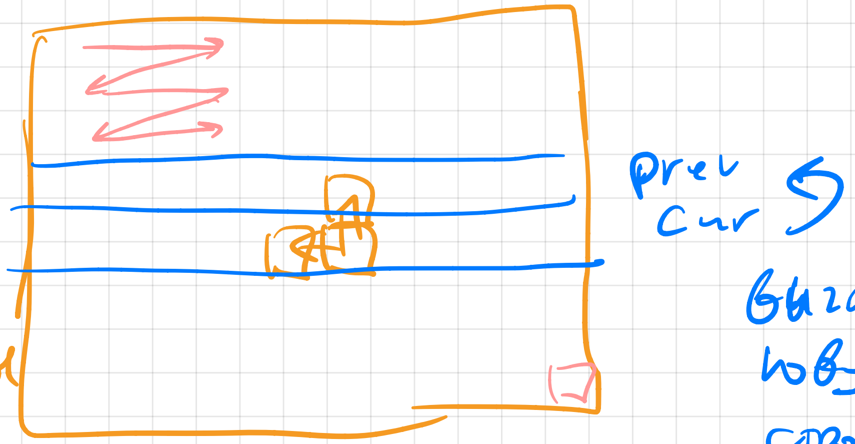


$O(nS)$ времени и памяти

Рюкзак за $O(S)$ памяти

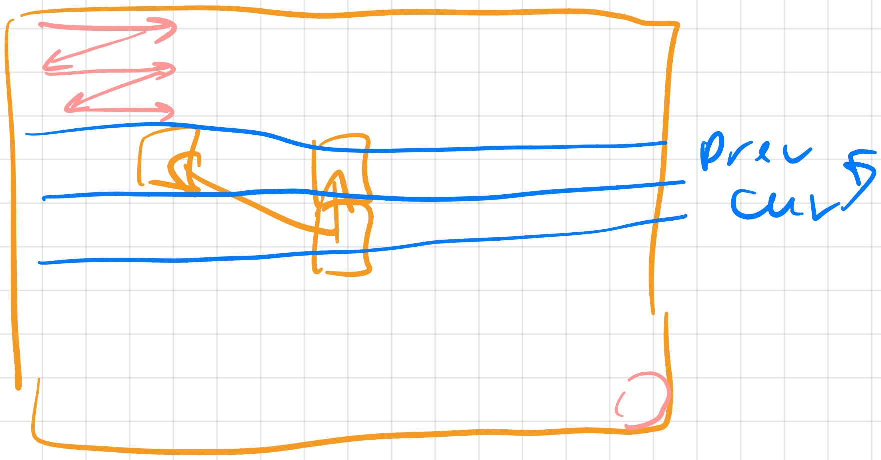
Общая техника.

Задача про путь на матрице



вычислять новую строку через пред.

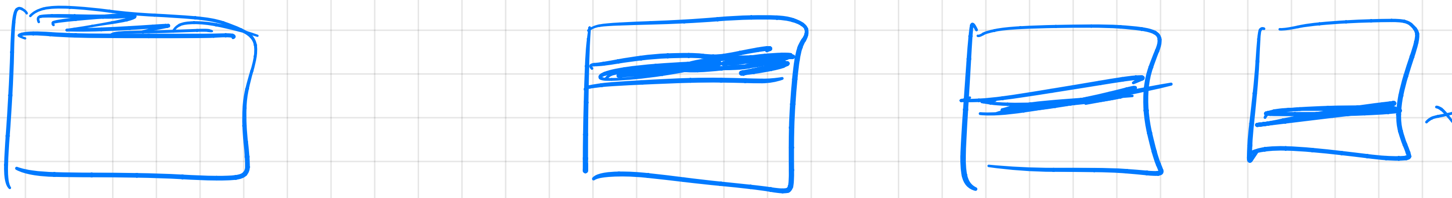
Задача про рюкзак



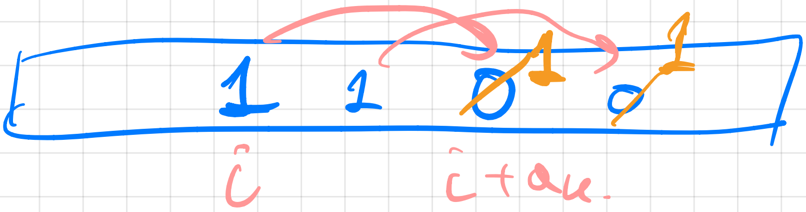
В задаче о рюкзаке.

$$can = [1\ 0\ \dots\ 0]$$

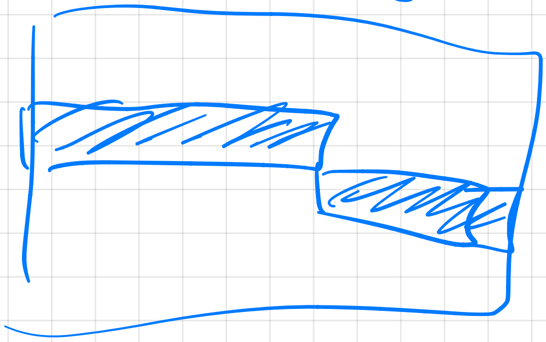
$$con \xrightarrow{a_0} con' \xrightarrow{a_1} con'' \to \dots$$



$$\underset{a_0 \dots a_{k-1}}{can} \xrightarrow{a_k} \underset{a_0 \dots a_{k-1},\ a_k}{can}$$




```
for w = S .. a_k
    if can[w - a_k]
        can[w] = 1
```



$O(S)$

Восстановление ответа в subset sum

for $w = S \ldots a_k$
    if $can[w - a_k]$ & $\neg can[w]$:
        $can[w] = 1$
        $parent[w] = a_k$

массив предков

Рюкзак со стоимостями

$w_0 \ldots w_{n-1}$ W

$c_0 \ldots c_{n-1}$

$i_1 \ldots i_k$ такие что

$$\sum_{t=1..k} w_{i_t} \leq W$$

$$\sum_{t=1..k} c_{i_t} \to MAX$$

Состояние

$dp[k][S]$ = max стоим, если можно исп предметы $0..k-1$ и $\sum w_i = S$.

$O(nW)$ времени

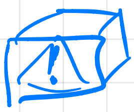
$O(nW)$ памяти.

Переход

Упражнение (1)

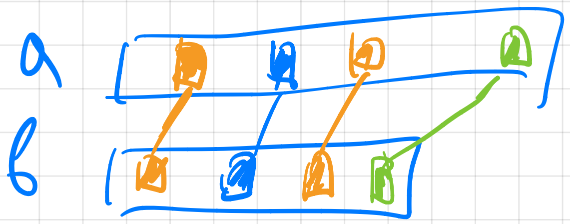
База

Упражнение 2: решить за $O(nW)$ времени и $O(W)$ памяти.

 Можно ли восстановить ответ в данных условиях.

Наибольшую общ подпослед.

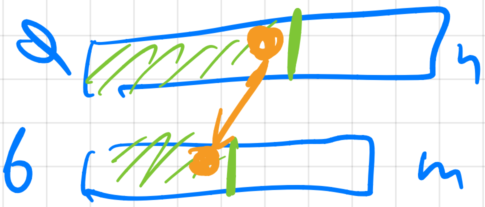
LCS



Макс: $\exists\ i_1 \dots i_k$ $\quad (0 \le i_1 < i_2 < \dots < i_k < |a|)$

$j_1 \dots j_k$ $\quad (0 \le j_1 < j_2 < \dots < j_k < |b|)$

и $\quad a_{i_t} = b_{j_t} \quad \forall t \in [1; k]$



$O(nm)$

Состояние

$$LCS_{k,l} = lcs(a_{0..k-1}, b_{0..l-1})$$

Переход

$$LCS_{k,l} = \max\begin{cases} LCS_{k-1,l} \\ LCS_{k,l-1} \\ LCS_{k-1,l-1} + 1 \quad \text{if} \end{cases}$$

if $a_k = b_l$.

База

$$LCS_{0,0} = 0.$$