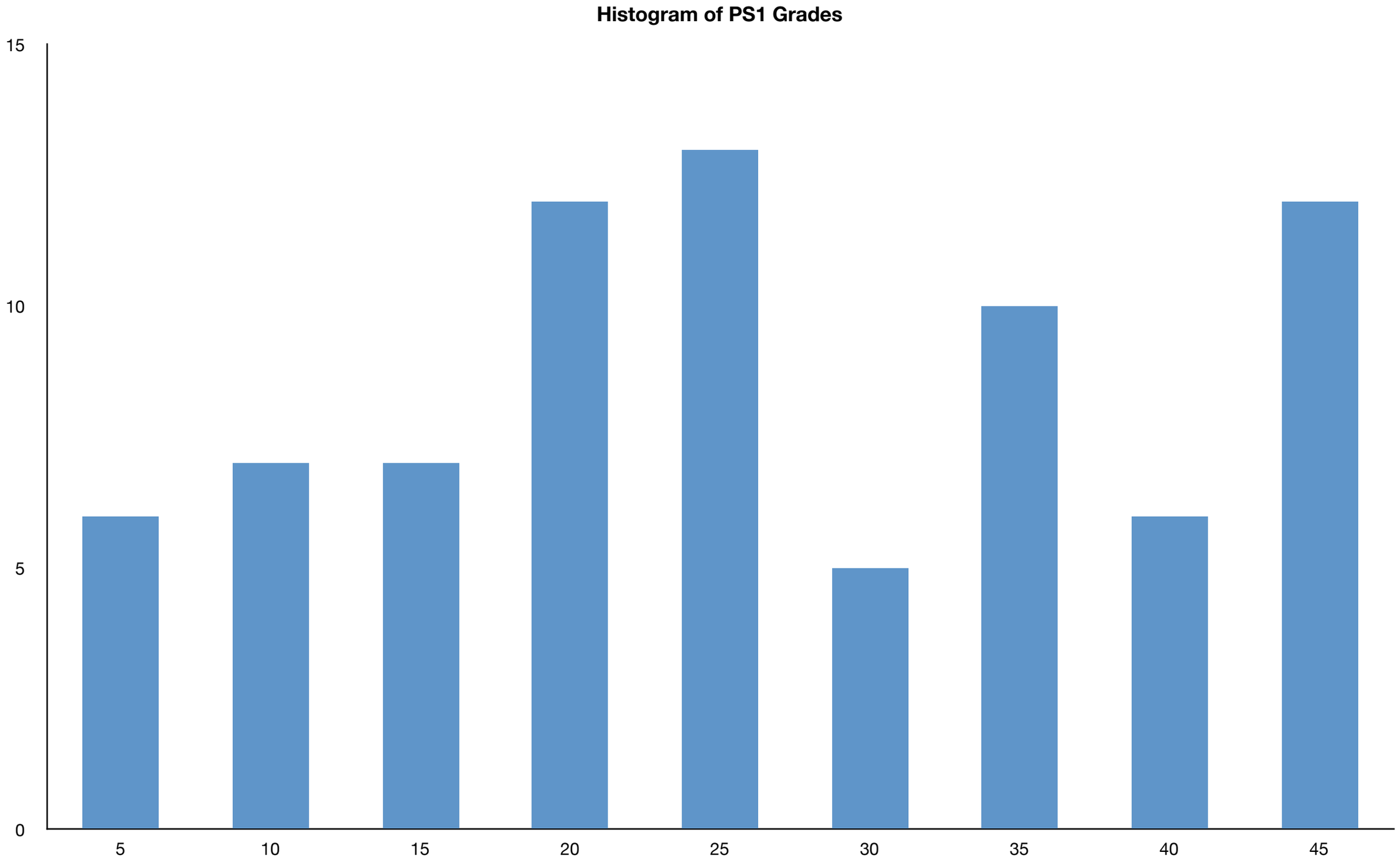
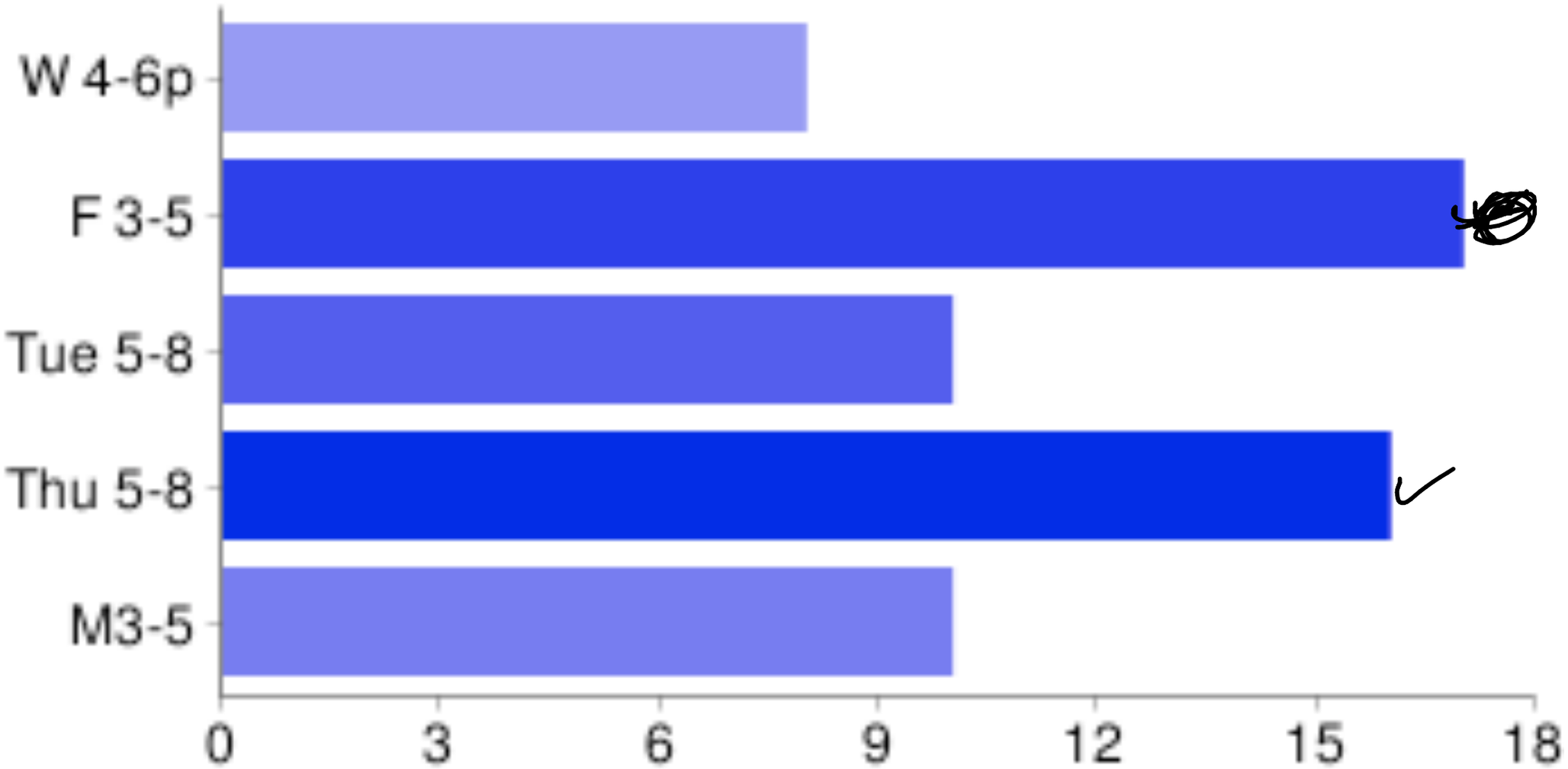
L&O

4102

Sep 17 2009

abhi shelat


Histogram of PS1 Grades
15
10
5
0
5
10
15
20
25
30
35
40
45



survey for office hours.

Dynamic Programming

two big ideas

recursive structure

+

remembering answers

wood cutting

http://snlm.files.wordpress.com/2008/08/bill-wakefield-and-carl-fie.gif

Log cutter dilemna

input to the problem: $n, (p_1, \ldots, p_n)$

goal: determine optimal plank widths

$i_1, i_2, \ldots, i_k$ that

$$\max p_{i_1} + p_{i_2} + \cdots + p_{i_k}$$

subject to

$$i_1 + i_2 + \cdots + i_k \leq n$$

Spot price for lumber

	1"	2"	3"	4"	5"
$	2.50	9	14	20	21

2.50 4.50 4.67 5$/in 4.20/in

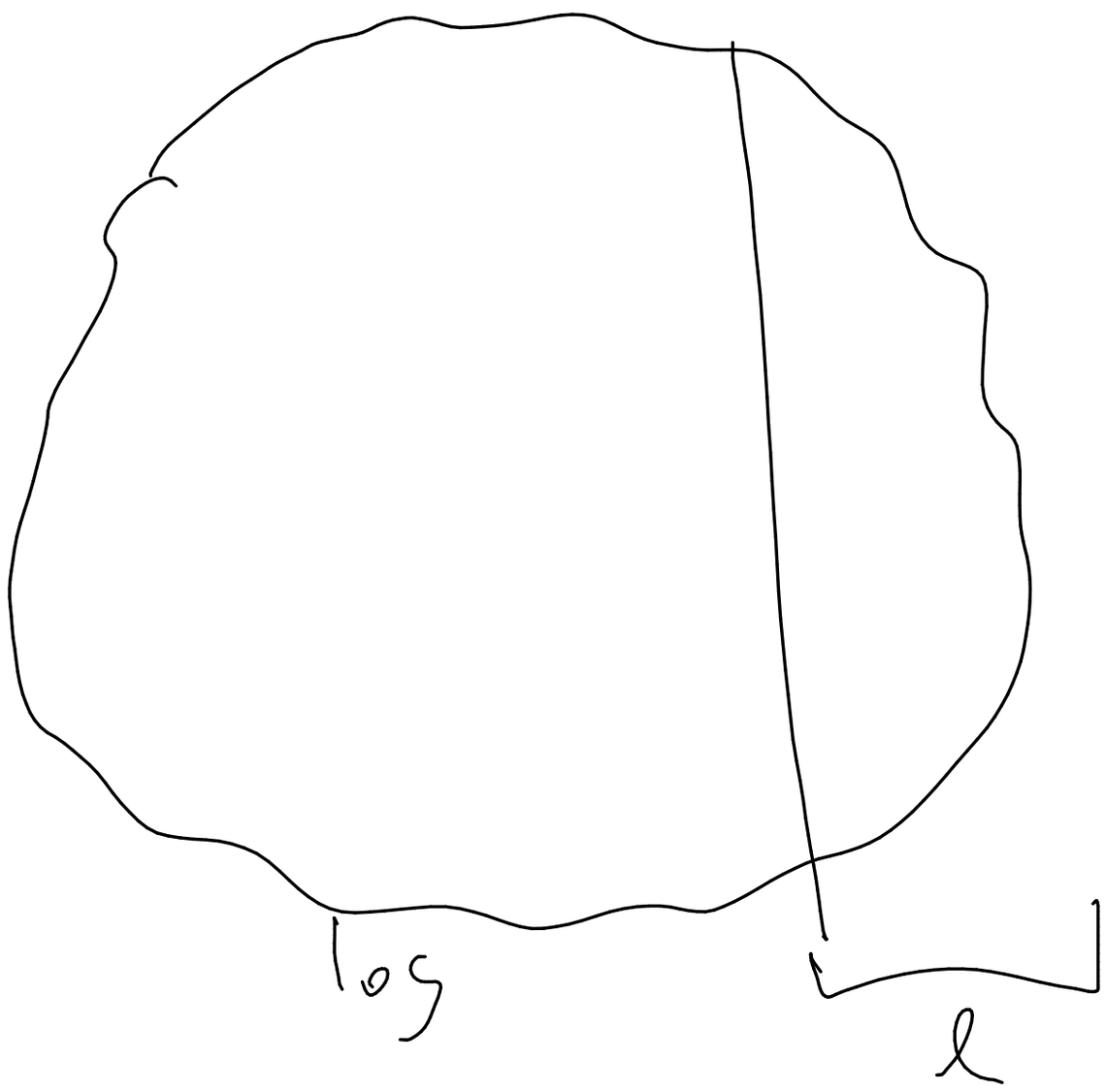
① Strategy 1: pick lagest price & slice.

n=5 21$

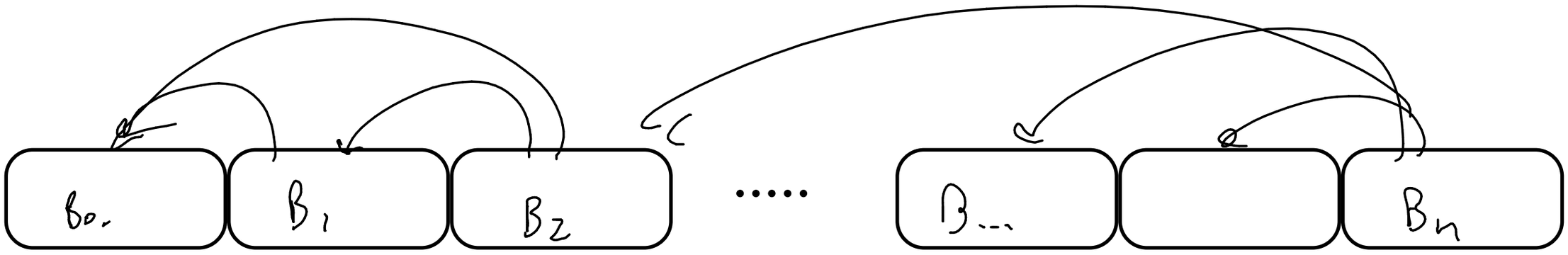
② compute "avg value". ie $\frac{\$}{\text{inch}}$. pick best value.

Solution equation

$Best\_n = Best_{n-l} + P_l$



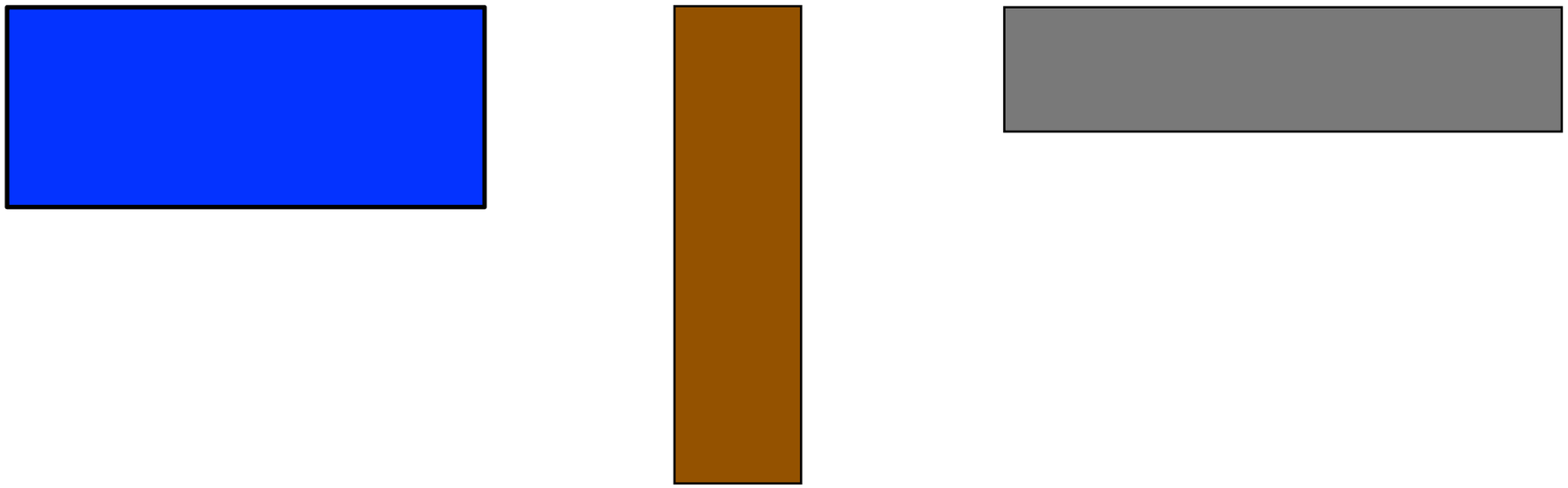
Better Approach



BestLogs($n, (p_1, \ldots, p_n)$)

```
if n<=0 return 0
for i=1 to n
 Best[i] = max{ pk + Best[k] }
```

Matrix



$$A_1 \cdot A_2 \cdot A_3$$

$$(A_1 \cdot A_2) \cdot A_3$$

7500

$$A_1 \cdot (A_2 \cdot A_3)$$

75,000

order matters

(for efficiency)

how many ways to compute?

$$A_1 A_2 A_3 \cdots A_n$$

$T(n) =$ # of ways to multiply a chain of n matricies.

$$T(n) = T(3) \cdot T(n-3) + T(4) \cdot T(n-4) + T(5) \cdot T(n-5) + \cdots$$

how many ways to compute?

$$A_1 A_2 A_3 \ldots A_n$$

$T(n) = T(1)\,T(n-1) + T(2)\,T(n-2) + \ldots +$

$T(n-2)\,T(2) + T(n-1)\cdot T(1).$

$$= \sum_{i=1}^{n-1} T(i)\,T(n-i) \quad > F_n = \Omega(\phi^n)$$

how do we solve it?

identify smaller instances of the problem

devise method to combine solutions

small # of different subproblems

solved them in the right order

optimal way to compute

$$A_1 A_2 A_3 \ldots A_n$$

- Suppose last op in the optimal order way to multiply

$r_1 \; (A_1 \ldots A_\ell) \cdot (A_{\ell+1} \ldots A_n)$ — c_ℓ, c_n, $r_{\ell+1}$

CostOfBest$_n$ = CostBest$(A_1 \ldots A_\ell)$ + CostBest$(A_{\ell+1} \ldots A_n)$ + $r_1 \cdot c_\ell \cdot c_n$

optimal way to compute

$$A_1 A_2 A_3 \ldots A_n$$

$$\text{BEST}_n = \text{BEST}_\ell + \text{BEST}_{n-\ell} + r_1 c_\ell c_n$$

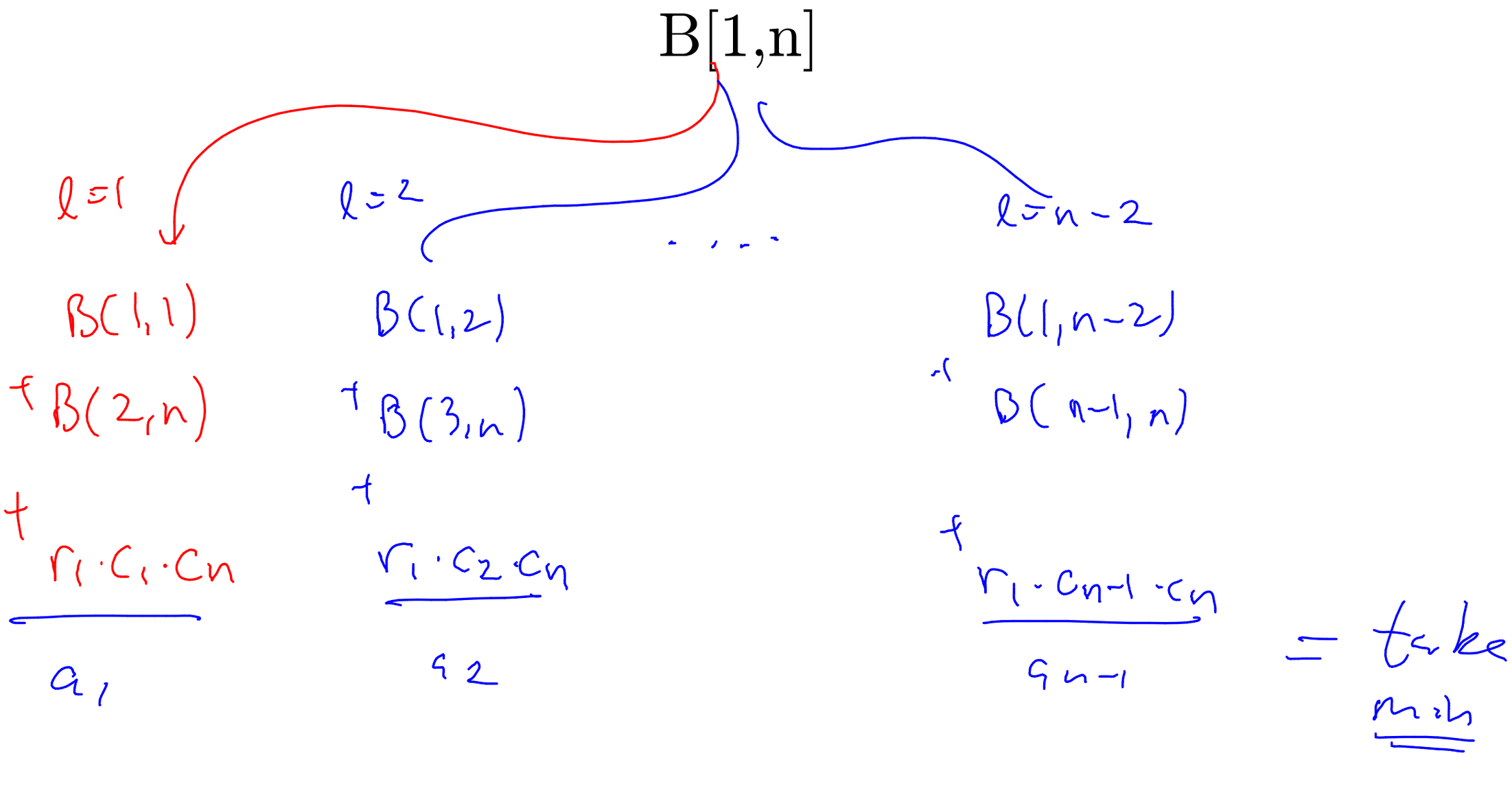
equation

how many choices are there for ℓ??

only n !!

optimal way to compute

$$A_1 A_2 A_3 \ldots A_n$$



optimal way to compute

$$A_1 A_2 A_3 \ldots A_n$$

B[1,n]

B[1,n-1] B[1,n-2] ... B[1,2] B[1,1]

B[n,n] B[n-1,n] ... B[3,n] B[2,n]

$R_1 C_{n-1} C_n$ $R_1 C_{n-2} C_n$ $R_1 C_1 C_n$

$$B(i,i) = 1$$

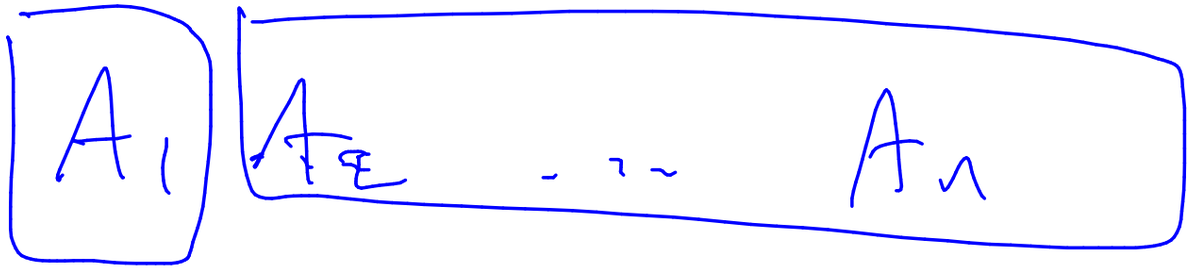
$$\underline{B(1,n)} = \min \begin{cases} B(1,1) + B(2,n) + r_1 \cdot c_1 \cdot c_n \\ B(1,2) + B(3,n) + r_1 \cdot c_2 \cdot c_n \\ \vdots \\ B(1,n-1) + B(n-1,n) + r_1 \cdot c_{n-1} \cdot c_n \end{cases}$$

$$B(i,i) = 1$$

$$B(1,n) = \min_{k=1 \text{ to } n-1} \begin{cases} B(1,1) + B(2,n) + r_1 c_1 c_n & k=1 \\ B(1,2) + B(3,n) + r_1 c_2 c_n \\ \vdots \\ B(1,n-1) + B(n,n) + r_1 c_{n-1} c_n \end{cases}$$

$$B(i,j) =$$

$$\begin{cases} 0 \text{ if } i = j \\ \min_k \{B(i,k) + B(k+1,j) + r_i c_k c_j \end{cases}$$



B(1,1)

0

B(2,n)

how did we solve it?

identified smaller instances of the problem ✓

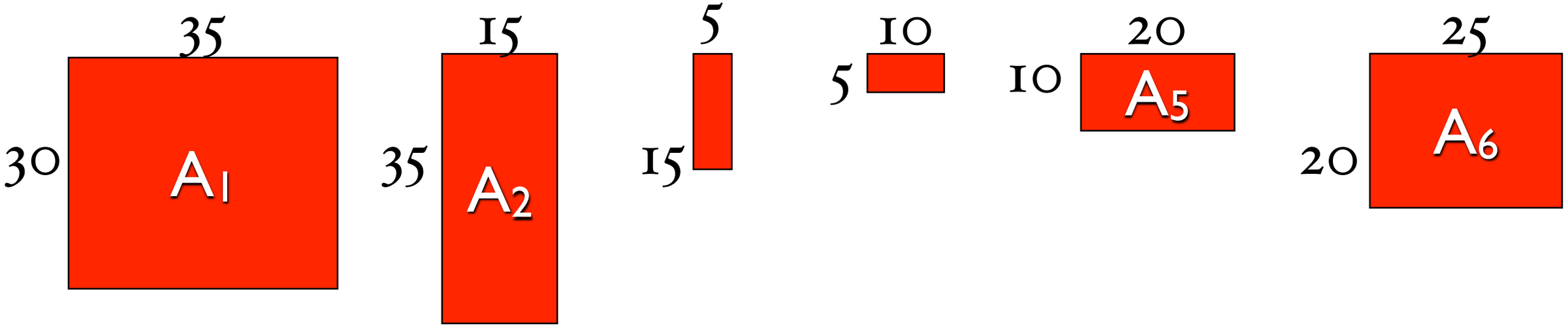
devised method to combine solutions ✓

small # of different subproblems –

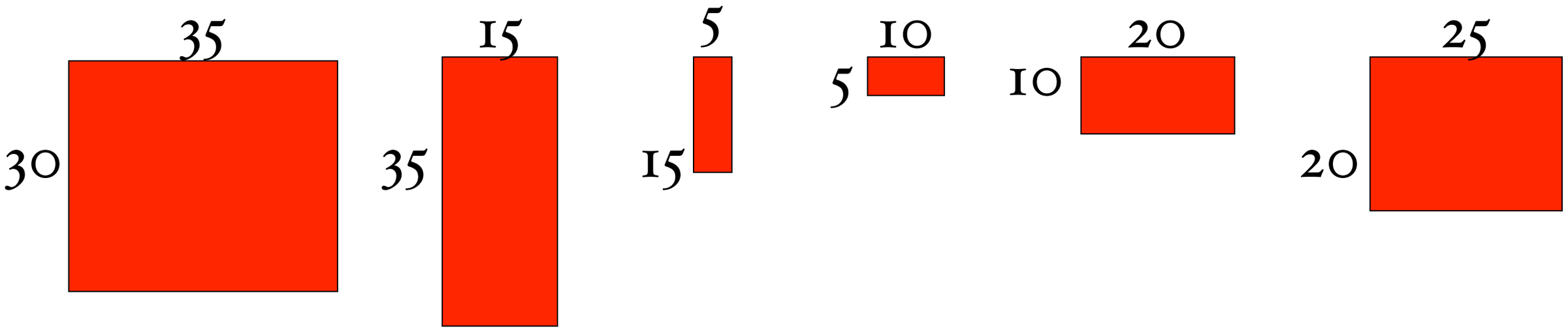
solved them in the right order

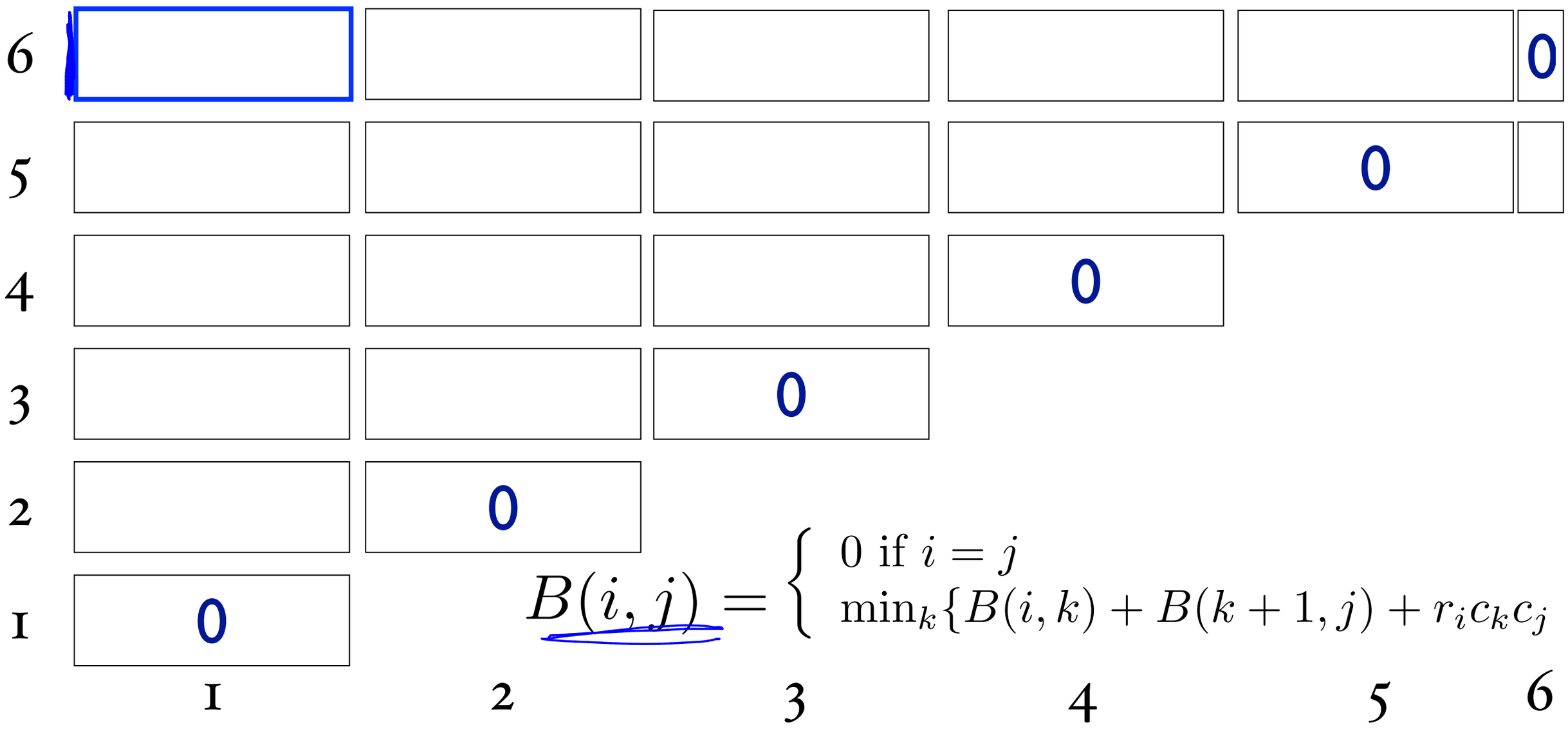
$$B(i,j) = \begin{cases} 0 \text{ if } i = j \\ \min_k \{B(i,k) + B(k+1,j) + r_i c_k c_j \end{cases}$$

which order to solve?

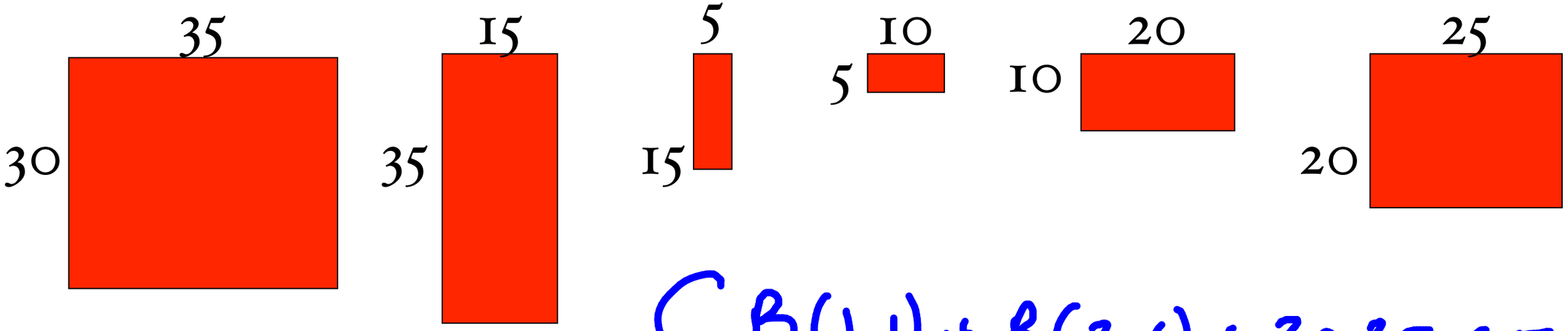


$$B(1,2) = 30 \cdot 35 \cdot 15 = 15750$$





$$B(i,j) = \begin{cases} 0 \text{ if } i = j \\ \min_k \{B(i,k) + B(k+1,j) + r_i c_k c_j \end{cases}$$



$C(1,6) =$ min

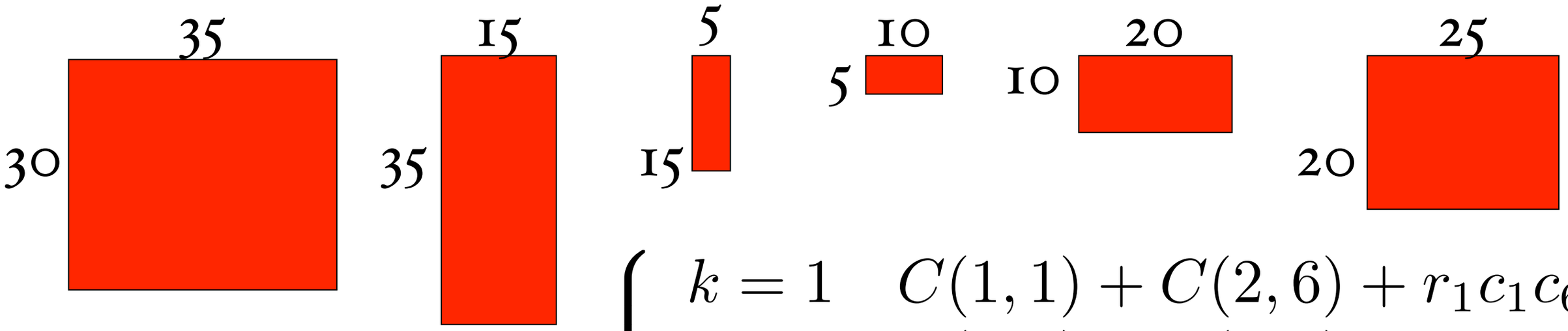
$B(1,1) + B(2,6) + 30 \cdot 35 \cdot 25$

$B(1,2) + B(3,6) + 30 \cdot 15 \cdot 25$

$B(1,3) + B(4,6) + 30 \cdot 5 \cdot 25$

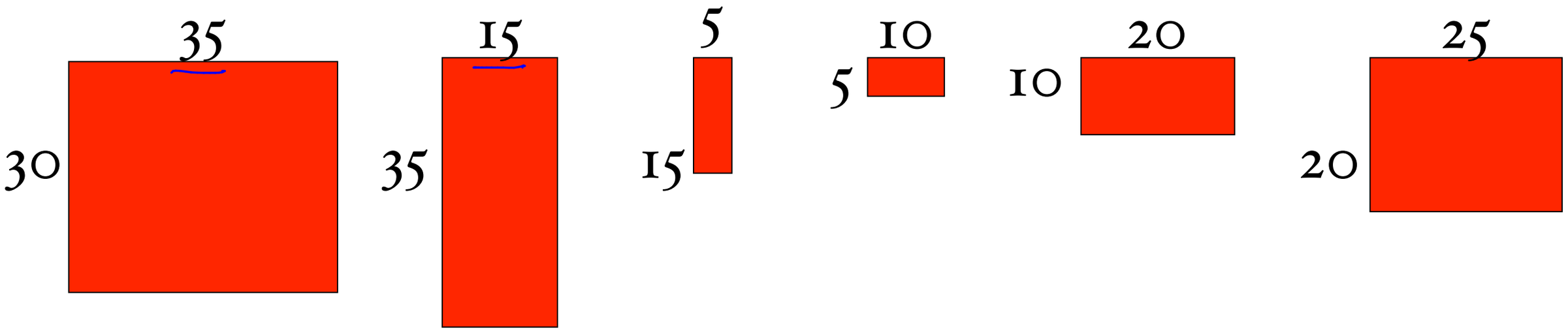
$B(1,4) + B(5,6) + 30 \cdot 10 \cdot 25$

$B(1,5) + B(6,1) + 30 \cdot 20 \cdot 25$

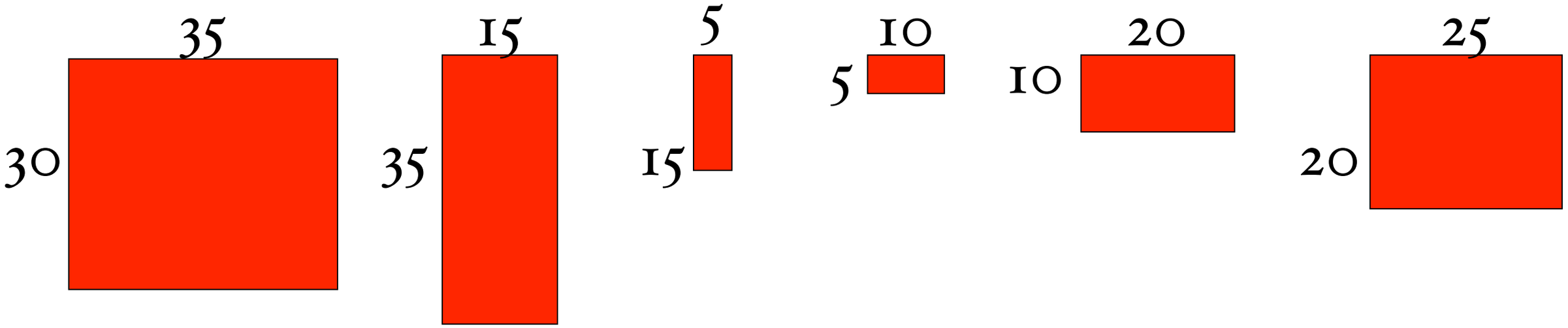


$$C(1,6) = \min \begin{cases} k=1 & C(1,1)+C(2,6)+r_1c_1c_6 \\ k=2 & C(1,2)+C(3,6)+r_1c_2c_6 \\ k=3 & C(1,3)+C(4,6)+r_1c_3c_6 \\ k=4 & C(1,4)+C(5,6)+r_1c_4c_6 \\ k=5 & C(1,5)+C(6,6)+r_1c_5c_6 \end{cases}$$

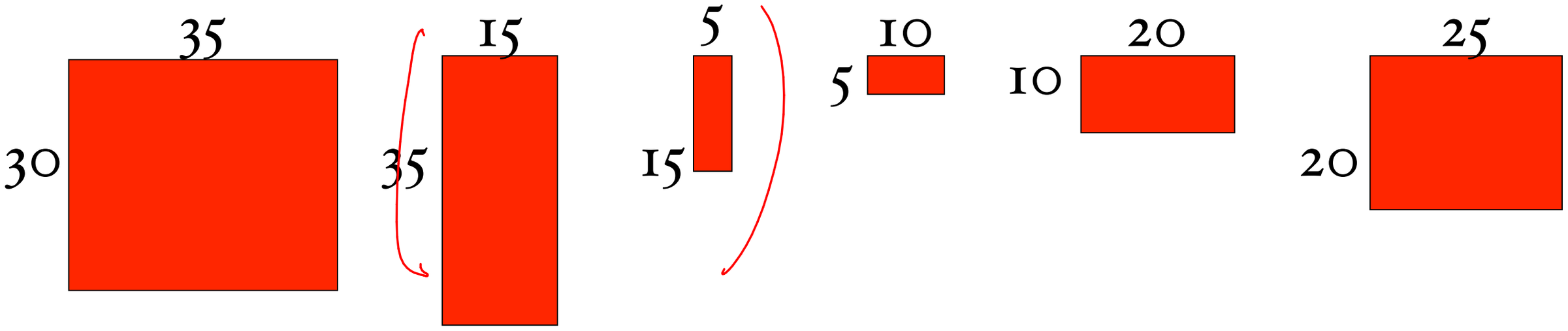
6						
5				0		
4				0		
3			0			
2		0				
1	0					
	1	2	3	4	5	6


35
30
15
35
5
15
10
5
20
10
25
20

6
5
4
3
2
1
0
0
0
0
0
0
30·35·15 =
1
2
3
4
5
6



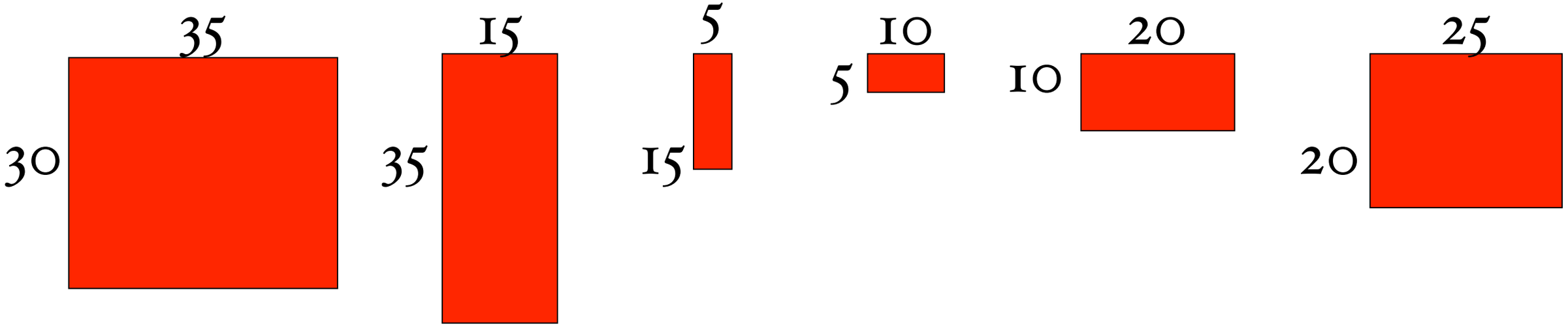
6					10*20*25 = 5000	0
5				5*10*20 = 1000	0	
4			15*5*10 = 750	0		
3		35*15*5 = 2625	0			
2	30*35*15 = 15750	0				
1	0					
	1	2	3	4	5	6



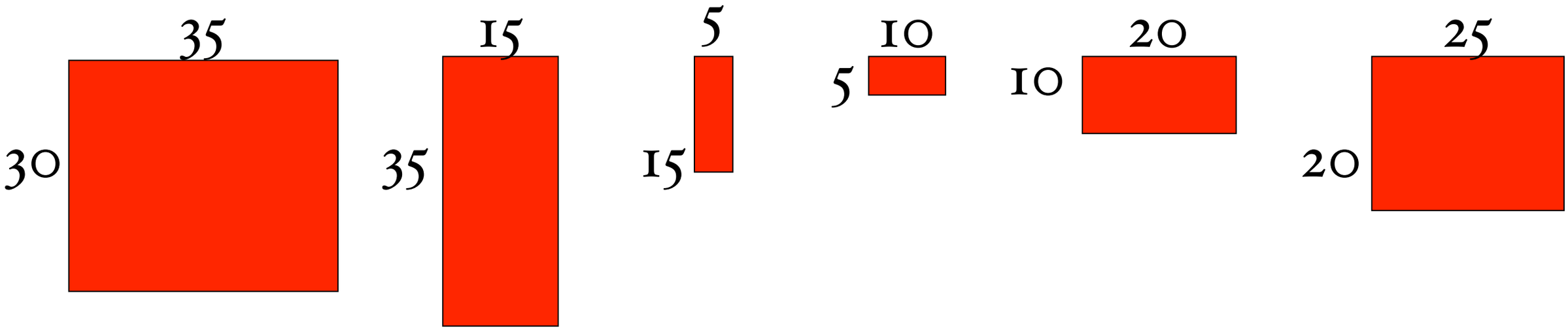
$$B(1,3) = \min \begin{cases} \overset{0}{B(1,1)} + \overset{2625}{B(2,3)} + \overset{5250}{30 \cdot 35 \cdot 5} = 7875 * \\ B(1,2) + \overset{0}{B(3,3)} + 30 \cdot 15 \cdot 5 = 17000 \end{cases}$$

15750 2250

3	7875	35*15*5 = 2625	0
2	30*35*15 = 15750	0	
1	0		
	1	2	3 4 5 6

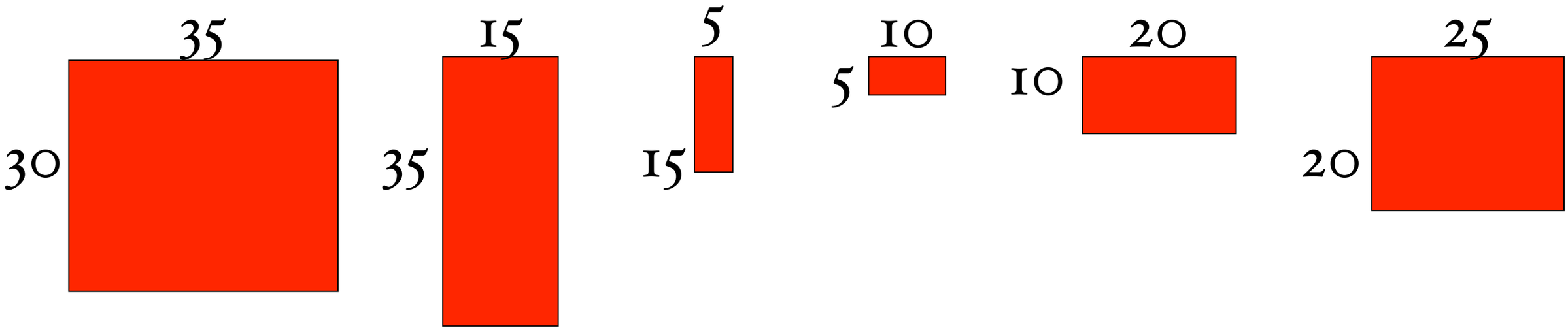


6		10500	5375	3500	10*20*25 = 5000	0
5	11875	7125	2500	5*10*20 = 1000	0	
4	9375	4375	15*5*10 = 750	0		
3	7875	35*15*5 = 2625	0			
2	30*35*15 = 15750	0				
1	0					
	1	2	3	4	5	6



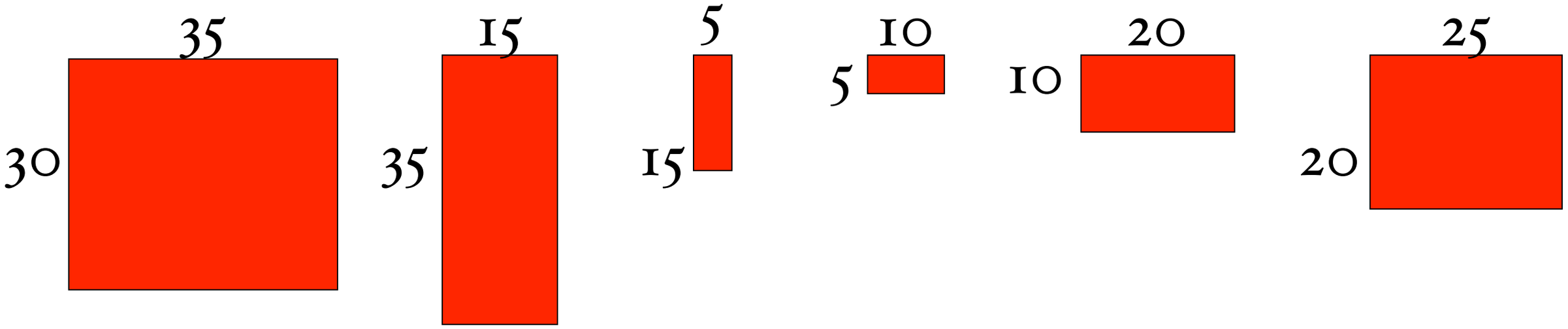
6 

$$C(1,6) = \min \begin{cases} k=1 & C(1,1) + C(2,6) + r_1 c_1 c_6 \\ k=2 & C(1,2) + C(3,6) + r_1 c_2 c_6 \\ k=3 & C(1,3) + C(4,6) + r_1 c_3 c_6 \\ k=4 & C(1,4) + C(5,6) + r_1 c_4 c_6 \\ k=5 & C(1,5) + C(6,6) + r_1 c_5 c_6 \end{cases}$$



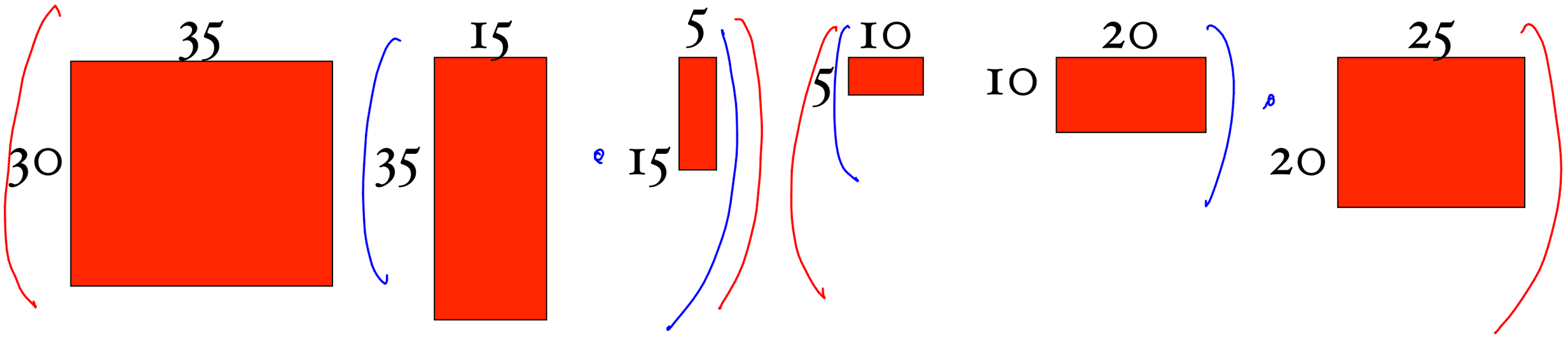
6 

$$C(1,6) = \min \begin{cases} k=1 & 0 + 10500 + 30 \cdot 35 \cdot 25 \\ k=2 & 15750 + 5375 + 30 \cdot 15 \cdot 25 \\ k=3 & 7875 + 3500 + 30 \cdot 5 \cdot 25 \\ k=4 & 9375 + 5000 + 30 \cdot 10 \cdot 25 \\ k=5 & 11875 + 0 + 30 \cdot 20 \cdot 25 \end{cases}$$

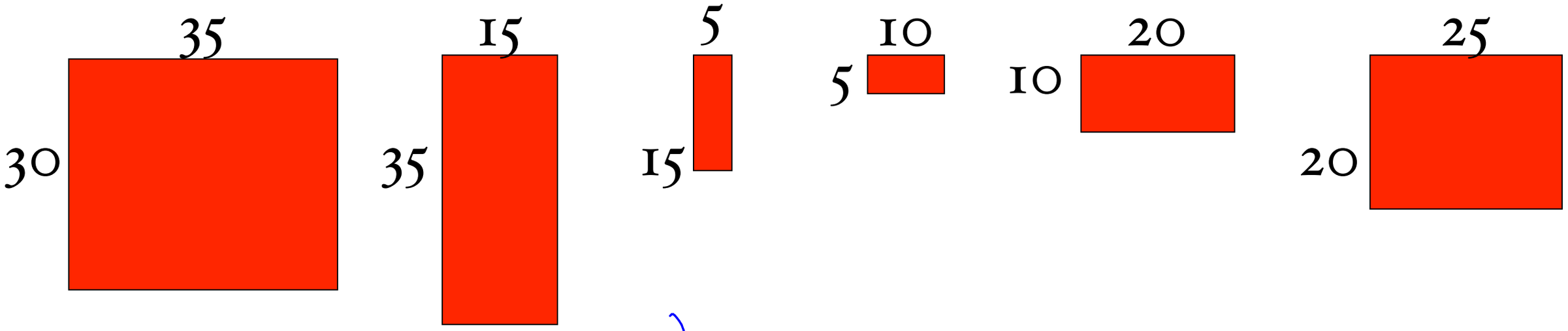


6

$$C(1,6) = \min \begin{cases} k=1 & 0 + 10500 + 26250 \\ k=2 & 15750 + 5375 + 11250 \\ k=3 & 7875 + 3500 + 3750 \\ k=4 & 9375 + 5000 + 7500 \\ k=5 & 11875 + 0 + 15000 \end{cases}$$



6	15125	10500	5375	3500	10*20*25 = 5000	0
5	11875	7125	2500	5*10*20 = 1000	0	
4	9375	4375	15*5*10 = 750	0		
3	7875	35*15*5 = 2625	0			
2	30*35*15 = 15750	0				
1	0					
	1	2	3	4	5	6



$\Theta(n)$

6	15125 3	10500	5375	3500 ★	10*20*25 = 5000	0
5	11875	7125	2500	5*10*20 = 1000 ★	0	
4	9375	4375	15*5*10 = 750	0		
3	7875 ★	35*15*5 = 2625 ★	0			
2	30*35*15 = 15750	0				
1	0					
	1	2	3	4	5	6

matrix-chain-mult(p)

initialize array m[x,y] to zero

compute along diagonals SE to NW {

$$B(i,j) = \min_{\substack{k=i \\ \text{to } j-1}} \begin{cases} B(i,k) + B(k+1,j) + r_i \cdot c_k \cdot c_j \\ B(i,i) = 0 \end{cases}$$

}

- $\Theta(n^2)$ squares to fill in $\Rightarrow$ $\underline{\Theta(n^3)}$
  $O(n)$ for each square

matrix-chain-mult(p)

initialize array m[x,y] to zero

starting at diagonal, working towards upper-left

compute m[i,j] according to

$$\begin{cases} 0 \text{ if } i = j \\ \min_k \{B(i,k) + B(k+1,j) + r_i c_k c_j \end{cases}$$

running time?

initialize array m[x,y] to zero

starting at diagonal, working towards upper-left

compute m[i,j] according to

$$\begin{cases} 0 \text{ if } i = j \\ \min_k \{B(i,k) + B(k+1,j) + r_i c_k c_j \end{cases}$$

Typesetting

The first problem, which we denote the *set membership* proof, occurs for instance in the context of anonymous credentials. Consider a user who is issued a credential containing a number of attributes such as address. Further assume the user needs to prove that she lives in a European capital. Thus, we are given a list of all such cities and the user has to show that she possesses a credential containing one of those cities as address (without of course, leaking the city the user lives in). Or, consider a user who has a subscription to a journal (e.g., the news and the sports section). Further assume that some general sections are to all subscribers of a list of sections. Thus, using our protocol, the user can efficiently show that she is a subscriber to one of the required kinds.

The second problem, which we denote the *range proof*, also occurs often in anonymous credential and e-cash scenarios. For example, a user with passport credential might wish to prove that her age is within some range, e.g. greater than 18, or say between 13 and 18 in the case of a teen-community website. This problem is a special case of the set membership proof. Since the elements of the set occur in consecutive order, special techniques can be applied.

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to heaven, we were all going direct the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to heaven, we were all going direct the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.

First rule of typesetting

never print in the margin!

 are simply not allowed

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to heaven, we were all going direct the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.

Second rule of typesetting

avoid big ugly whitespaces (slack)

_____ is....

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to heaven, we were all going direct the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.

It was the best of times, it was the
worst of times, it was the age of wisdom,
it was the age of foolishness, it was the
epoch of belief, it was the epoch of
incredulity, it was the season of Light,
it was the season of Darkness, it was the
spring of hope, it was the winter of
despair, we had everything before us, we
had nothing before us, we were all going
direct to heaven, we were all going direct
the other way - in short, the period was
so far like the present period, that some
of its noisiest authorities insisted on
its being received, for good or for evil,
in the superlative degree of comparison
only.

pretty print problem

input: $W = \{w_1, w_2, w_3, \ldots, w_n\}$
$C = \{c_1, c_2, c_3, \ldots, c_n\}$
M

output: $L = (w_1, \ldots, w_{\ell_1}), (w_{\ell_1+1}, \ldots, w_{\ell_2}), \ldots, (w_{\ell_{x+1}, \ldots, w_n})$

such that

pretty print problem

input: $W = \{w_1, w_2, w_3, \ldots, w_n\}$
$C = \{c_1, c_2, c_3, \ldots, c_n\}$
M

output: $L = (w_1, \ldots, w_{\ell_1}), (w_{\ell_1+1}, \ldots, w_{\ell_2}), \ldots, (w_{\ell_{x+1}, \ldots, w_n})$

such that

typesetting problem

input: $W = \{w_1, w_2, w_3, \ldots, w_n\}$
$C = \{c_1, c_2, c_3, \ldots, c_n\}$
M

output: $L = (w_1, \ldots, w_{\ell_1}), (w_{\ell_1+1}, \ldots, w_{\ell_2}), \ldots, (w_{\ell_{x+1}, \ldots, w_n})$

such that $\left(\sum_{j=\ell_i+1}^{\ell_{i+1}} c_j \right) + (\ell_{i+1} - \ell_i - 1) \leq M$

$\min \sum S_i^2$

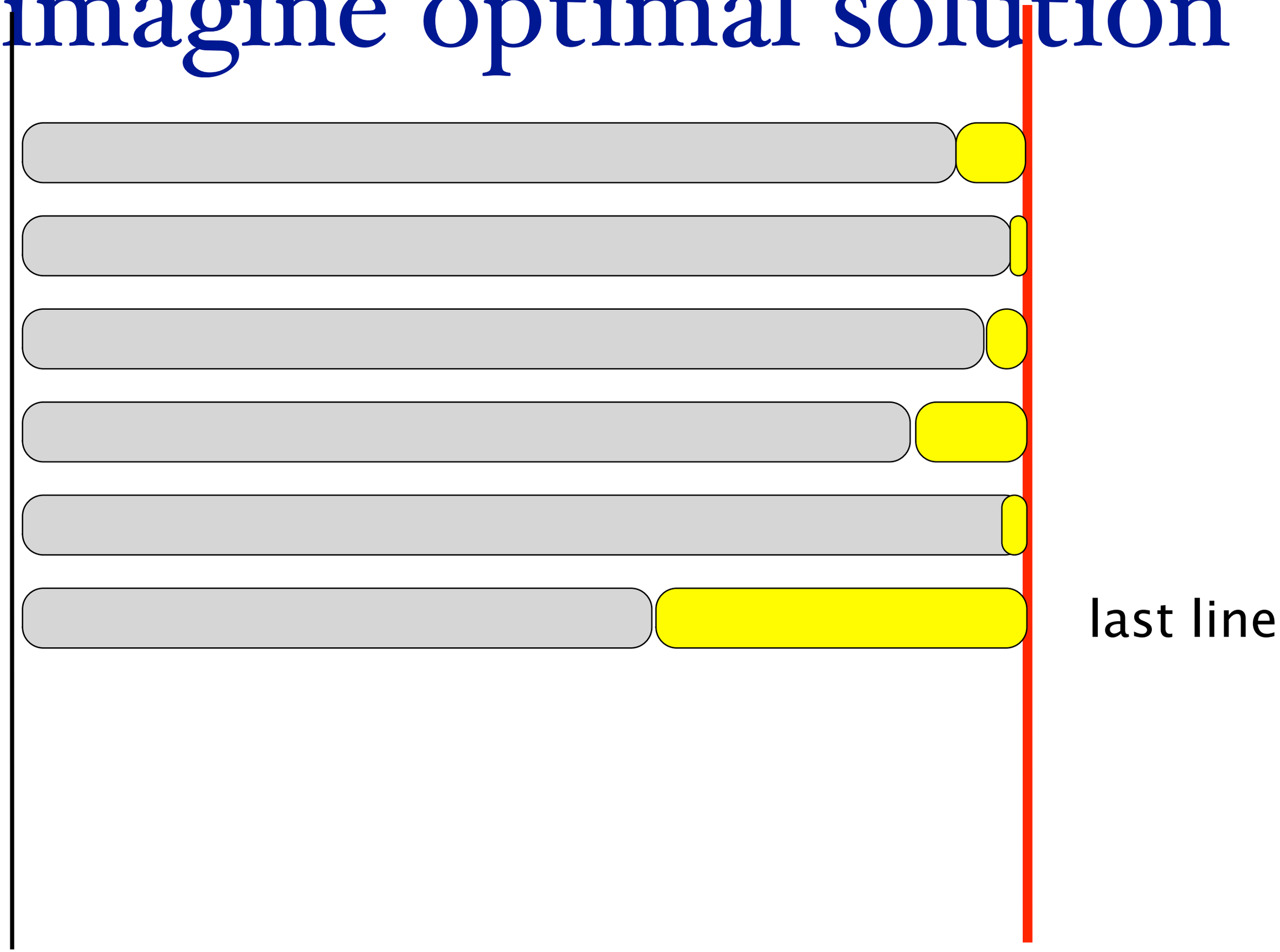
$S_i = M - \left(\sum_{j=\ell_i+1}^{\ell_{i+1}} c_j \right) - (\ell_{i+1} - \ell_i - 1)$

how to solve

define the right variable:

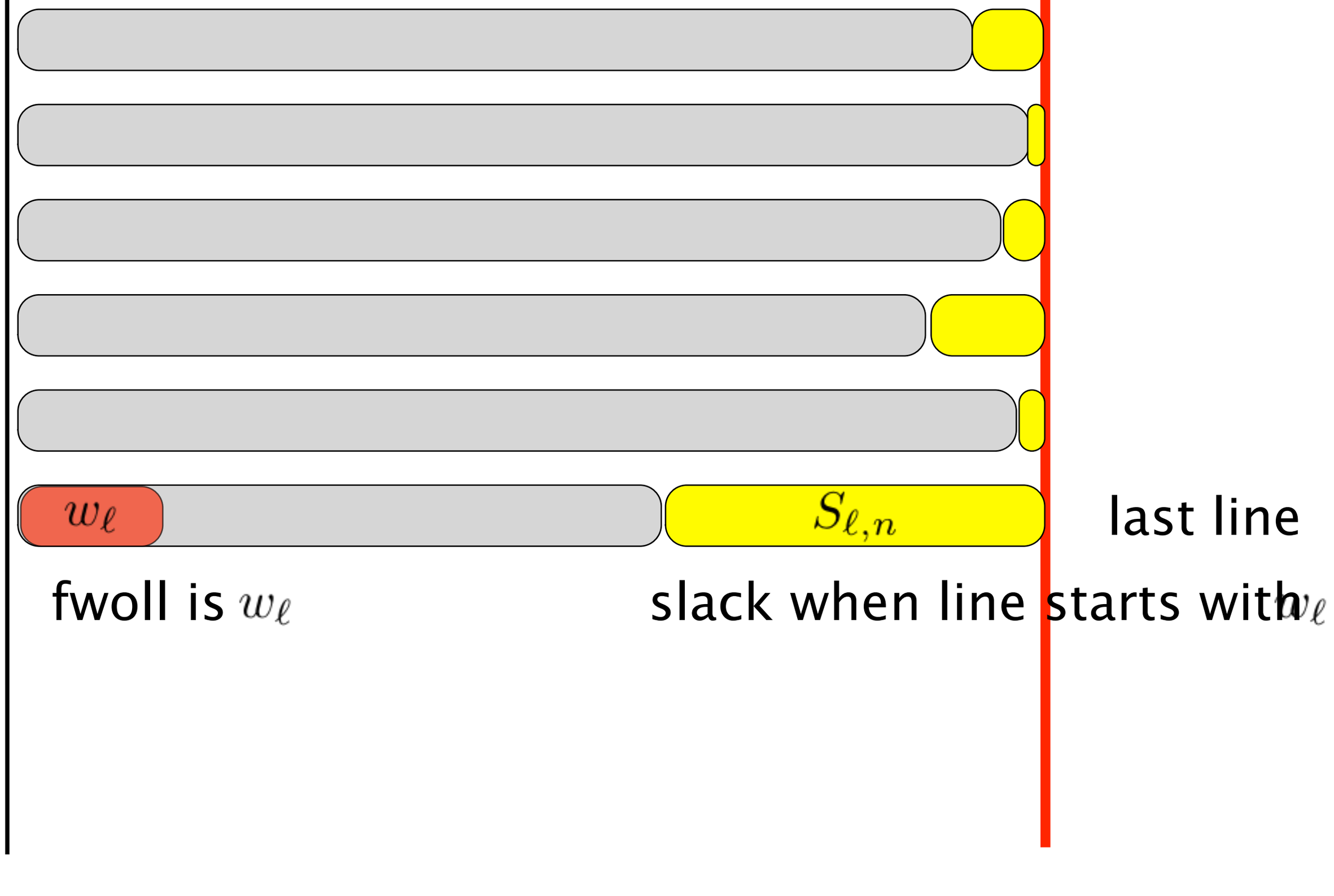
imagine optimal solution

imagine optimal solution

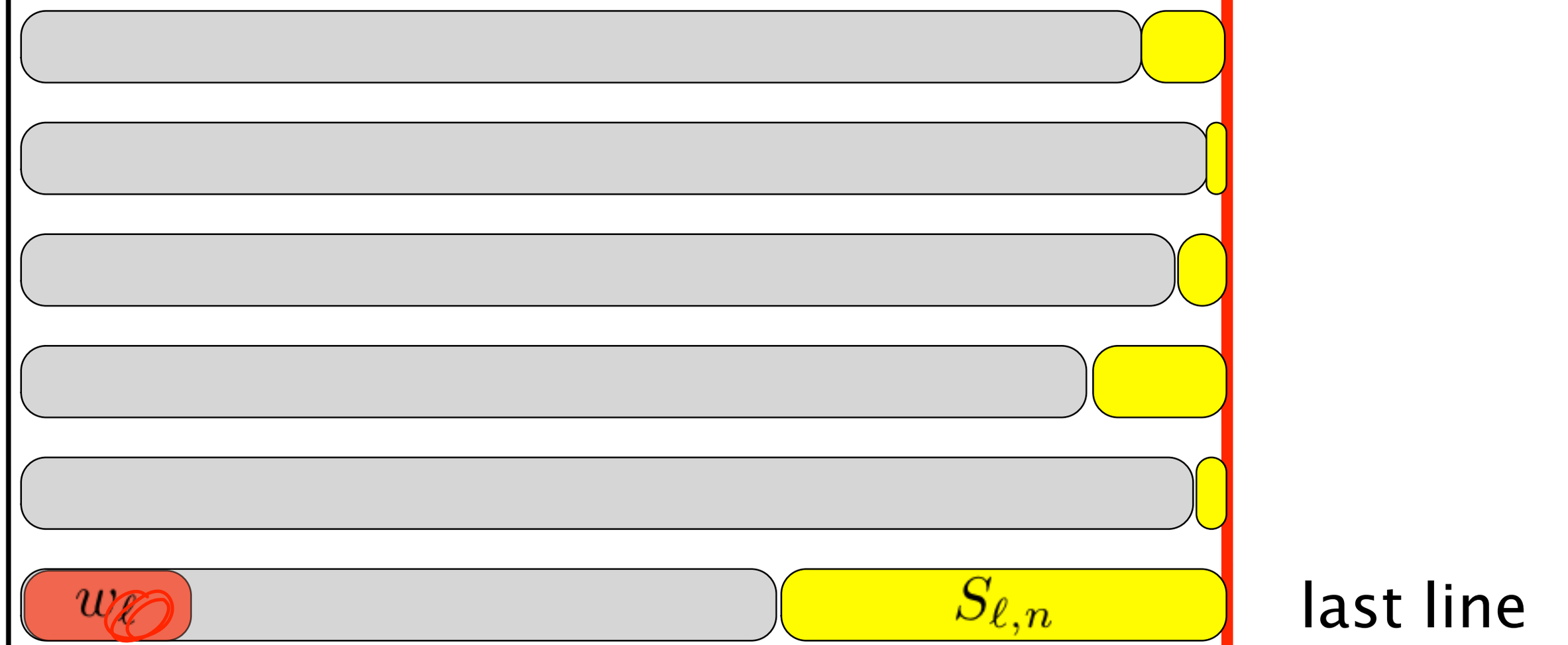


some word has to be the first-word-of-last-line (fwoll)

imagine optimal solution



imagine optimal solution



$$\mathrm{BEST}_n = \mathrm{BEST}_{\ell-1} + S_{\ell,n}^2$$

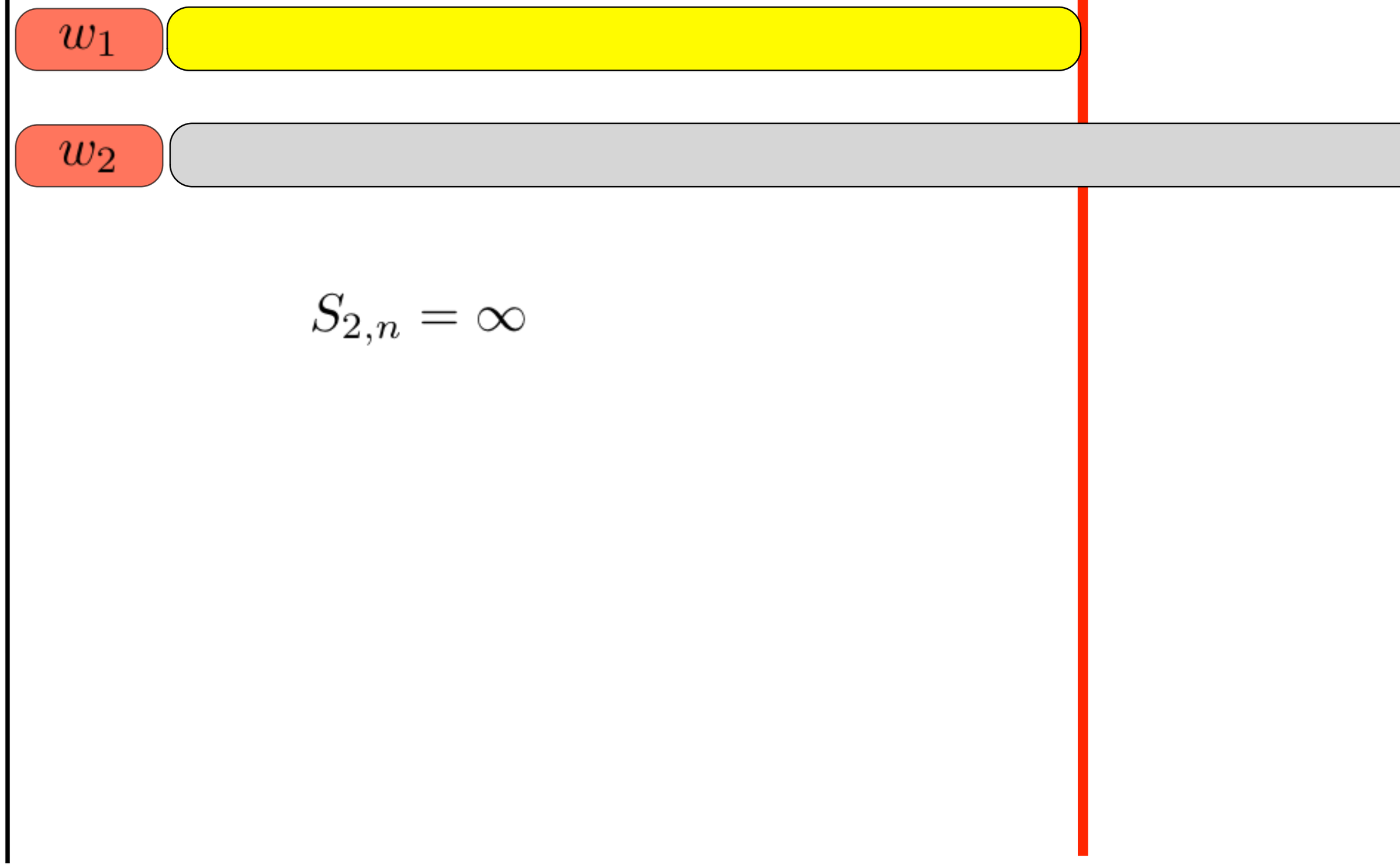
how many candidates are there for office of fwoll?

is w_1 fwoll?

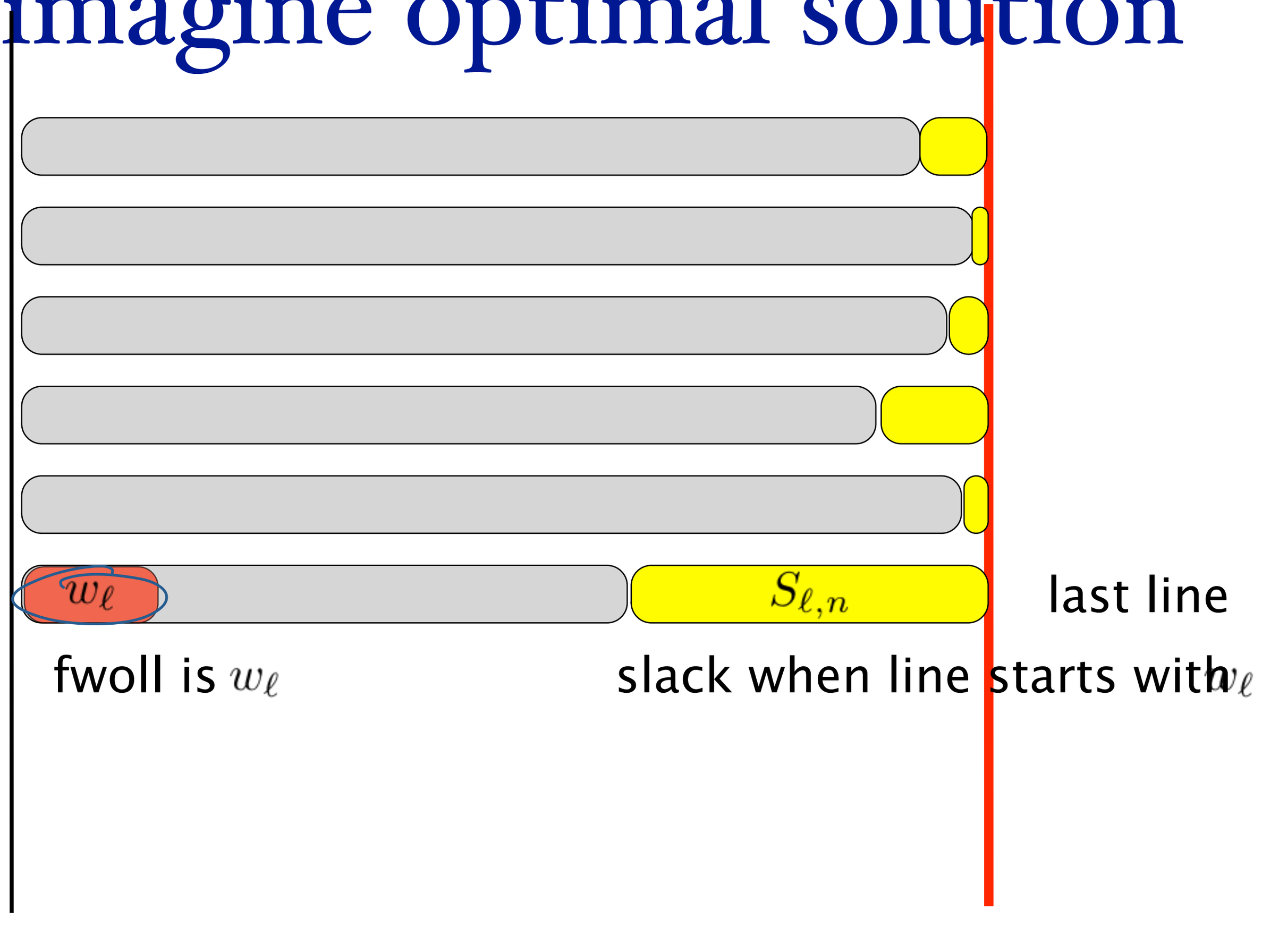
there is no slack (no solution even)
because words go beyond edge!

define $S_{1,n} = \infty$ if this happens

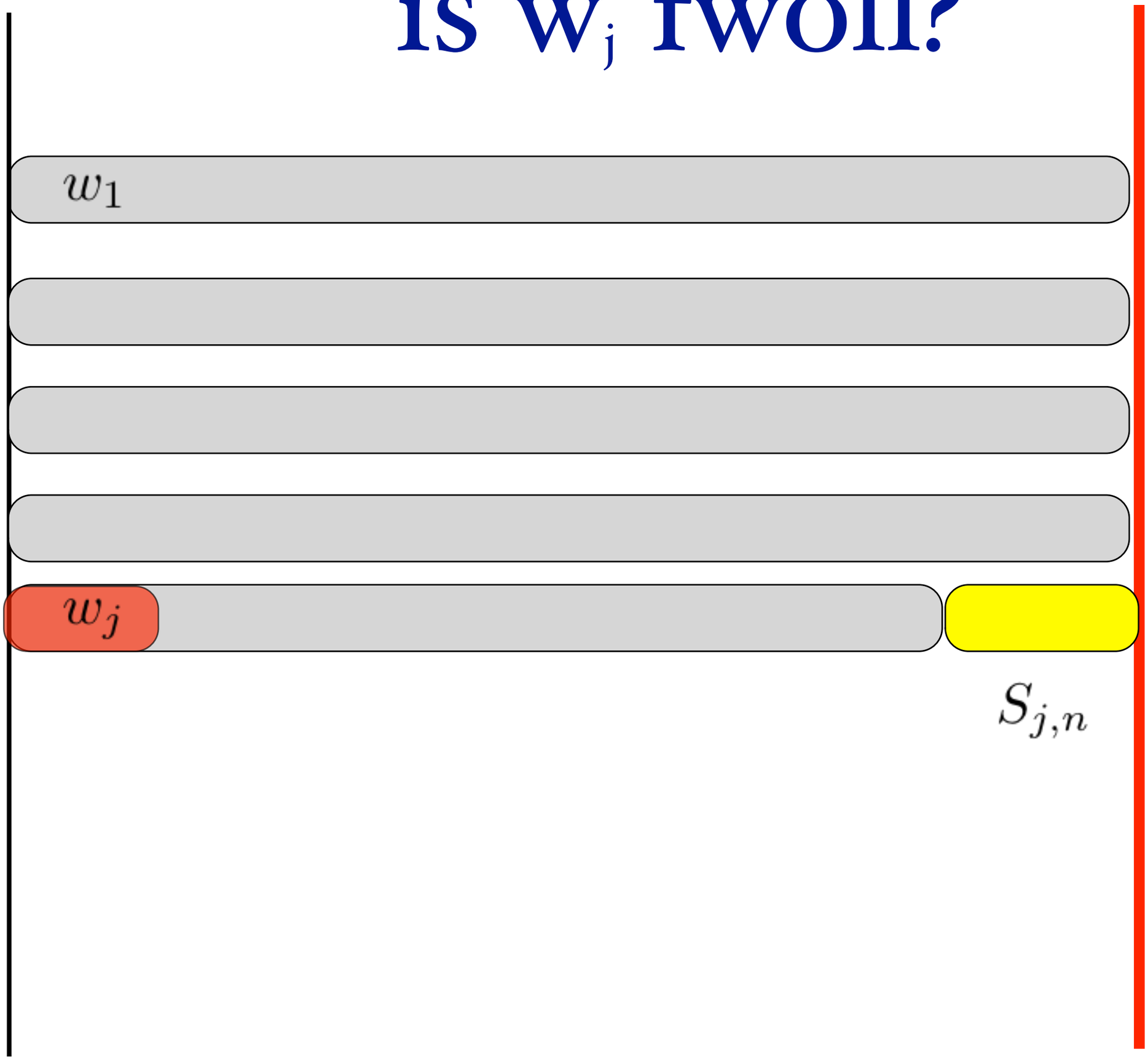
is w_2 fwoll?



imagine optimal solution



is w_j fwoll?



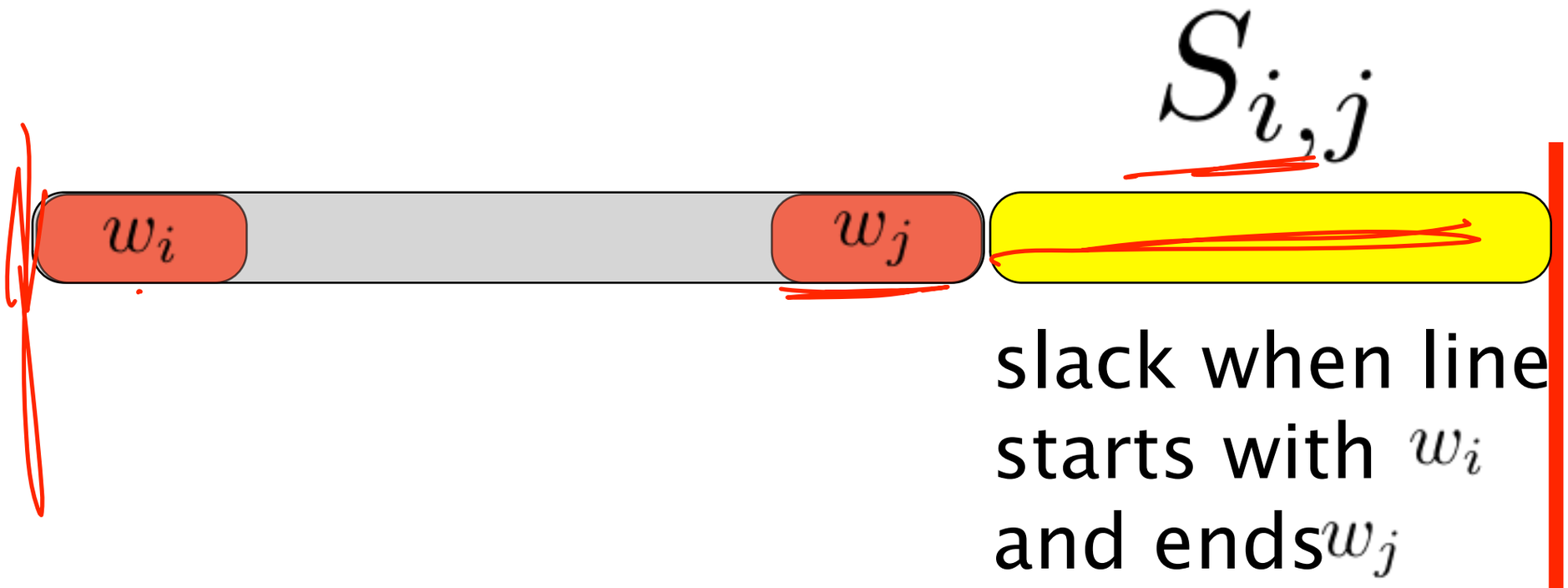
which word is fwoll?

$$\mathrm{BEST}_n = \min \left\{ \right.$$

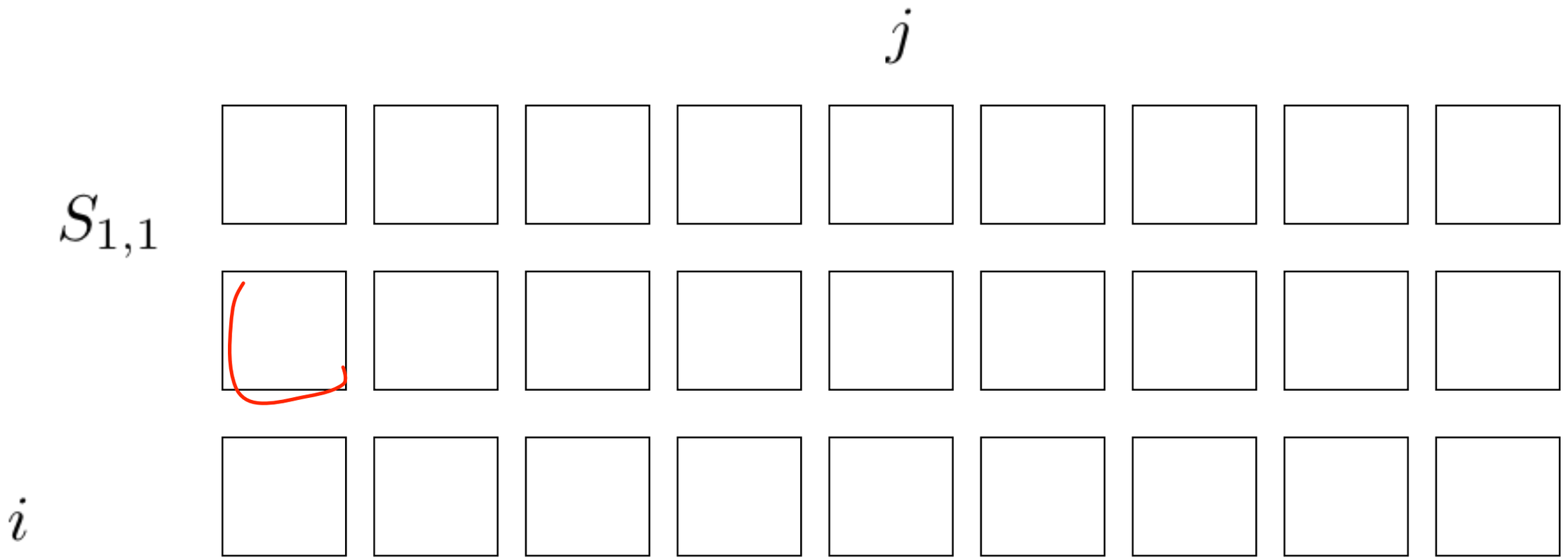
which word is fwoll?

$$\text{BEST}_n = \min \begin{cases} \text{BEST}_0 + S_{1,n}^2 \\ \text{BEST}_1 + S_{2,n}^2 \\ \text{BEST}_2 + S_{3,n}^2 \\ \cdots \\ \text{BEST}_{\ell-1} + S_{\ell,n}^2 \\ \\ \cdots \\ \\ \text{BEST}_{n-1} + S_{n,n}^2 \end{cases}$$

what about $S_{i,j}$



slack when line starts with w_i and ends w_j


j
$S_{1,1}$
i

typesetting algorithm

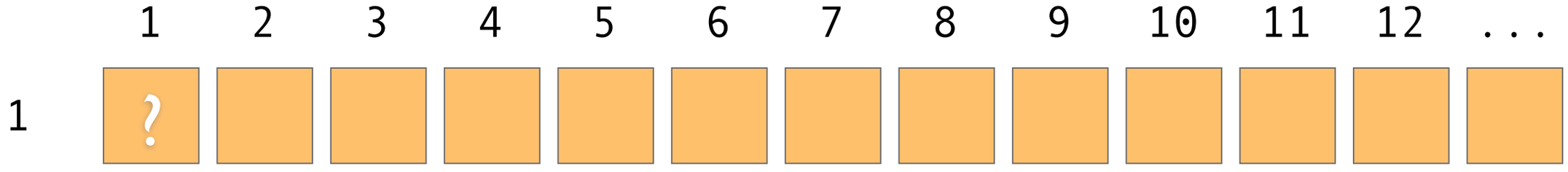
make table for $S_{i,j}$

example

```
It was the best of times, it was the worst of times; it was the age of
wisdom, it was the age of foolishness; it was the epoch of belief, it
was the epoch of incredulity; it was the season of


  2  3  3  4  2  6  2  3  3  5  2  6  2  3  3  3  2  7  2  3  3
  3  2 12  2  3  3  5  2  7  2  3  3  5  2 12  2  3  3  6  2
```

first step: make $S_{i,j}$



2 3 3 4 2 6 2 3 3 5 2 6 2 3 3 3 2 7 2 3 3
3 2 12 2 3 3 5 2 7 2 3 3 5 2 12 2 3 3 6 2

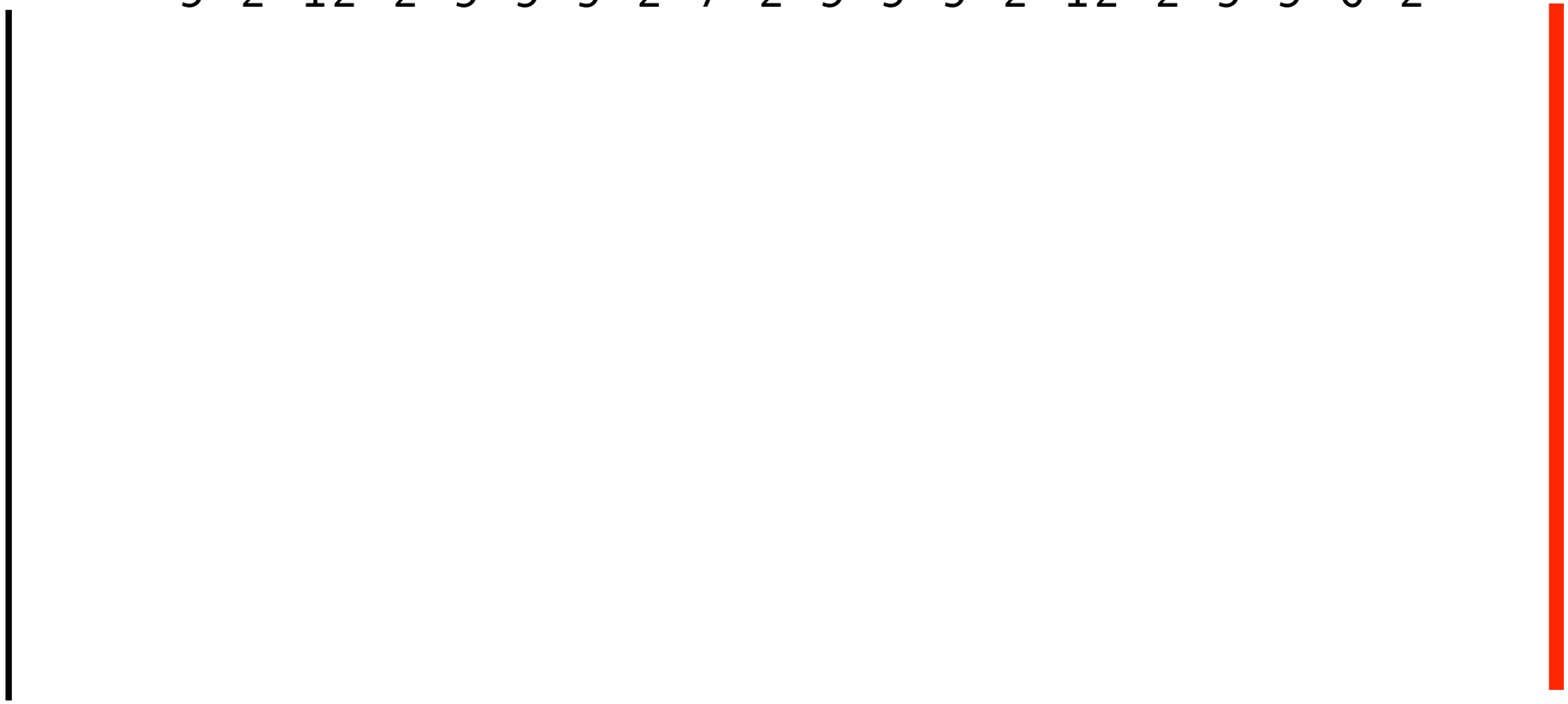
$M = 42$

first step: make $S_{i,j}$

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	40	36	32	27	24	17	14	10	6	0	99	99	99
2													

2 3 3 4 2 6 2 3 3 5 2 6 2 3 3 3 2 7 2 3 3
3 2 12 2 3 3 5 2 7 2 3 3 5 2 12 2 3 3 6 2

$M = 42$

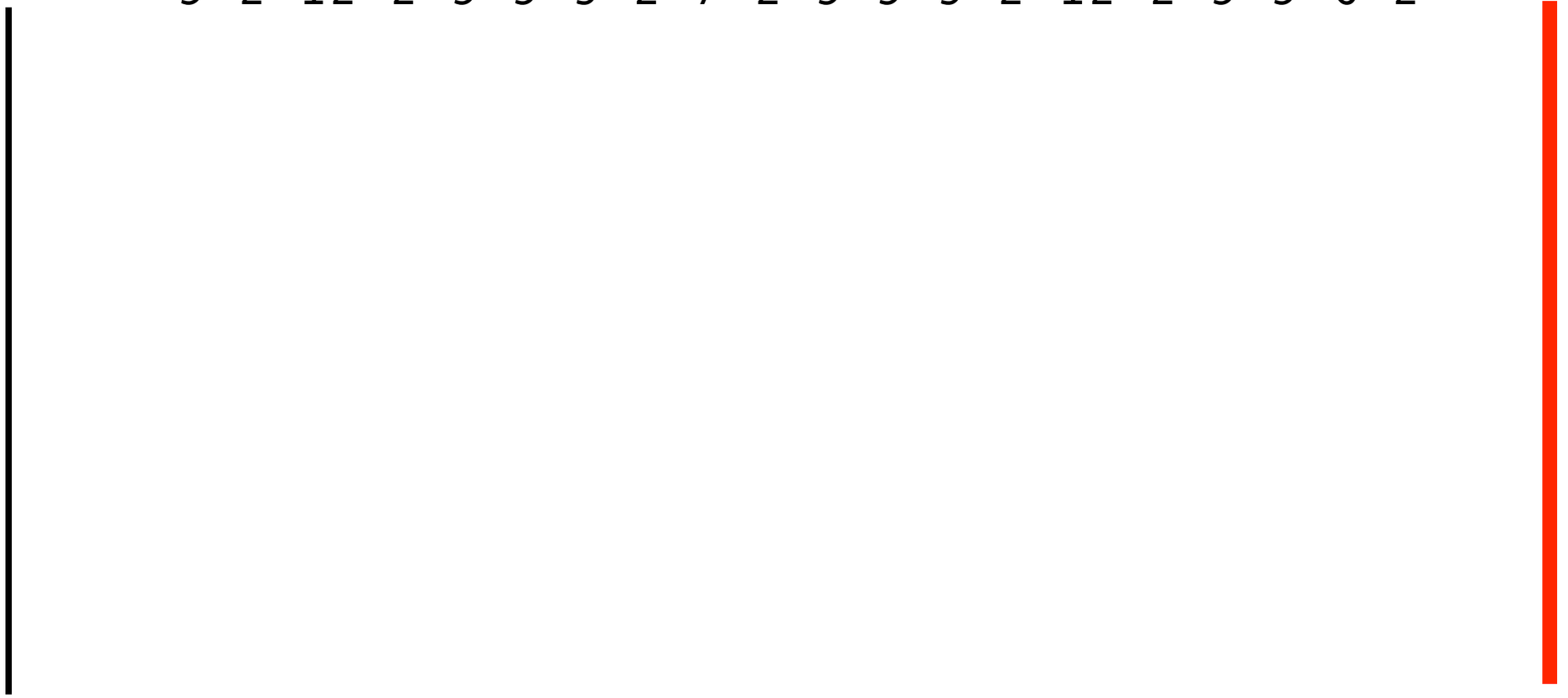


first step: make $S_{i,j}$

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	40	36	32	27	24	17	14	10	6	0	99	99	99
2	39	35	30	27	20	17	13	9	3	0	99	99	99

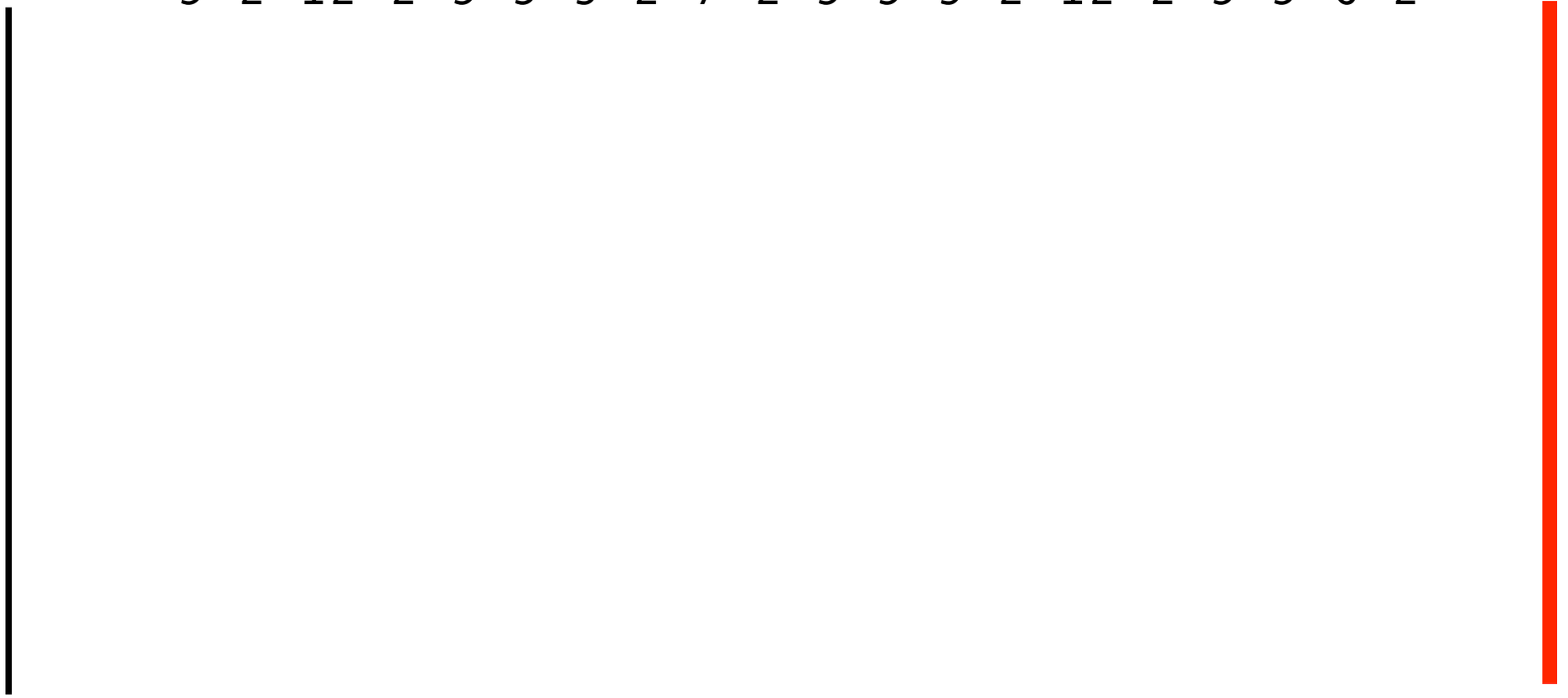
2 3 3 4 2 6 2 3 3 5 2 6 2 3 3 3 2 7 2 3 3

3 2 12 2 3 3 5 2 7 2 3 3 5 2 12 2 3 3 6 2

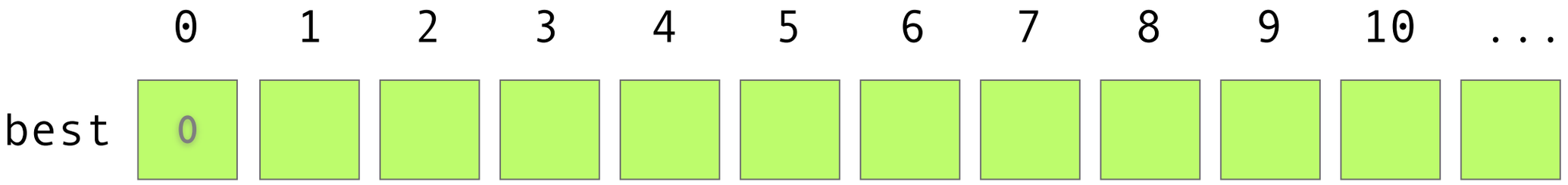


	1	2	3	4	5	6	7	8	9	10	11	12	13
1	40	36	32	27	24	17	14	10	6	0	99	99	99
2		39	35	30	27	20	17	13	9	3	0	99	99
3													

2 3 3 4 2 6 2 3 3 5 2 6 2 3 3 3 2 7 2 3 3
3 2 12 2 3 3 5 2 7 2 3 3 5 2 12 2 3 3 6 2



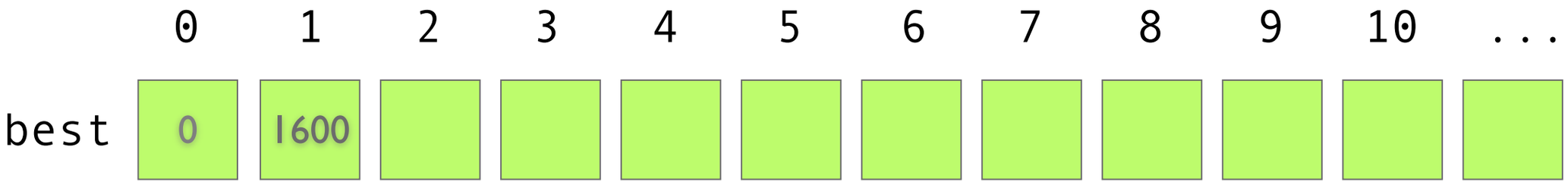
second step: compute best



$$\mathrm{BEST}_i = \min_{j=0}^{i-1} \left\{ \mathrm{BEST}_j + S^2_{j+1,i} \right\}$$

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	40	36	32	27	24	17	14	10	6	0	99	99	99
2		39	35	30	27	20	17	13	9	3	0	99	99

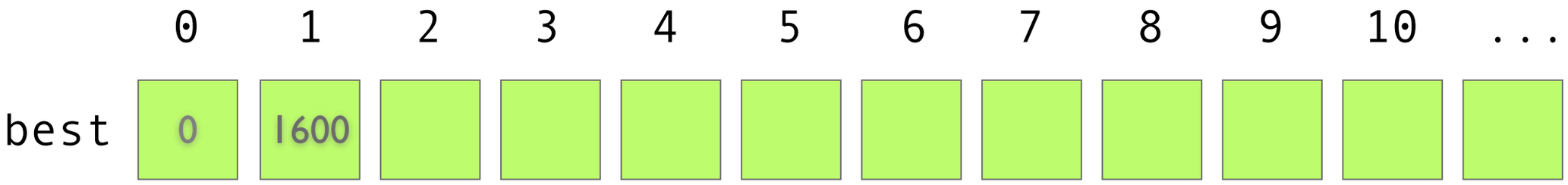
second step: compute best



$$\mathrm{BEST}_i = \min_{j=0}^{i-1} \left\{ \mathrm{BEST}_j + S^2_{j+1,i} \right\}$$

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	40	36	32	27	24	17	14	10	6	0	99	99	99
2		39	35	30	27	20	17	13	9	3	0	99	99

second step: compute best



$$\mathrm{BEST}_i = \min_{j=0}^{i-1} \left\{ \mathrm{BEST}_j + S^2_{j+1,i} \right\}$$

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	40	36	32	27	24	17	14	10	6	0	99	99	99
2		39	35	30	27	20	17	13	9	3	0	99	99