

Задача о редакционном расстоянии

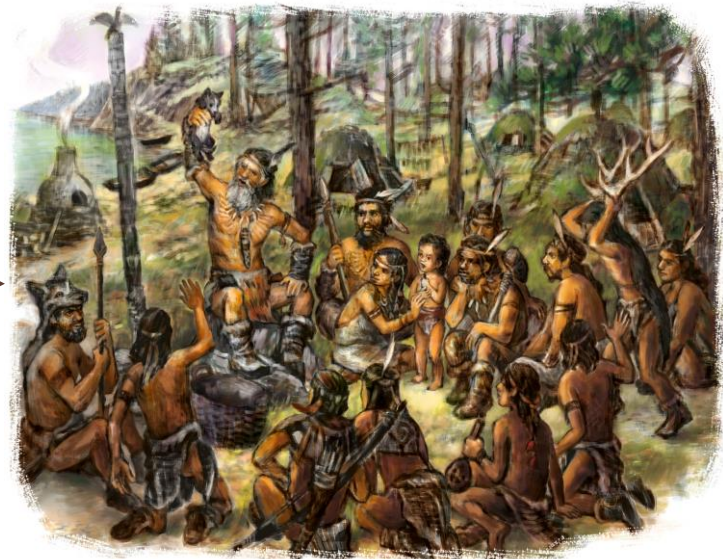


ПОЛЕНО



ПЛЕМЯ

Задача о редакционном расстоянии



ПОЛЕНО



ПЛЕНО

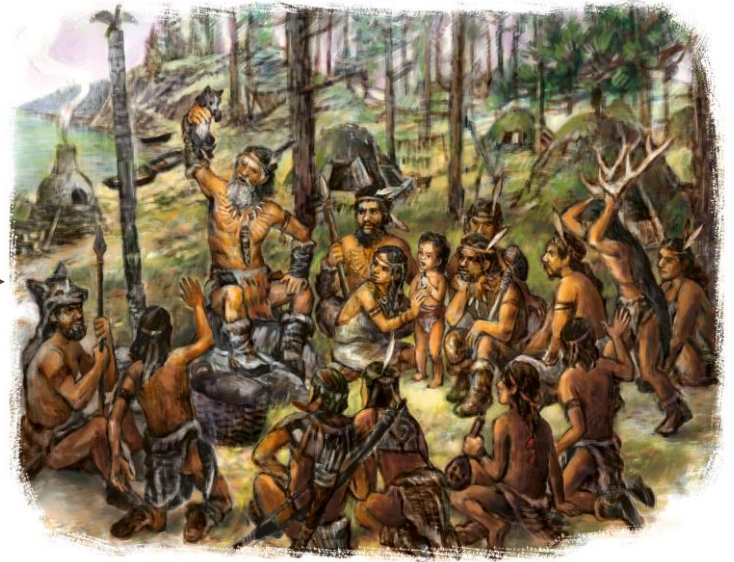


ПЛЕМО



ПЛЕМЯ

Задача о редакционном расстоянии



ПОЛНО → ПЛНО → ПЛЕМО → ПЛЕМЯ

Задача: Имеем два слова. Хотим из первого получить второе.
Можем делать три операции: удалять букву, вставлять букву, изменять букву.

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Задача: Имеем два слова. Хотим из первого получить второе.

Можем делать три операции: удалять букву, вставлять букву, изменять букву.

Динамика?

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Задача: Имеем два слова. Хотим из первого получить второе.

Можем делать три операции: удалять букву, вставлять букву, изменять букву.

Будем пользоваться **принципом оптимальности на префиксе**:
сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

Что тогда будем хранить в $d[i][j]$?

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Задача: Имеем два слова. Хотим из первого получить второе.

Можем делать три операции: удалять букву, вставлять букву, изменять букву.

Будем пользоваться **принципом оптимальности на префиксе:**

сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

$d[i][j]$ – стоимость получения из префикса длины i первого слова префикс длины j второго слова

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Будем пользоваться **принципом оптимальности на префиксе**:
сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

$d[i][j]$ – стоимость получения из префикса длины i первого слова префикс длины j второго слова

вставка:

ab → abc ← хотим получить префикс
второго слова [0...2]

↑
префикс
первого
слова
a[0...1]

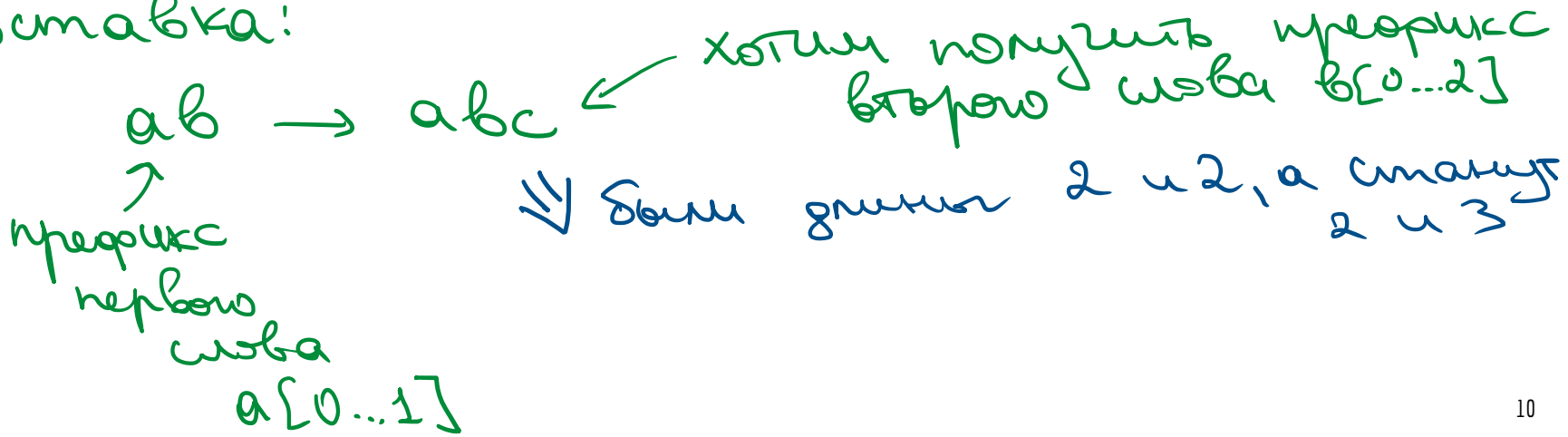
Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Будем пользоваться **принципом оптимальности на префиксе**:
сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

$d[i][j]$ – стоимость получения из префикса длины i первого слова префикс длины j второго слова

вставка:



Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Будем пользоваться **принципом оптимальности на префиксе**:
сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

$d[i][j]$ – стоимость получения из префикса длины i первого слова префикс длины j второго слова

вставка:

$ab \rightarrow abc$

↑
префикс
первого
слова
 $a[0...1]$

← хотим получить префикс
второго слова $b[0...2]$

⇒ были длины 2 и 2, а станут
2 и 3

тогда в динамике продвинемся из
 $d[2][2]$ в $d[2][3]$

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Будем пользоваться **принципом оптимальности на префиксе**:
сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

$d[i][j]$ – стоимость получения из префикса длины i первого слова префикс длины j второго слова

вставка:

ав → авс

↑
префикс
первого
слова
 $a[0...1]$

хотим получить префикс
второго слова $b[0...2]$

Первое уравнение (insert): $d[i][j] = d[i][j-1] + 1$

Тогда в динамике продвинемся из
 $d[2][2]$ в $d[2][3]$

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Будем пользоваться **принципом оптимальности на префиксе**:
сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

$d[i][j]$ – стоимость получения из префикса длины i первого слова префикс длины j второго слова

удаление:
abc → ab ← хотим получить префикс второго слова $b[0...1]$

↑
префикс первого слова $a[0...2]$

Второе уравнение (delete): $d[i][j] = d[i-1][j] + 1$

Тогда в динамике продвинемся из $d[3][2]$ в $d[2][2]$

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Будем пользоваться **принципом оптимальности на префиксе**:
 сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

$d[i][j]$ – стоимость получения из префикса длины i первого слова префикс длины j второго слова

изменения:

оптимальные префиксы

хотим получить префикс второго слова $b[0...2]$

префикс первого слова $a[0...2]$

Третье уравнение (change): $d[i][j] = d[i-1][j] + 1$

тогда в динамике продвинемся из $d[1][1]$ в $d[2][2]$

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Будем пользоваться **принципом оптимальности на префиксе**:
сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

$d[i][j]$ – стоимость получения из префикса длины i первого слова префикс длины j второго слова

Итого три правила:

- 1) Delete: $d[i][j] = d[i-1][j] + \text{deleteCost}$
- 2) Insert: $d[i][j] = d[i][j-1] + \text{insertCost}$
- 3) Change: $d[i][j] = d[i-1][j-1] + \text{changeCost}$

Эти числа могут быть \neq , в зависимости от задачи

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Будем пользоваться **принципом оптимальности на префиксе**:
сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

$d[i][j]$ – стоимость получения из префикса длины i первого слова префикс длины j второго слова

Итого три правила:

- 1) Delete: $d[i][j] = d[i-1][j] + \text{deleteCost}$
- 2) Insert: $d[i][j] = d[i][j-1] + \text{insertCost}$
- 3) Change: $d[i][j] = d[i-1][j-1] + \text{changeCost}$

Из этих трех возможностей хотим ту, что имеет минимальную стоимость

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Будем пользоваться **принципом оптимальности на префиксе**:
сколько надо сделать изменений, чтобы получить из префикса первого слова префикс второго?

$d[i][j]$ – стоимость получения из префикса длины i первого слова префикс длины j второго слова

Из этих трех возможностей хотим ту, что имеет минимальную стоимость

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ \quad D(i - 1, j - 1) + replaceCost & \\) & \end{cases}$$

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

первую строку и индекс заполняем
 просто вставками и удалениями, т.к.
 какое-то из слов вообще не
 рассматривается

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ D(i, j - 1) + insertCost \\ D(i - 1, j) + deleteCost \\ D(i - 1, j - 1) + replaceCost &) \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1					
2 (о)	2					
3 (л)	3					
4 (е)	4					
5 (н)	5					
6 (о)	6					

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

Буквы совпали

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ D(i, j - 1) + insertCost & \\ D(i - 1, j) + deleteCost & \\ D(i - 1, j - 1) + replaceCost & \\) & \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0				
2 (о)	2					
3 (л)	3					
4 (е)	4					
5 (н)	5					
6 (о)	6					

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

делаем вставку буквы "н" во второе слово

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ \quad D(i, j - 1) + insertCost \\ \quad D(i - 1, j) + deleteCost \\ \quad D(i - 1, j - 1) + replaceCost & \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0 ← +1	1			
2 (о)	2					
3 (л)	3					
4 (е)	4					
5 (н)	5					
6 (о)	6					

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

продолжаем делать вставки

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ \quad D(i, j - 1) + insertCost \\ \quad D(i - 1, j) + deleteCost \\ \quad D(i - 1, j - 1) + replaceCost \\) \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2					
3 (л)	3					
4 (е)	4					
5 (н)	5					
6 (о)	6					

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

из "по" удаляем "о"

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ \quad D(i, j - 1) + insertCost \\ \quad D(i - 1, j) + deleteCost \\ \quad D(i - 1, j - 1) + replaceCost &) \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1				
3 (л)	3					
4 (е)	4					
5 (н)	5					
6 (о)	6					

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

замена "о" на "л"

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ \quad D(i, j - 1) + insertCost \\ \quad D(i - 1, j) + deleteCost \\ \quad D(i - 1, j - 1) + replaceCost &) \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1			
3 (л)	3					
4 (е)	4					
5 (н)	5					
6 (о)	6					

↖ +1

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

тут совпадение

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & \\ \quad D(i - 1, j - 1) + replaceCost & \\) & ; j > 0, i > 0, S_1[i] \neq S_2[j] \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1			
4 (е)	4					
5 (н)	5					
6 (о)	6					

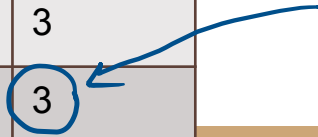
Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ \quad D(i - 1, j - 1) + replaceCost & \\) & \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1	2	3	4
4 (е)	4	3	2	1	2	3
5 (н)	5	4	3	2	2	3
6 (о)	6	5	4	3	3	3

редакционное расстояние



Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

как восстановить ответ?

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ \quad D(i - 1, j - 1) + replaceCost & \\) & \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1	2	3	4
4 (е)	4	3	2	1	2	3
5 (н)	5	4	3	2	2	3
6 (о)	6	5	4	3	3	3

идем по мин. значениям:

вверх — delete
 влево — insert
 диагональ — change/
 equal

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

как восстановить ответ?

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & \\ \quad D(i - 1, j - 1) + replaceCost & \\) & ; j > 0, i > 0, S_1[i] \neq S_2[j] \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1	2	3	4
4 (е)	4	3	2	1	2	3
5 (н)	5	4	3	2	2	3
6 (о)	6	5	4	3	3	3

change



Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

как восстановить ответ?

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & \\ \quad D(i - 1, j - 1) + replaceCost & \\) & ; j > 0, i > 0, S_1[i] \neq S_2[j] \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1	2	3	4
4 (е)	4	3	2	1	2	3
5 (н)	5	4	3	2	2	3
6 (о)	6	5	4	3	3	3

change

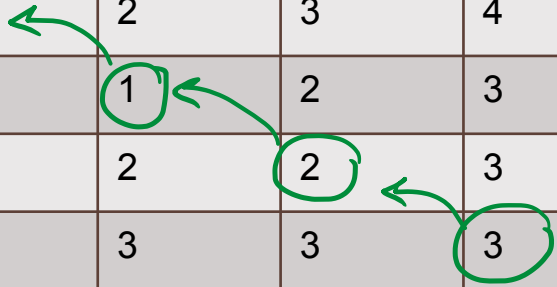
Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

как восстановить ответ?

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & \\ \quad D(i - 1, j - 1) + replaceCost & \\) & ; j > 0, i > 0, S_1[i] \neq S_2[j] \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1	2	3	4
4 (е)	4	3	2	1	2	3
5 (н)	5	4	3	2	2	3
6 (о)	6	5	4	3	3	3



equal

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

как восстановить ответ?

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & \\ \quad D(i - 1, j - 1) + replaceCost & \\) & ; j > 0, i > 0, S_1[i] \neq S_2[j] \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1	2	3	4
4 (е)	4	3	2	1	2	3
5 (н)	5	4	3	2	2	3
6 (о)	6	5	4	3	3	3

equal

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

как восстановить ответ?

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & \\ \quad D(i - 1, j - 1) + replaceCost & \\) & ; j > 0, i > 0, S_1[i] \neq S_2[j] \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1	2	3	4
4 (е)	4	3	2	1	2	3
5 (н)	5	4	3	2	2	3
6 (о)	6	5	4	3	3	3

delete

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

как восстановить ответ?

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & \\ \quad D(i - 1, j - 1) + replaceCost & \\) & ; j > 0, i > 0, S_1[i] \neq S_2[j] \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1	2	3	4
4 (е)	4	3	2	1	2	3
5 (н)	5	4	3	2	2	3
6 (о)	6	5	4	3	3	3

equal

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

время работы?

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & \\ \quad D(i - 1, j - 1) + replaceCost & \\) & ; j > 0, i > 0, S_1[i] \neq S_2[j] \end{cases}$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1	2	3	4
4 (е)	4	3	2	1	2	3
5 (н)	5	4	3	2	2	3
6 (о)	6	5	4	3	3	3

Задача о редакционном расстоянии

ПОЛЕНО → ПЛЕНО → ПЛЕМО → ПЛЕМЯ

время работы?

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ \quad D(i - 1, j - 1) + replaceCost & \\) & \end{cases}$$

размер слова

$$O(|S_1| \cdot |S_2|)$$

	0	1 (п)	2 (л)	3 (е)	4 (м)	5 (я)
0	0	1	2	3	4	5
1 (п)	1	0	1	2	3	4
2 (о)	2	1	1	2	3	4
3 (л)	3	2	1	2	3	4
4 (е)	4	3	2	1	2	3
5 (н)	5	4	3	2	2	3
6 (о)	6	5	4	3	3	3

Задача о редакционном расстоянии

```
int levenshteinInstruction(String s1, String s2, int InsertCost, int DeleteCost, int ReplaceCost):  
    D[0][0] = 0  
    for j = 1 to N  
        D[0][j] = D[0][j - 1] + InsertCost  
    for i = 1 to M  
        D[i][0] = D[i - 1][0] + DeleteCost  
        for j = 1 to N  
            if s1[i] != s2[j]  
                D[i][j] = min(D[i - 1][j] + DeleteCost,  
                             D[i][j - 1] + InsertCost,  
                             D[i - 1][j - 1] + ReplaceCost)  
            else  
                D[i][j] = D[i - 1][j - 1]  
    return D[M][N]
```

слова

минимум из трех
стоимости операций

проверяем, что изменение
всегда нужно
выбор оптимальна

Задача о расстановке знаков в выражении

$$1 + 3 + 2 + 4 = 10$$

Задача о расстановке знаков в выражении

$$1 + 3 + 2 + 4 = 10$$

$$(1 + 3) * (2 + 4) = 24$$

Задача о расстановке знаков в выражении

$$1 + 3 + 2 + 4 = 10$$

$$(1 + 3) * (2 + 4) = 24$$

$$(1 + 3 + 2) * 4 = 24$$

Задача о расстановке знаков в выражении

$$1 + 3 + 2 + 4 = 10$$

$$(1 + 3) * (2 + 4) = 24$$

$$(1 + 3 + 2) * 4 = 24$$

$$1 * 3 * 2 * 4 = 24$$

Задача о расстановке знаков в выражении

$$1 + 3 + 2 + 4 = 10$$

$$(1 + 3) * (2 + 4) = 24$$

$$(1 + 3 + 2) * 4 = 24$$

$$1 * 3 * 2 * 4 = 24$$

$$(1 + 3) * 2 * 4 = 32$$

Задача о расстановке знаков в выражении

$$1 + 3 + 2 + 4 = 10$$

$$(1 + 3) * (2 + 4) = 24$$

$$(1 + 3 + 2) * 4 = 24$$

$$1 * 3 * 2 * 4 = 24$$

$$(1 + 3) * 2 * 4 = 32$$

Задача: Даны числа. Хотим расставить +/*/скобки так, чтобы выражение имело максимальное значение

Задача о расстановке знаков в выражении

1 3 2 4

Задача: Даны числа. Хотим расставить +/*/скобки так, чтобы выражение имело максимальное значение

Идеи?

Задача о расстановке знаков в выражении

1 3 2 4

Задача: Даны числа. Хотим расставить +/* /скобки так, чтобы выражение имело максимальное значение

Динамика: будем пользоваться принципом оптимальности на подотрезке

Задача о расстановке знаков в выражении

1 3 2 4

Задача: Даны неотрицательные числа. Хотим расставить +/*/скобки так, чтобы выражение имело максимальное значение

Динамика: будем пользоваться принципом оптимальности на подотрезке

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

Если известны значения на отрезках $[i \dots k]$, $[k+1 \dots j]$, то на $[i \dots j]$ будем $\max(d[i \dots k] + d[k+1 \dots j], d[i \dots k] \cdot d[k+1 \dots j])$

Задача о расстановке знаков в выражении

$$\begin{array}{c} 3 \quad 8 \\ \underbrace{\quad} \quad \underbrace{\quad} \\ 1 \quad 3 \quad 2 \quad 4 \end{array}$$

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

$$3 + 8 \quad ? \quad 3 * 8$$

Если известны значения на отрезках $[i \dots k]$, $[k+1 \dots j]$, то на $[i \dots j]$ будем $\max(d[i \dots k] + d[k+1 \dots j], d[i \dots k] \cdot d[k+1 \dots j])$

Задача о расстановке знаков в выражении

$$\begin{array}{c} 3 \quad 8 \\ \underbrace{\quad} \quad \underbrace{\quad} \\ 1 \quad 3 \quad 2 \quad 4 \end{array}$$

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

$$3 + 8 < 3 * 8$$

Если известны значения на отрезках $[i \dots k]$, $[k+1 \dots j]$, то на $[i \dots j]$ будем $\max(d[i \dots k] + d[k+1 \dots j], d[i \dots k] \cdot d[k+1 \dots j])$

Задача о расстановке знаков в выражении

$$\begin{array}{cc} 3 & 8 \\ \underbrace{\quad} & \underbrace{\quad} \\ 1 & 3 & 2 & 4 \end{array}$$

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

$$3 + 8$$

<

$$3 * 8$$

← лучший вариант!

Если известны значения на отрезках $[i \dots k]$, $[k+1 \dots j]$, то на $[i \dots j]$ будем $\max(d[i \dots k] + d[k+1 \dots j], d[i \dots k] \cdot d[k+1 \dots j])$

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)				
1 (3)	0			
2 (2)	0	0		
3 (4)	0	0	0	

нуш,
т.к
не бывает
отрезков,
где левая
грань
правее правой

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1			
1 (3)	0	3		
2 (2)	0	0	2	
3 (4)	0	0	0	4

на отрезках
длиной 1 имеем
только один операнд
⇓
сразу записываем

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1			
1 (3)	0	3		
2 (2)	0	0	2	8
3 (4)	0	0	0	4

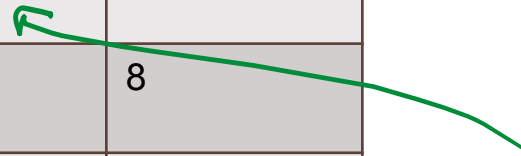
← $\max\left(\begin{matrix} 2+4 \\ 2*4 \end{matrix}\right)$

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1			
1 (3)	0	3	6	
2 (2)	0	0	2	8
3 (4)	0	0	0	4


$$\max \begin{pmatrix} 3+2 \\ 3*2 \end{pmatrix}$$

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1			
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

$3 + 8$
 $3 * 8$
 $6 + 4$
 $6 * 4$ } max

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4		
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

$1 * 3 \Big/ \max$
 $1 + 3 \Big/$

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4	8	
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

$$\left. \begin{array}{l} 1 + 6 \\ 1 * 6 \\ 4 * 2 \\ 4 + 2 \end{array} \right\} \text{max}$$

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4	8	32
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

$$\left. \begin{array}{l} 1+24 \\ 1*24 \\ 4+8 \\ 4*8 \\ 8+4 \\ 8*4 \end{array} \right\} \text{max}$$

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4	8	32
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

Как восстановить
ответ?

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4	8	32
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

Как восстановить
ответ?

Хранить, какую
пару выбрали

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4	8	32 $(0,2), (3,3)$
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

Как восстановить
ответ?

Хранить, какую
пару выбрали

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4	8	32 $(0,2), (3,3)$
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

Как восстановить
ответ?

Хранить, какую
пару выбрали

$$32 = (\quad) * (\quad)$$

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4	8 $(0,1)(2,2)$	32 $(0,2), (3,3)$
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

Как восстановить ответ?

$$32 = (() * 2) * (4)$$

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4 $(0,0) + (1,1)$	8 $(0,1) (2,2)$	32 $(0,2), (3,3)$
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

Как восстановить ответ?

$$32 = ((1+3) * 2) * (1)$$

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4 $(0,0) + (1,1)$	8 $(0,1) (2,2)$	32 $(0,2), (3,3)$
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

Как восстановить ответ?

$$32 = ((1+3) * 2) * (4)$$

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4	8	32
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

Время
работы?

Задача о расстановке знаков в выражении

1 3 2 4

Пусть $d[i][j]$ - максимальное значение, которое мы можем получить на отрезке от i до j .

	0 (1)	1 (3)	2 (2)	3 (4)
0 (1)	1	4	8	32
1 (3)	0	3	6	24
2 (2)	0	0	2	8
3 (4)	0	0	0	4

Время
работы?

$O(N^3)$ 😞

Задача о расстановке знаков в выражении

// a - заданная последовательность из n элементов

```
int maxValueOfExpression(a, n):
```

```
  for i = 1 to n:  
    d[i][i] = a[i]
```

← диагональ

```
  for i = n - 1 downto 1:
```

← цикл по строкам ↑

```
    for j = i + 1 to n:
```

← по столбцам →

```
      for k = i to j - 1:
```

← перебор всех комбинаций

```
        d[i][j] = max(d[i][j], max(d[i][k] + d[k + 1][j], d[i][k] * d[k + 1][j]))
```

```
  return d[1][n]
```

↑

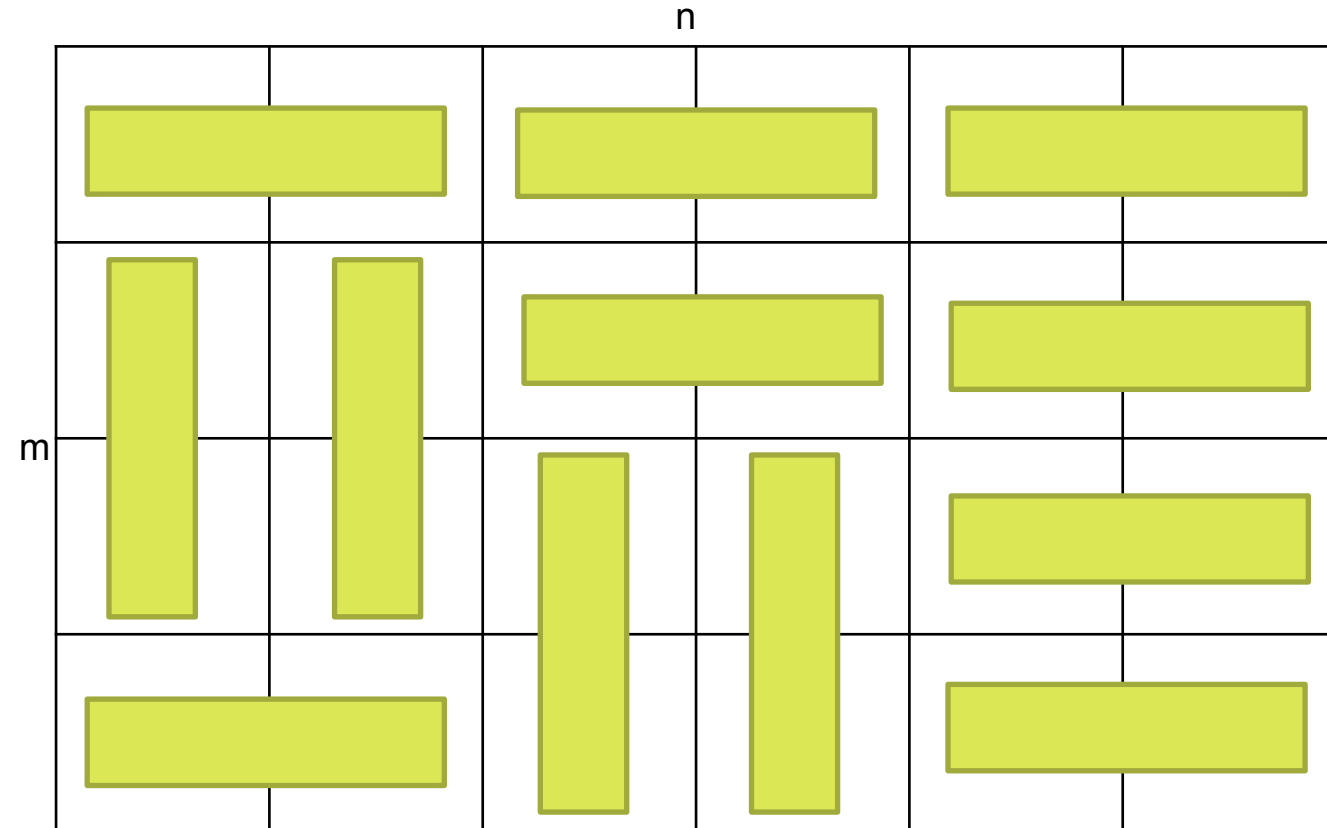
можем
оставить то, что
было

↑

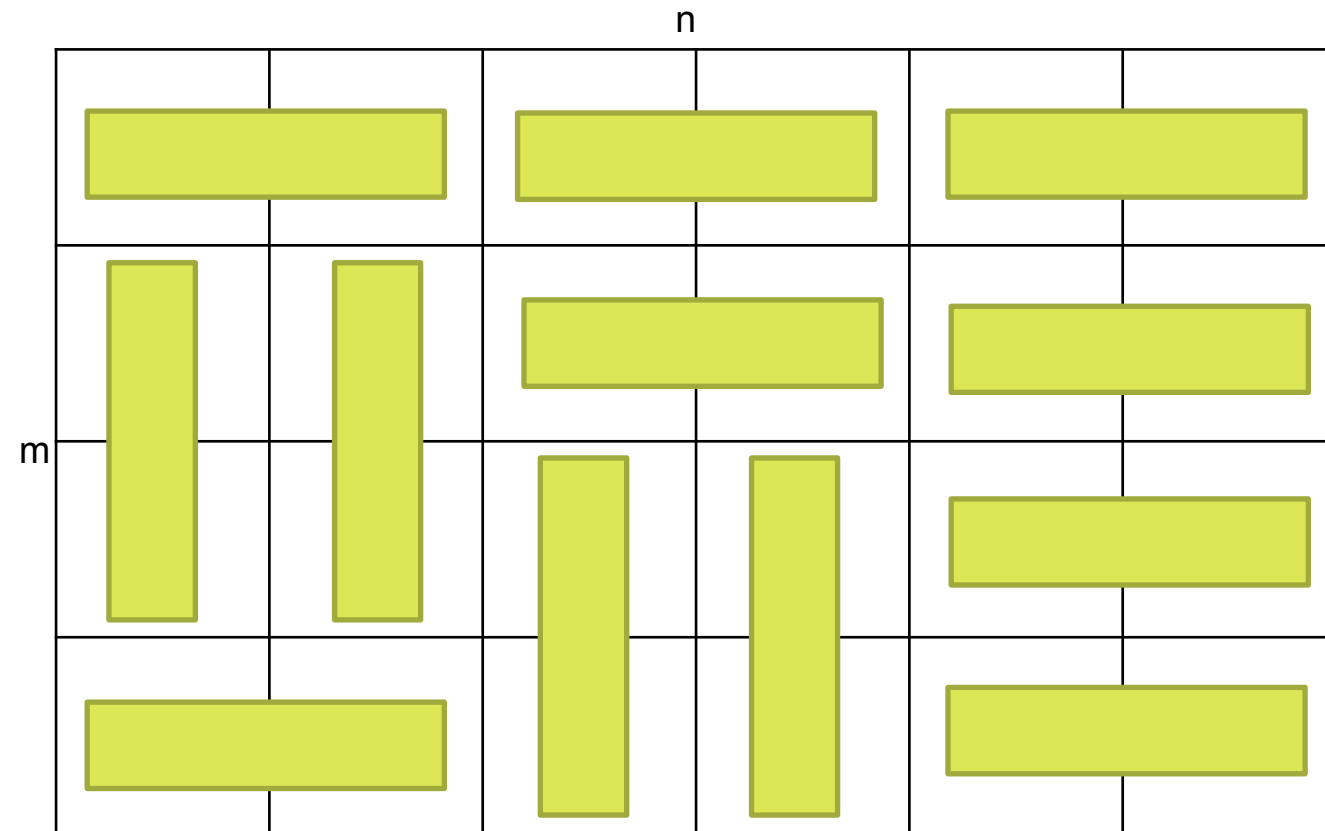
можем
взять новое
значение

↑
нужен
отрезок
от 1 до n (весь)

Задача о замощении домино

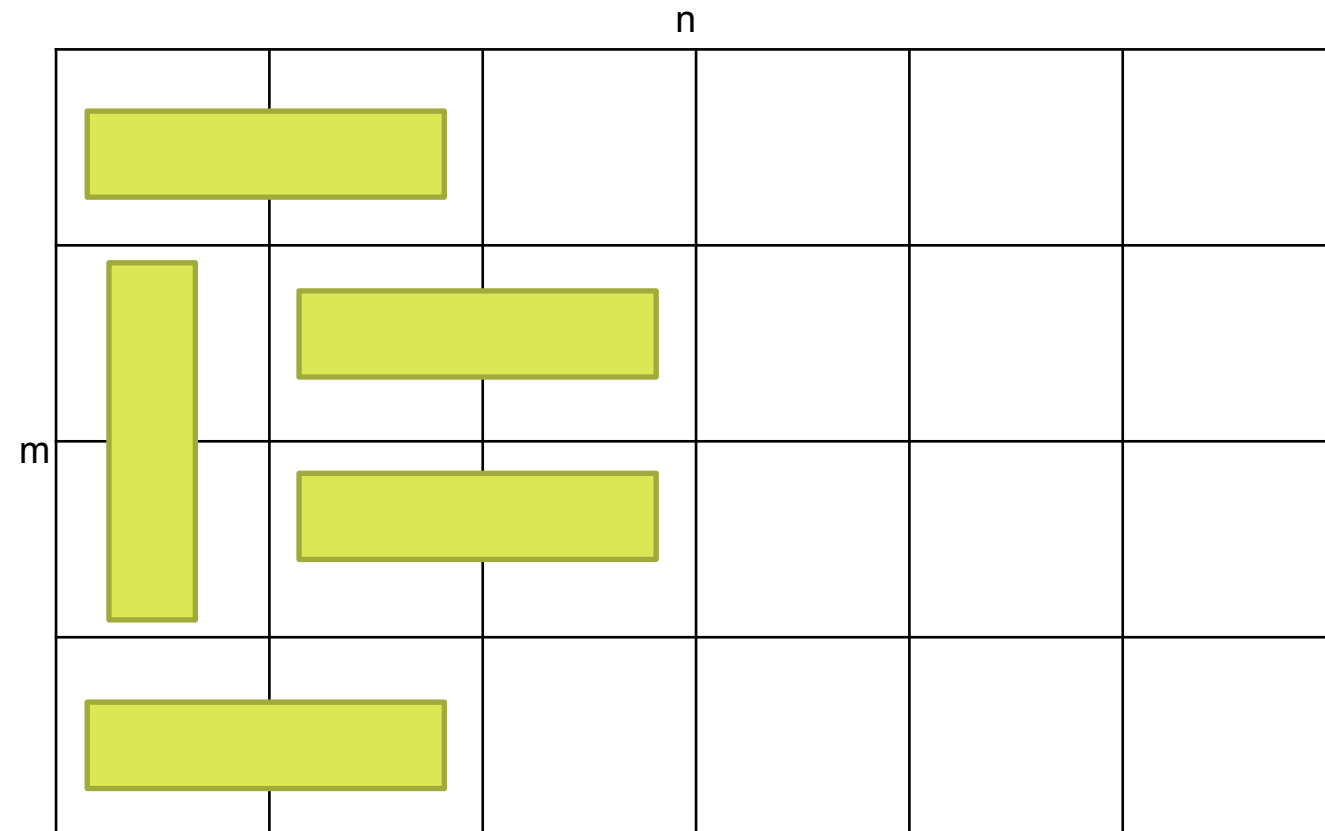


Задача о замощении домино



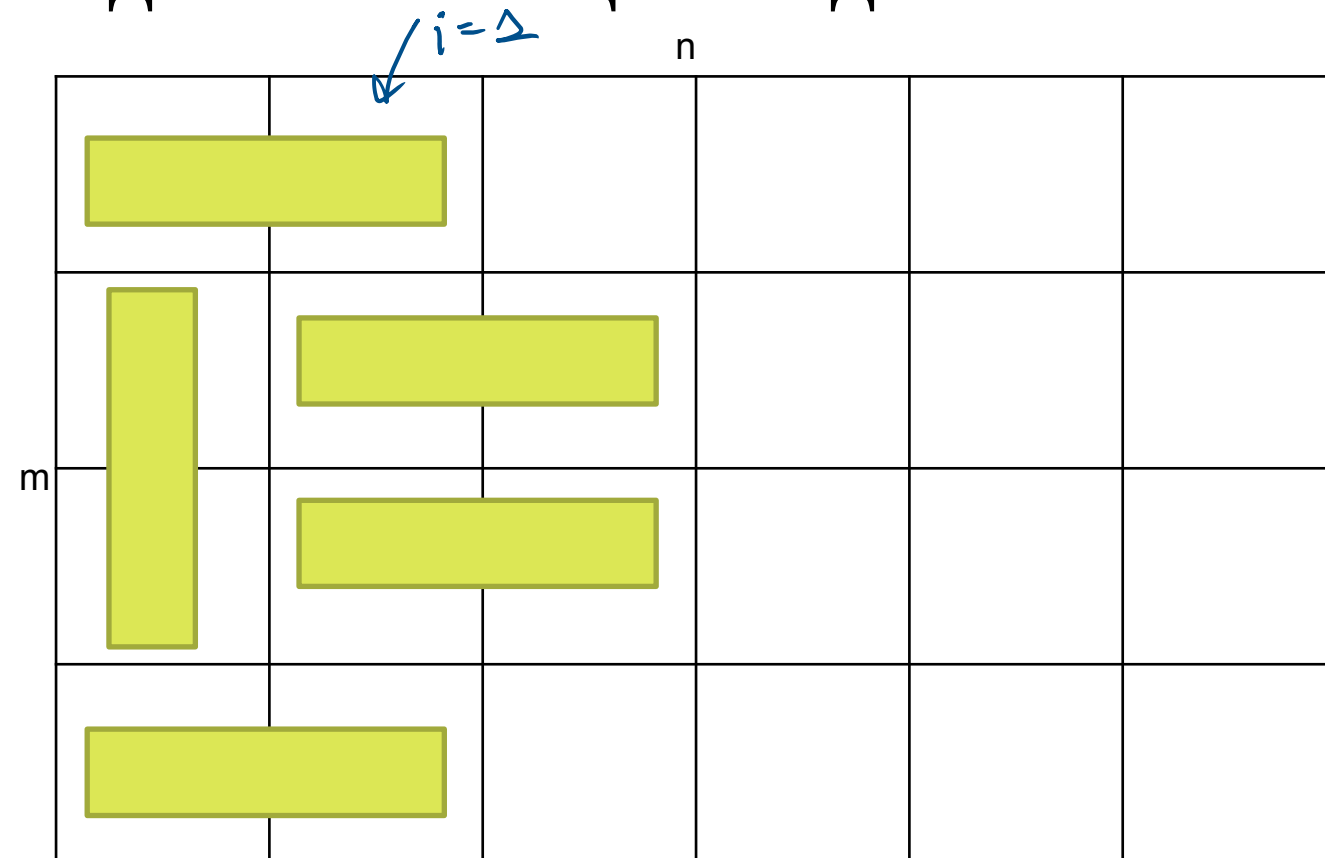
Дана таблица $m \times n$.
Необходимо вычислить
количество способов
заполнить ее доминошками

Задача о замощении домино



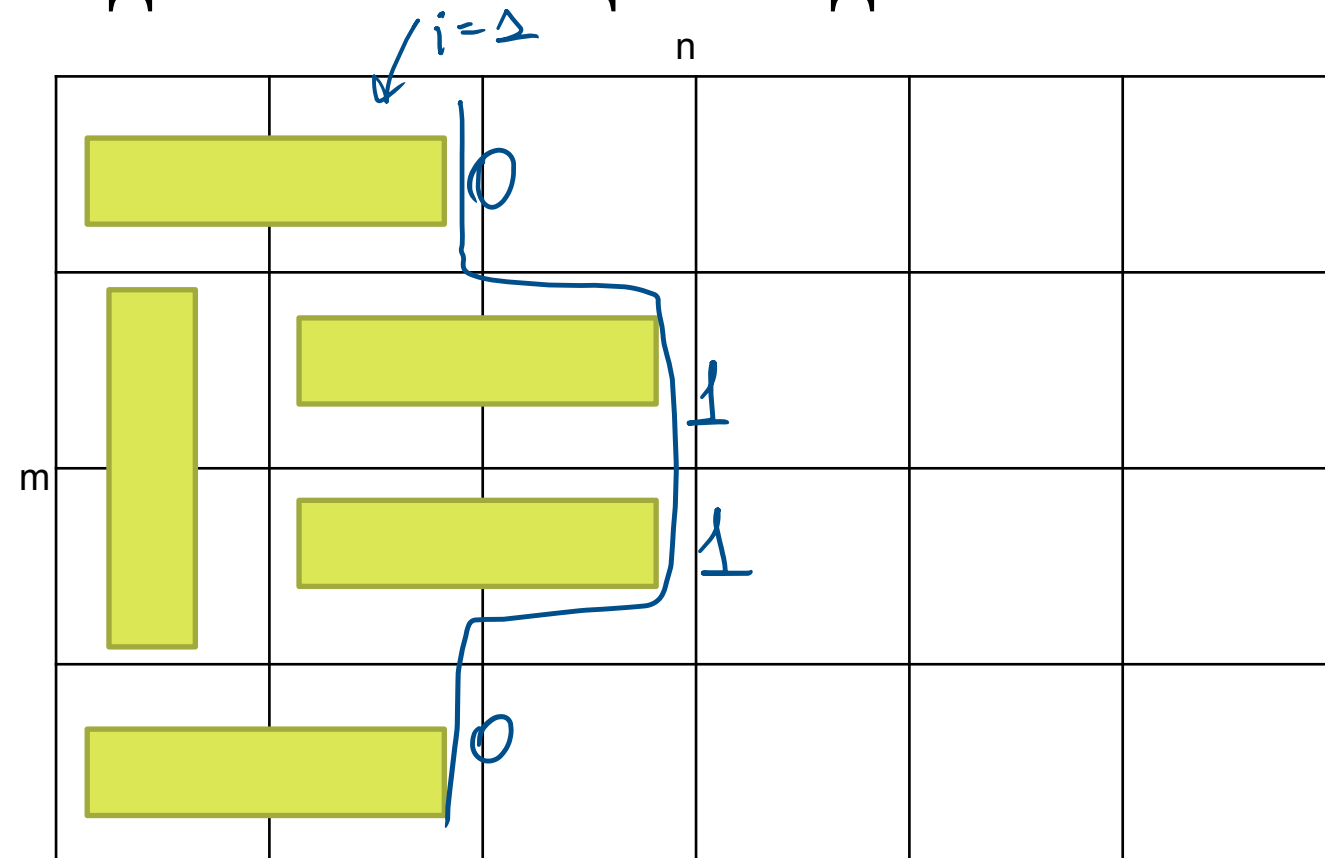
Профиль – для i -ого столбца набор из ноликов и единичек.
0 – домино «не вылезает»
1 – домино вылезает за границы

Задача о замощении домино



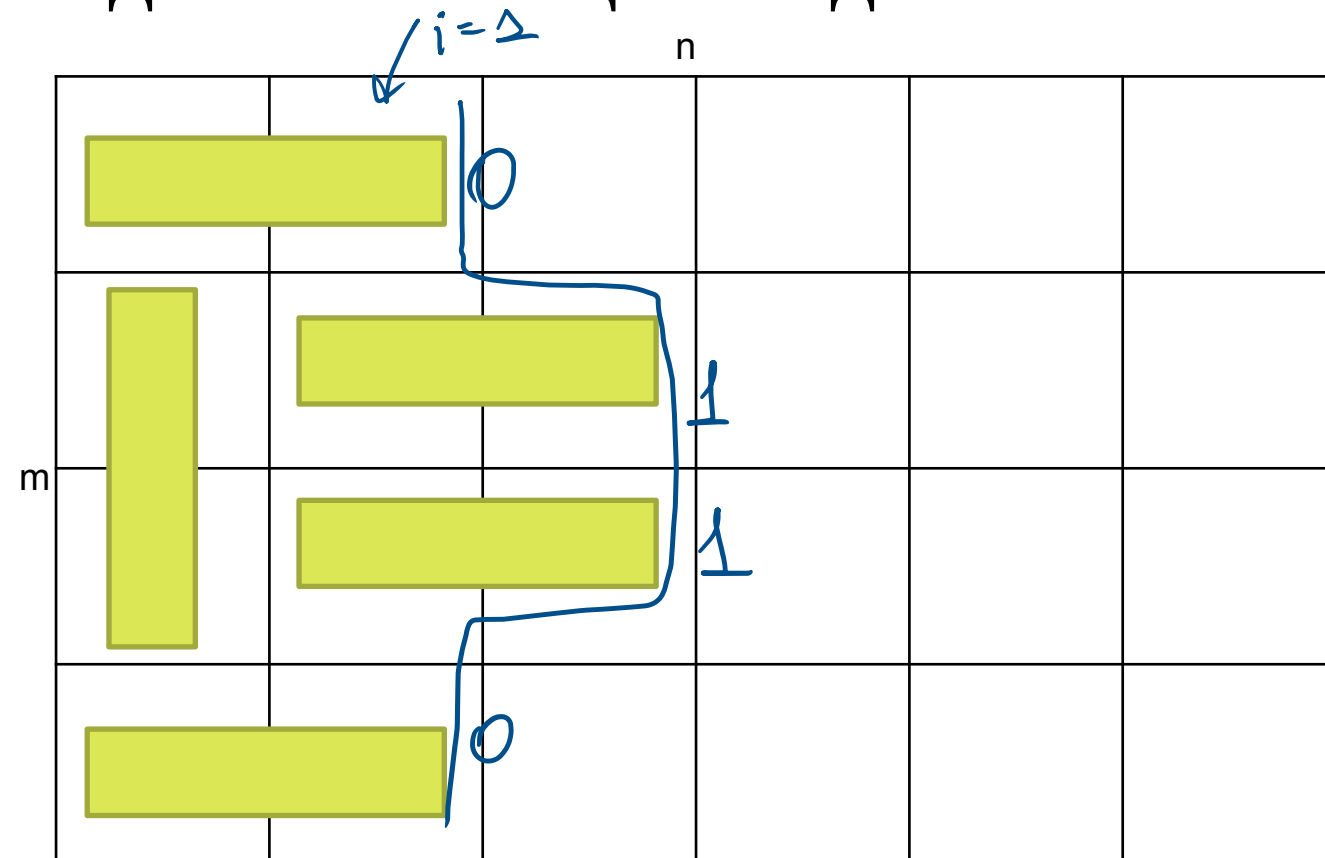
Профиль – для i -ого столбца набор из ноликов и единичек.
0 – домино «не вылезает»
1 – домино вылезает за границы

Задача о замощении домино



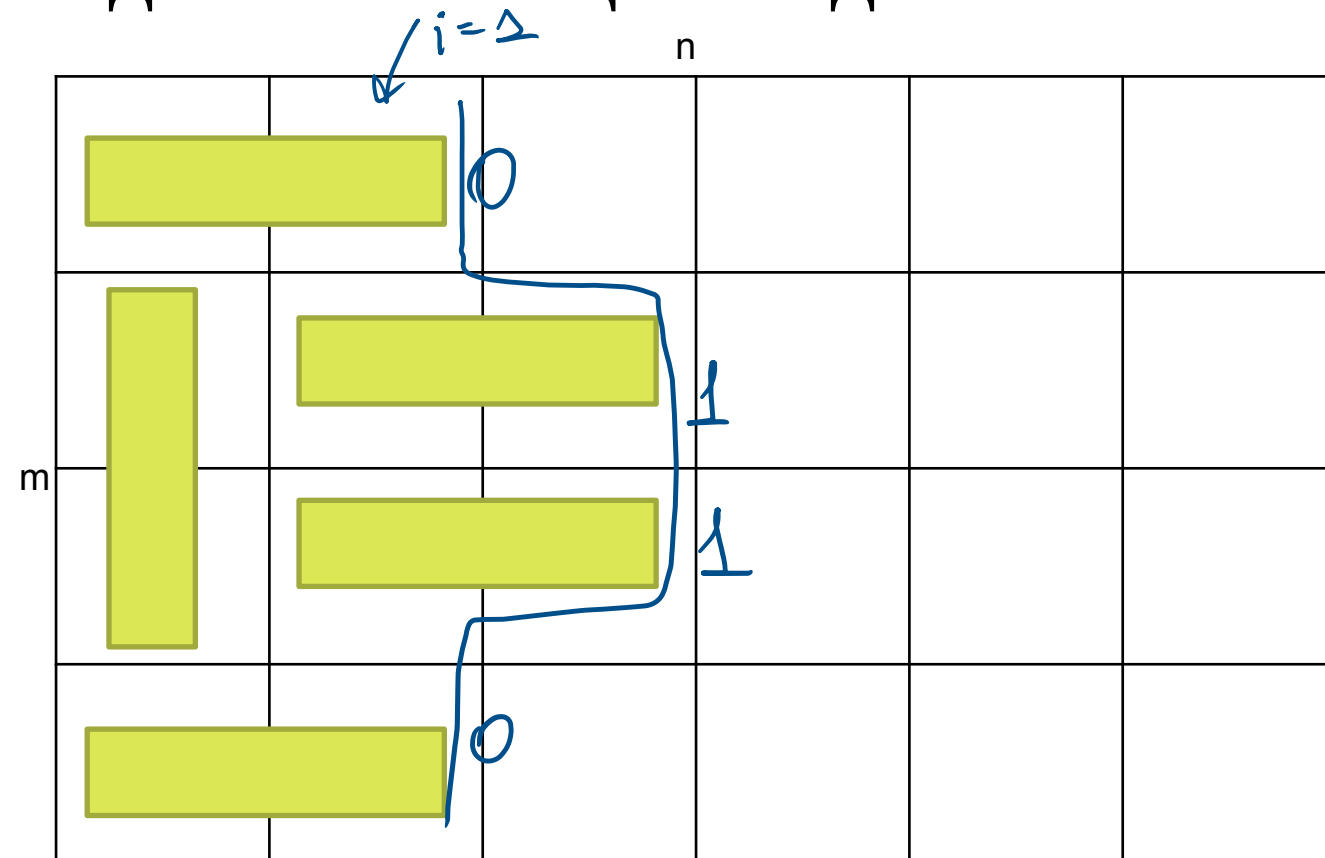
Профиль – для i -ого столбца набор из ноликов и единичек.
0 – домино «не вылезает»
1 – домино вылезает за границы

Задача о замощении домино



Динамика по (i, p) – по номеру столбца и профилю.

Задача о замощении домино

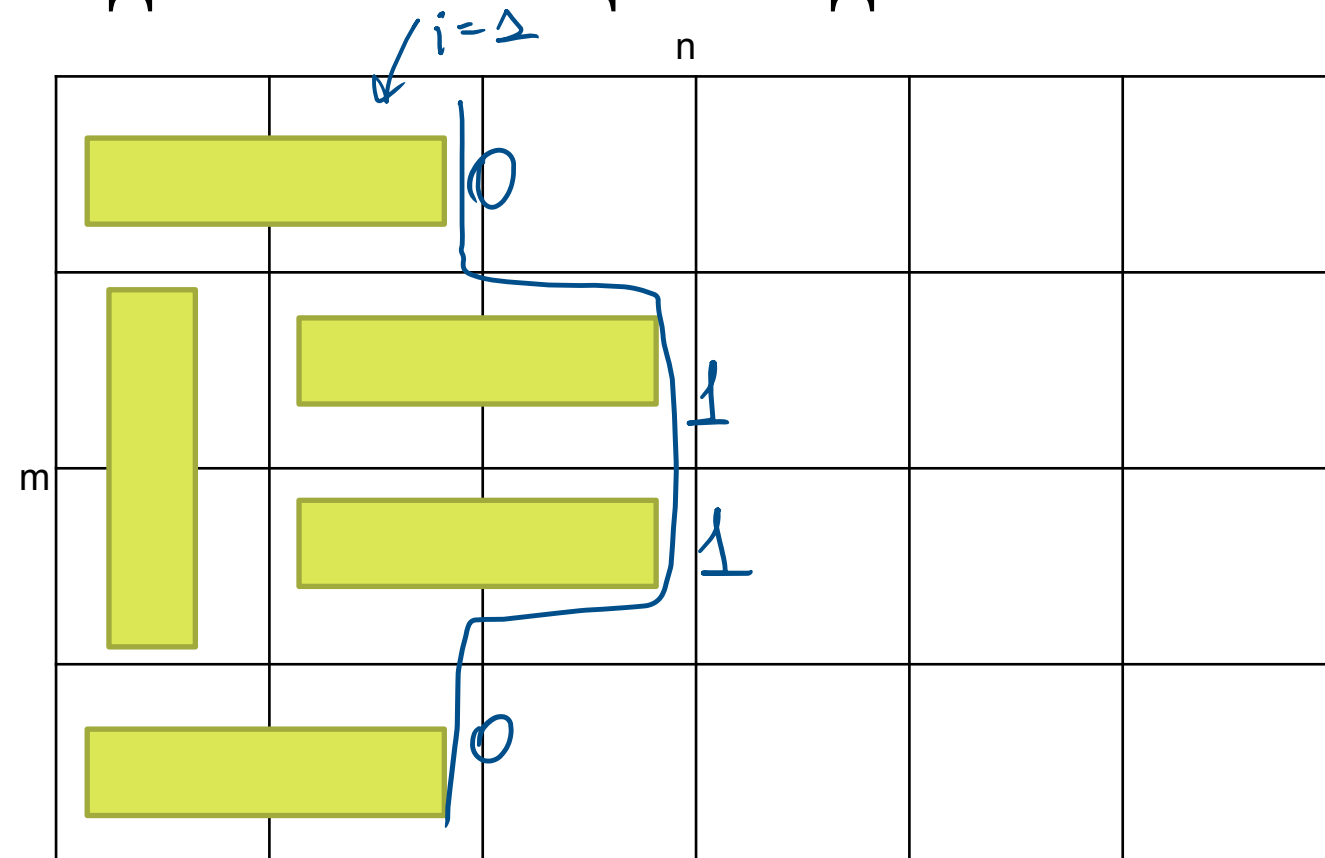


Динамика по (i, p) – по номеру столбца и профилю.

Сколько может быть профилей?

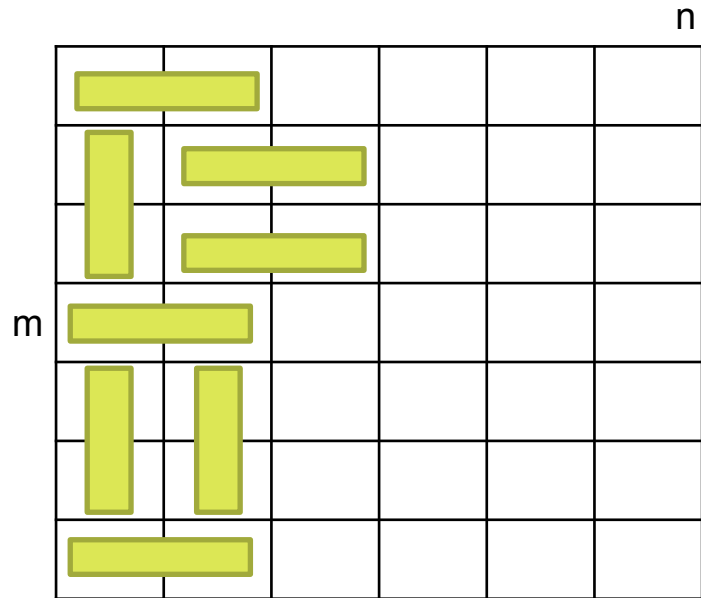
2^m

Задача о замощении домино



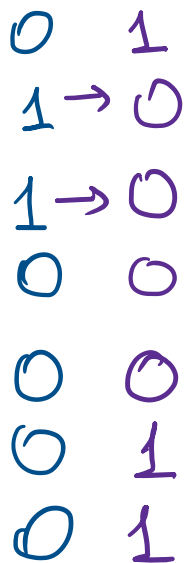
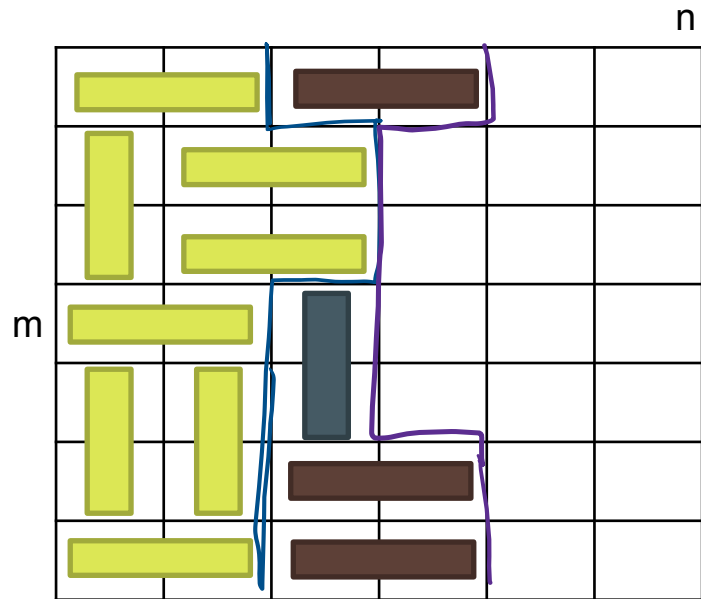
Какие могут быть переходы между профилями?

Задача о замощении домино



Какие могут быть переходы между профилями?

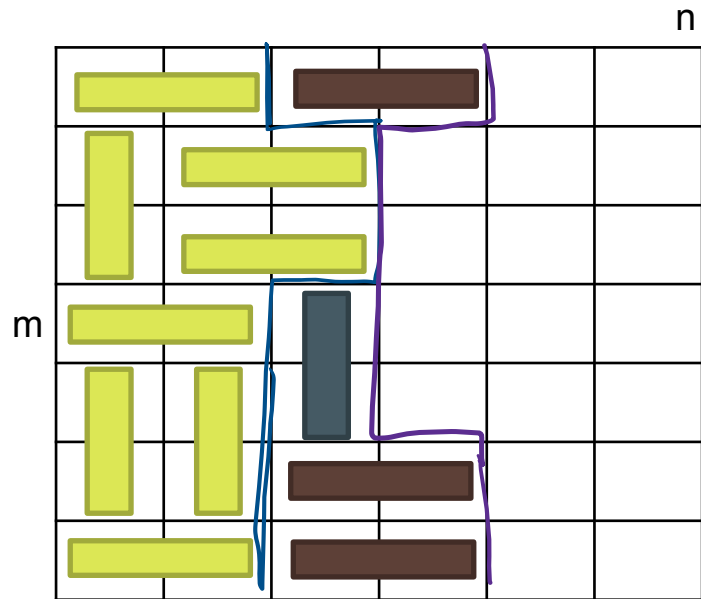
Задача о замощении домино



Из профиля i в профиль j можно перейти если выполняются условия:

- Можно положить горизонтальные домино. То есть там где в j профиле стоит 1, в i профиле должен стоять 0.
- Можно доложить в оставшиеся клетки вертикальные домино. То есть оставшиеся 0 в i профиле должны образовывать четные подстроки.

Задача о замощении домино



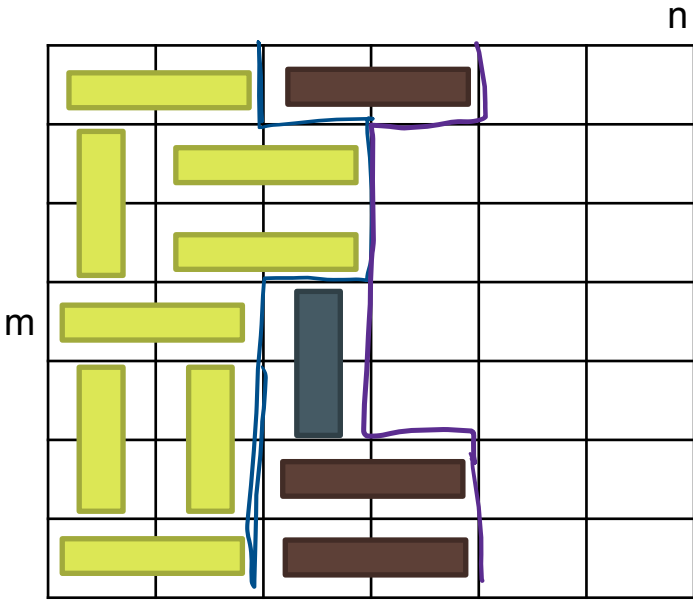
0 1
 1 → 0
 1 → 0
 0 0
 0 0
 0 1
 0 1

Будем для всех
 пар профилей хра-
 нить, можно ли
 между ними перейти.

Из профиля i в профиль j можно перейти если выполняются условия:

- Можно положить горизонтальные домино. То есть там где в j профиле стоит 1, в i профиле должен стоять 0.
- Можно доложить в оставшиеся клетки вертикальные домино. То есть оставшиеся 0 в i профиле должны образовывать четные подстроки.

Задача о замощении домино



0	1
1	→ 0
1	→ 0
0	0
0	0
0	1
0	1

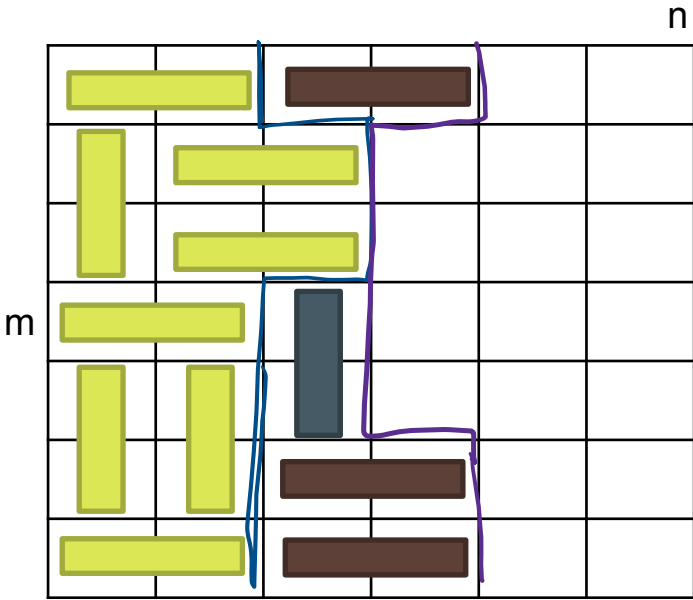
Будем для всех пар профилей хранить, можно ли между ними перейти.

$d[i][j]$ - возможен ли переход из i профиля в j -ый

Из профиля i в профиль j можно перейти если выполняются условия:

- Можно положить горизонтальные домино. То есть там где в j профиле стоит 1, в i профиле должен стоять 0.
- Можно доложить в оставшиеся клетки вертикальные домино. То есть оставшиеся 0 в i профиле должны образовывать четные подстроки.

Задача о замощении домино



0	1
1	→ 0
1	→ 0
0	0
0	0
0	1
0	1

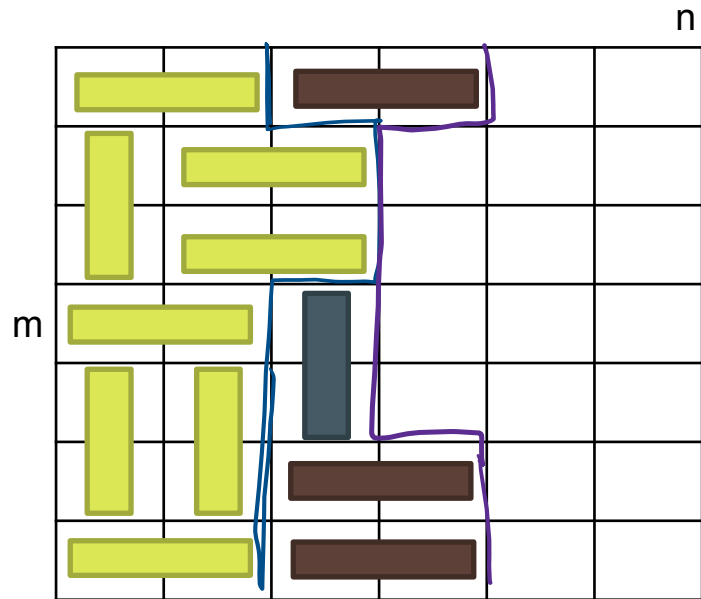
Будем для всех пар профилей хранить, можно ли между ними перейти.

$d[i][j]$ - возможен ли переход из i профиля в j -ый

Из профиля i в профиль j можно перейти если выполняются условия:

- Можно положить горизонтальные домино. То есть там где в j профиле стоит 1, в i профиле должен стоять 0.
- Можно доложить в оставшиеся клетки вертикальные домино. То есть оставшиеся 0 в i профиле должны образовывать четные подстроки.

Задача о замощении домино



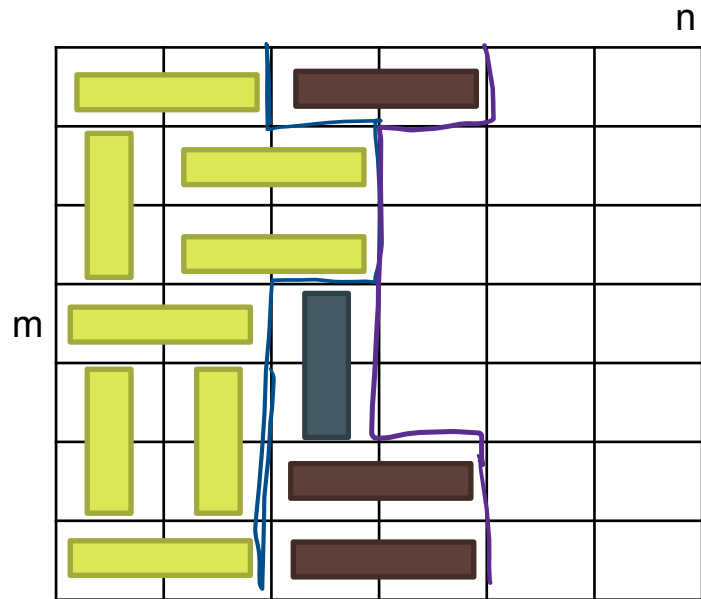
$A[i][p]$ – количество способов замостить первые i столбцов так, чтобы они заканчивались на p -ый профиль

$D[i][j]$ – можно ли из профиля i попасть в профиль j

Из профиля i в профиль j можно перейти если выполняются условия:

- Можно положить горизонтальные домино. То есть там где в j профиле стоит 1 , в i профиле должен стоять 0 .
- Можно доложить в оставшиеся клетки вертикальные домино. То есть оставшиеся 0 в i профиле должны образовывать четные подстроки.

Задача о замощении домино



$A[i][p]$ – количество способов замостить первые i столбцов так, чтобы они заканчивались на p -ый профиль

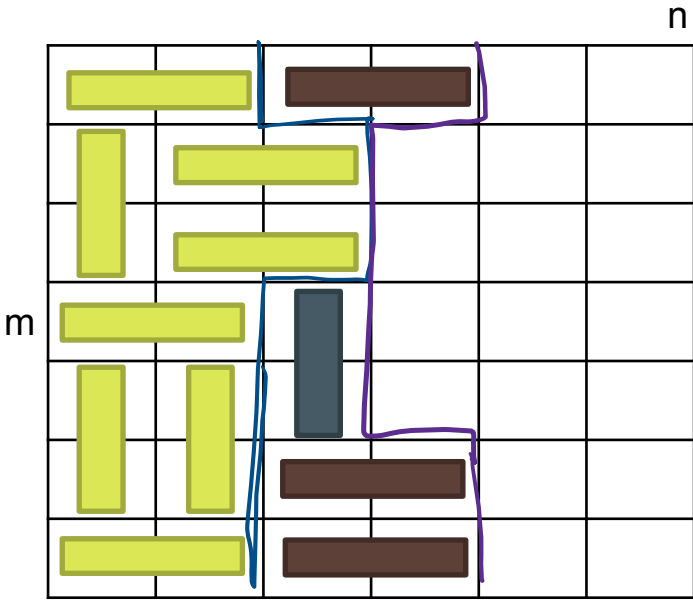
$D[i][j]$ – можно ли из профиля i попасть в профиль j

$A[i][p] - ?$

Из профиля i в профиль j можно перейти если выполняются условия:

- Можно положить горизонтальные домино. То есть там где в j профиле стоит 1, в i профиле должен стоять 0.
- Можно доложить в оставшиеся клетки вертикальные домино. То есть оставшиеся 0 в i профиле должны образовывать четные подстроки.

Задача о замощении домино



$A[i][p]$ – количество способов замостить первые i столбцов так, чтобы они заканчивались на p -ый профиль

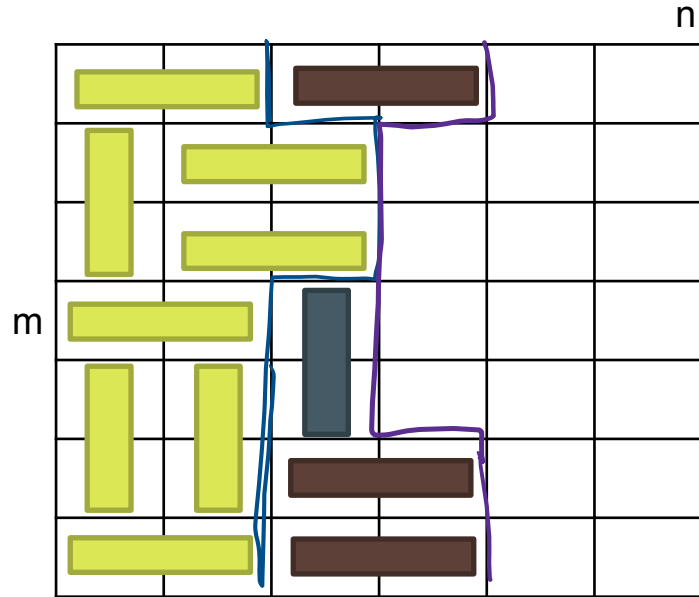
$D[i][j]$ – можно ли из профиля i попасть в профиль j

$$A[i][p] = \sum_{\substack{\text{по} \\ \text{всем} \\ \text{профилям} \\ j}} A[i-1][j] \cdot d[j][p]$$

Из профиля i в профиль j можно перейти если выполняются условия:

- Можно положить горизонтальные домино. То есть там где в j профиле стоит 1, в i профиле должен стоять 0.
- Можно доложить в оставшиеся клетки вертикальные домино. То есть оставшиеся 0 в i профиле должны образовывать четные подстроки.

Задача о замощении домино



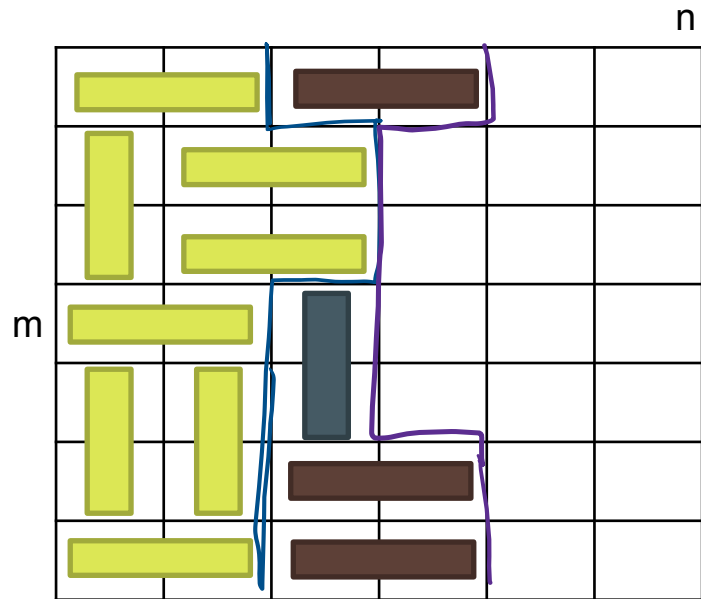
$A[i][p]$ – количество способов замостить первые i столбцов так, чтобы они заканчивались на p -ый профиль

$D[i][j]$ – можно ли из профиля i попасть в профиль j

$$A[i][p] = \sum_{\substack{\text{по} \\ \text{всем} \\ \text{профилям} \\ j}} A[i-1][j] \cdot d[j][p]$$

Где живет ответ?

Задача о замощении домино



$A[i][p]$ – количество способов замостить первые i столбцов так, чтобы они заканчивались на p -ый профиль

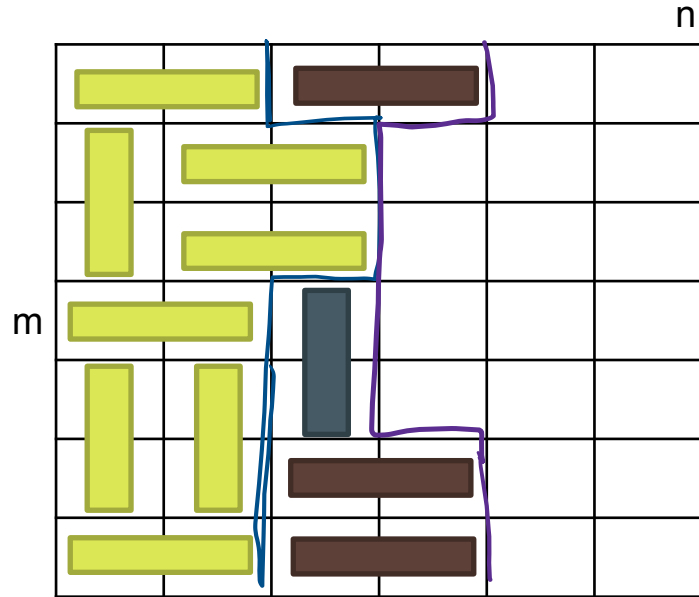
$D[i][j]$ – можно ли из профиля i попасть в профиль j

$$A[i][p] = \sum_{\substack{\text{по} \\ \text{всем} \\ \text{профилям} \\ j}} A[i-1][j] \cdot d[j][p]$$

Где живет ответ?

Ответом будет $\sum a[m][i]$, где i — профиль, который может быть последним (т.е. все группы из 0 имеют четные размеры).

Задача о замощении домино



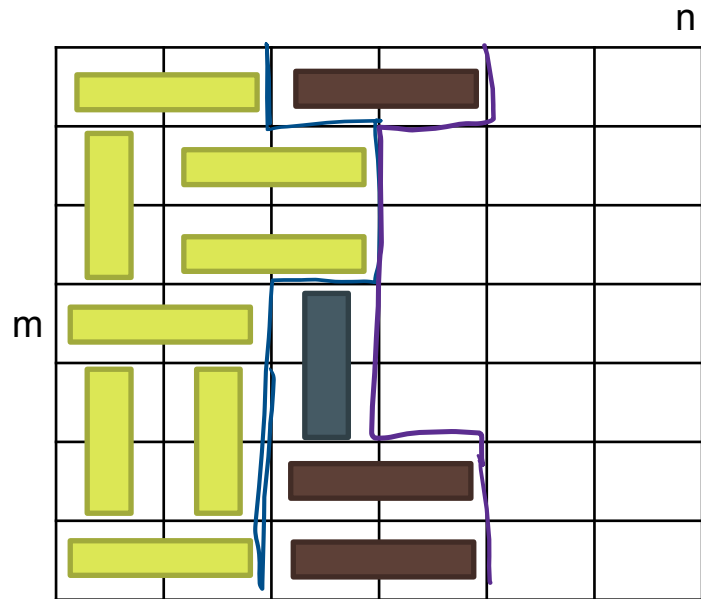
Сложность?

$A[i][p]$ – количество способов замостить первые i столбцов так, чтобы они заканчивались на p -ый профиль

$D[i][j]$ – можно ли из профиля i попасть в профиль j

$$A[i][p] = \sum_{\substack{\text{по} \\ \text{всем} \\ \text{профилям} \\ j}} A[i-1][j] \cdot d[j][p]$$

Задача о замощении домино



$A[i][p]$ – количество способов замостить первые i столбцов так, чтобы они заканчивались на p -ый профиль

$D[i][j]$ – можно ли из профиля i попасть в профиль j

$$A[i][p] = \sum_{\substack{\text{по} \\ \text{всем} \\ \text{профилям} \\ j}} A[i-1][j] \cdot d[j][p]$$

Сложность?

Оценка сложности: подсчет $d - 2^{2m}$, и подсчет $a - 2^{2m} n$ в итоге $O(2^{2m} n)$.

Задача о замощении домино

```
// n, m – размер таблицы
for i = 0..(1 << n) - 1
  for j = 0..(1 << n) - 1
    if можно перейти из i в j профиль
      d[i][j] = 1
    else
      d[i][j] = 0
```

для каждой пары профилей проверяем совместимость

```
a[0][0] = 1 // Так как мы можем начать только с профиля где все клетки 0
```

```
for k = 1..m - 1
  for i = 0..(1 << n) - 1
    for j = 0..(1 << n) - 1
      a[k][i] = a[k][i] + a[k - 1][j] · d[j][i]
```

составляем динамику для всех профилей

```
ans = 0
for i = 0..(1 << n) - 1
  if можно закончить i профилем
    ans = ans + a[m - 1][i]
return ans
```

проверяем профилем, что ОК где конца

Итоги

Рассмотрели три способа задавать динамику:

1. Динамика на префиксе (рюкзак, подпоследовательности, редакционное расстояние)
2. Динамика по подотрезкам (расстановка знаков)
3. Динамика по профилю (замощение доминошками)

Также стоит помнить, что динамику можно делать «вперед» (итеративно) и «назад» (рекурсивно).