DP Lecture 4

Typesetting

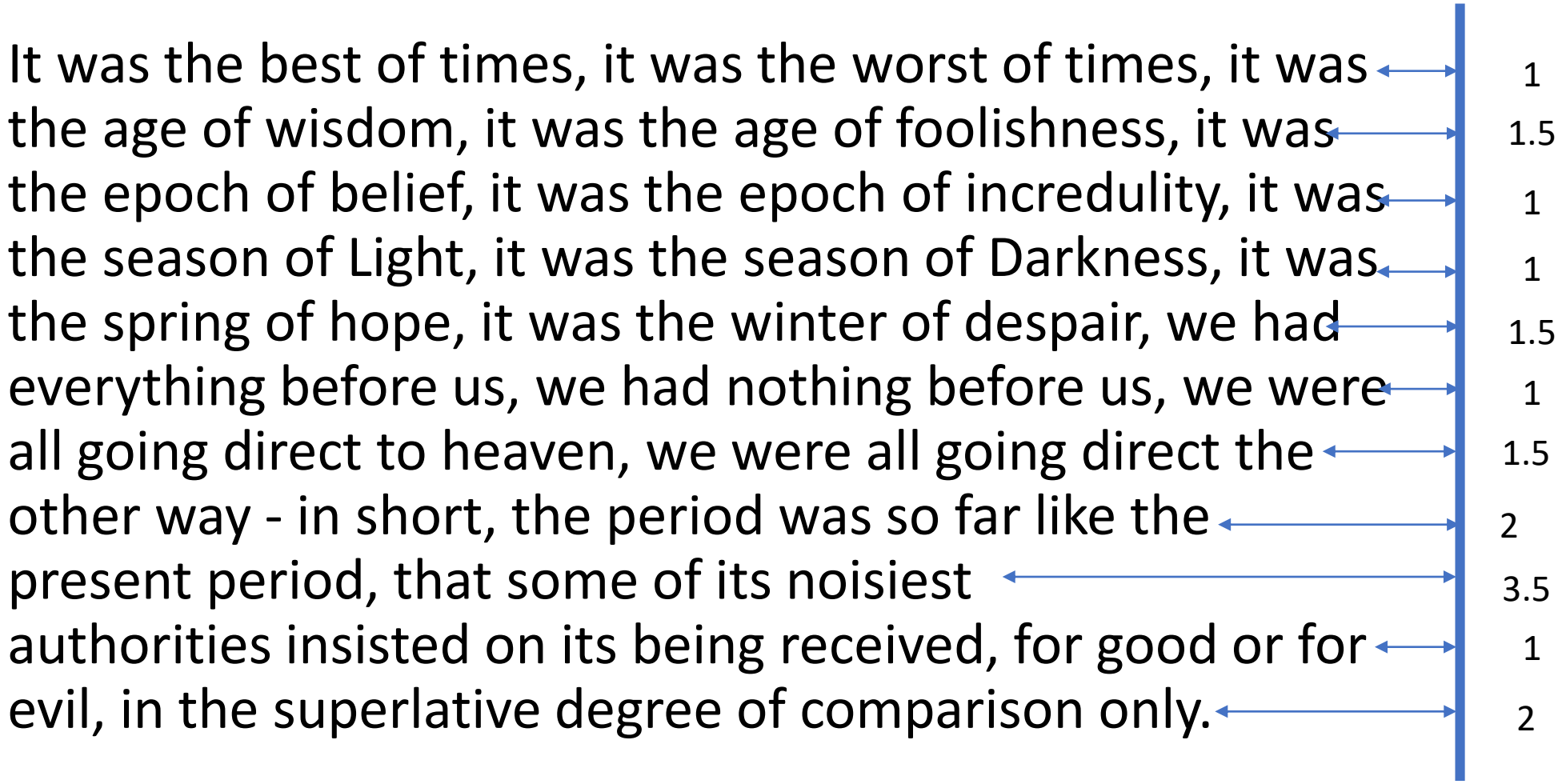
Typesetting

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.

Typesetting

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.

Typesetting



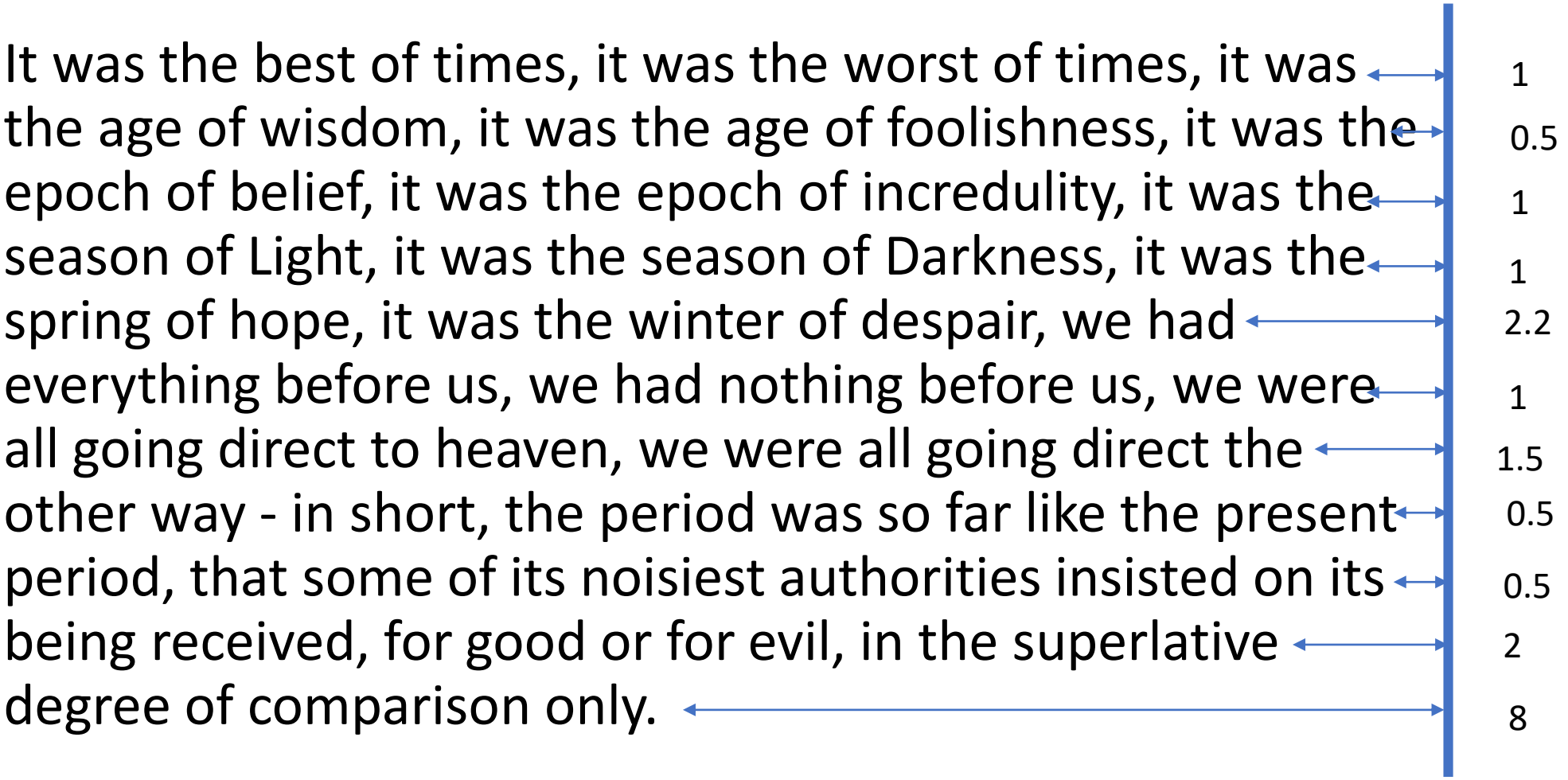
1	1
1.5	2.25
1	1
1	1
1.5	2.25
1	1
1.5	2.25
2	4
3.5	12.25
1	1
2	4
	32

Typesetting

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.

Typesetting

79.84



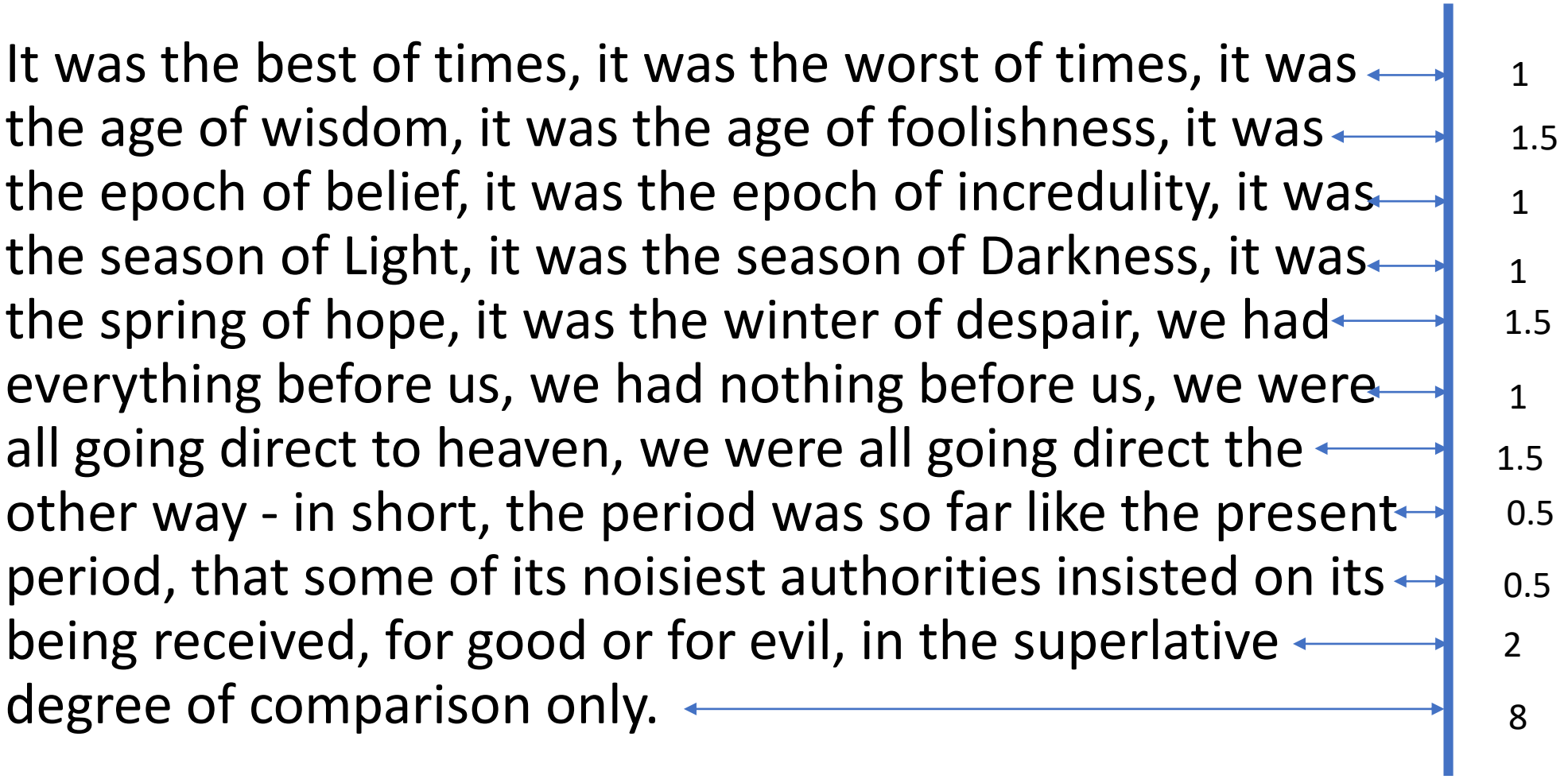
1	1
0.5	0.25
1	1
1	1
2.2	4.84
1	1
1.5	2.25
0.5	0.25
0.5	0.25
2	4
8	64
	79.84

Typesetting

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.

Typesetting

79.25



Typesetting

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.

Typesetting Problem definition

Input: $W=\{w_1, w_2, w_3, \dots, w_n\}$

Output: $L= (w_1,\dots,w_{l1}), (w_{l1+1}, \dots, w_{l2}), \dots, (w_{lx+1},\dots,w_n)$

$$C_i = \left(\sum_{l_i+1}^{l_{i+1}} |wj|\right) + l_{i+1} - l_i - 1$$

Such that, $ci \leq M, \ \forall_l$

$$\text{Min } \sum (M-c_i)^2$$

Typesetting Problem definition

Input: $W=\{w_1, w_2, w_3, \dots, w_n\}$

Output: $L= (w_1,\dots,w_{l1}), (w_{l1+1}, \dots, w_{l2}), \dots, (w_{lx+1},\dots,w_n)$

$$C_i = \left(\sum_{l_i+1}^{l_{i+1}} |wj|\right) + l_{i+1} - l_i - 1$$

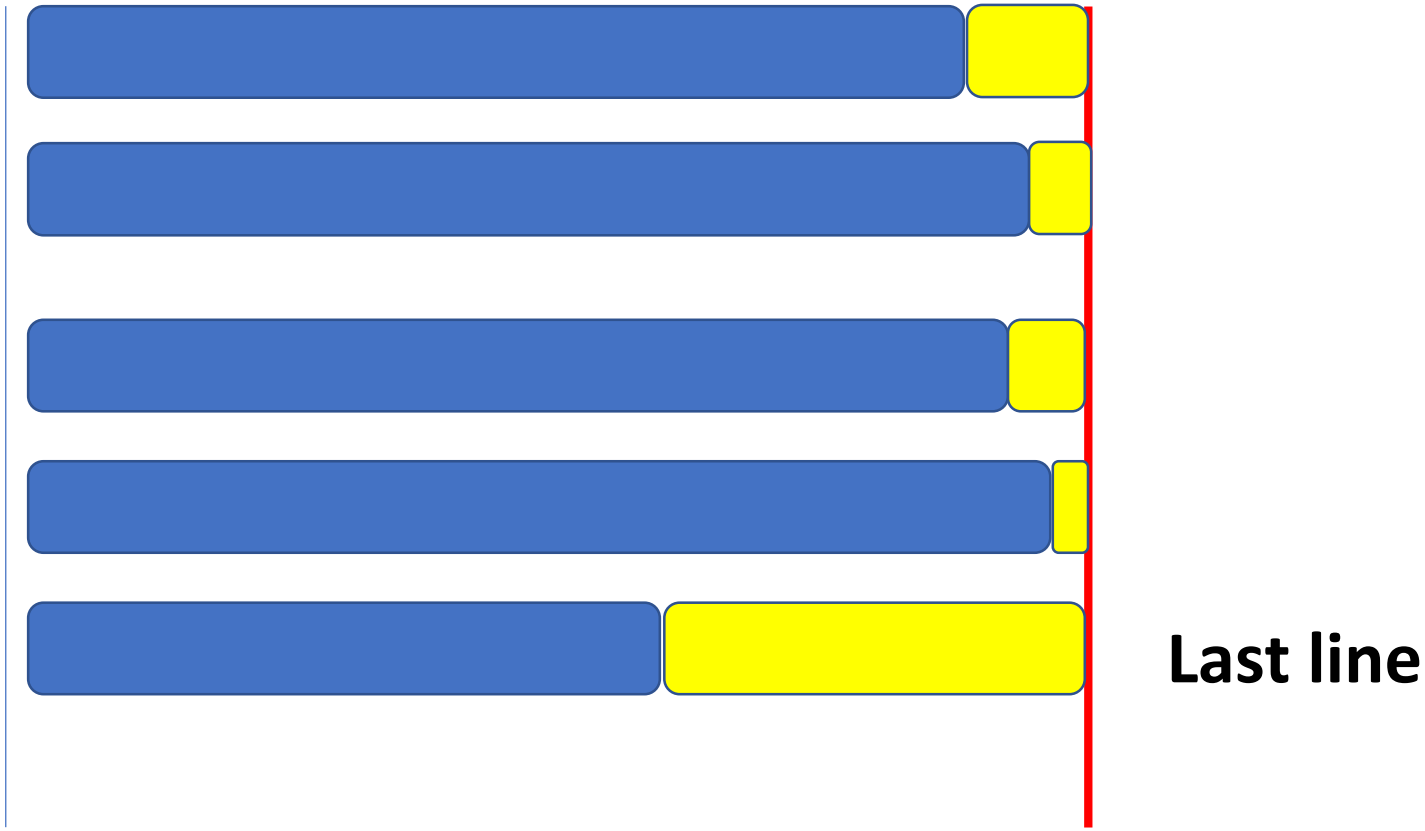
Such that, $ci \leq M, \ \forall_l$

$$\text{Min } \sum (M-c_i)^2$$

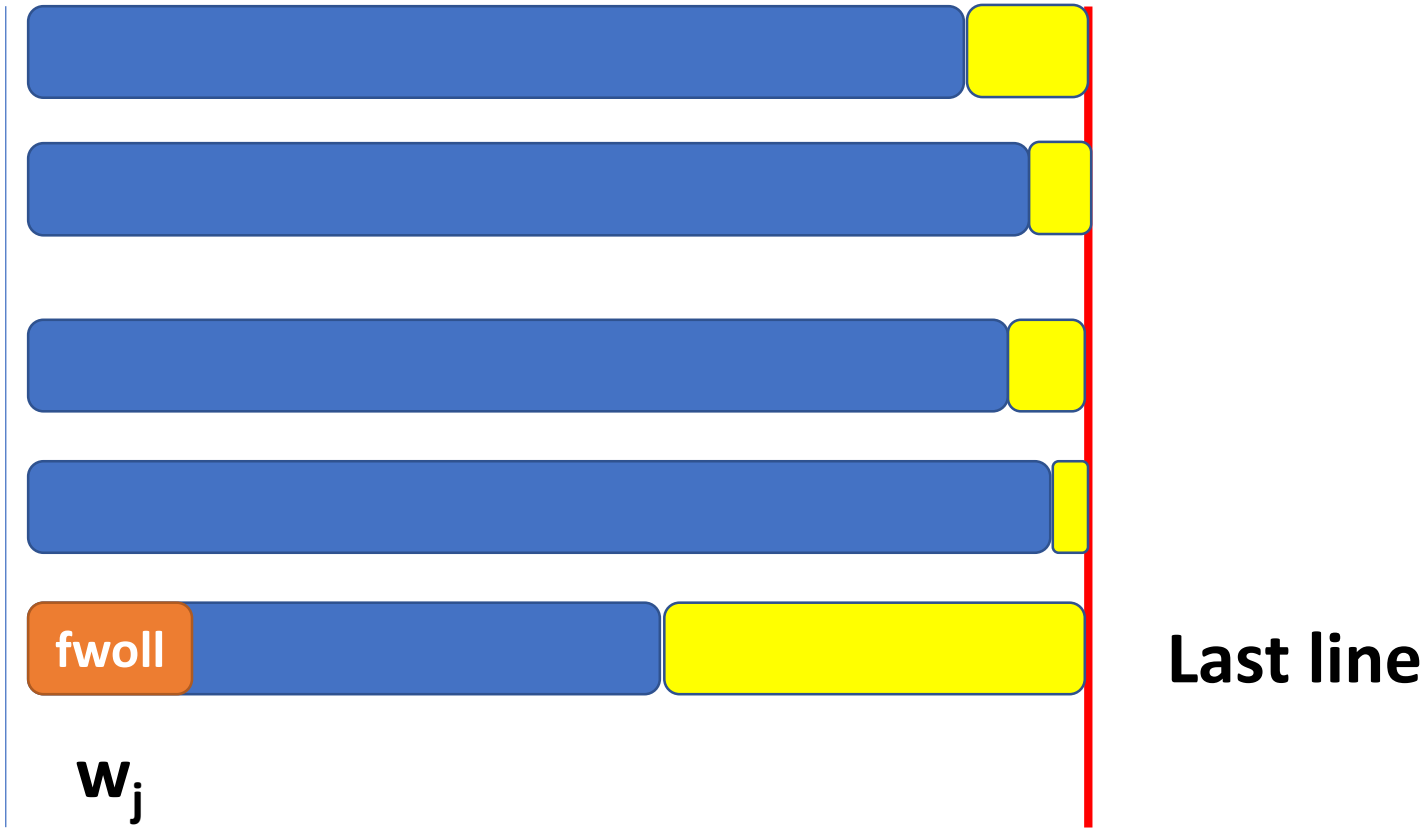
How to solve DP problem

- Define the right memory variable
  - $Best_n$: minimum penalty for typesetting the first n words of the problem.

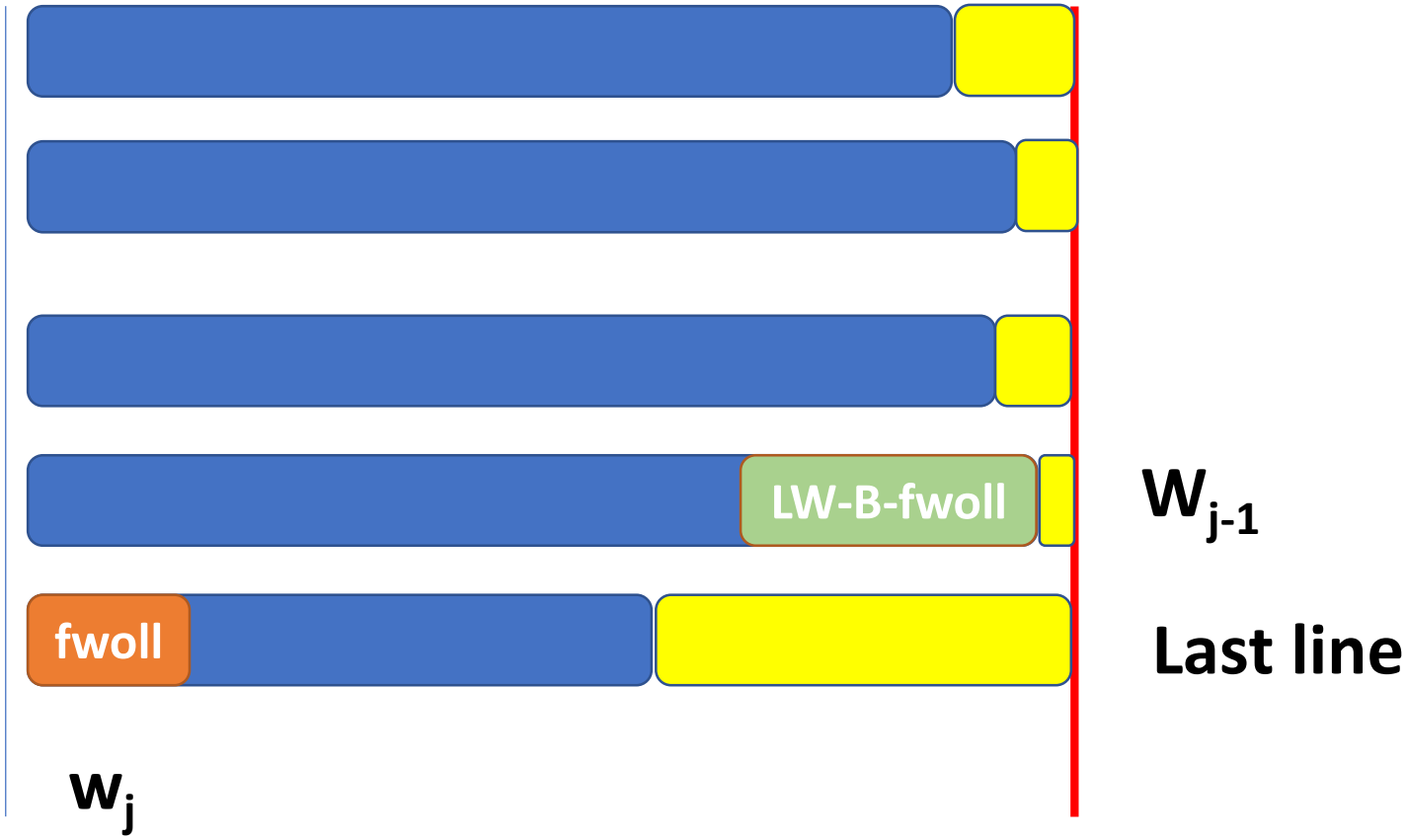
How to solve DP problem (Last Step)



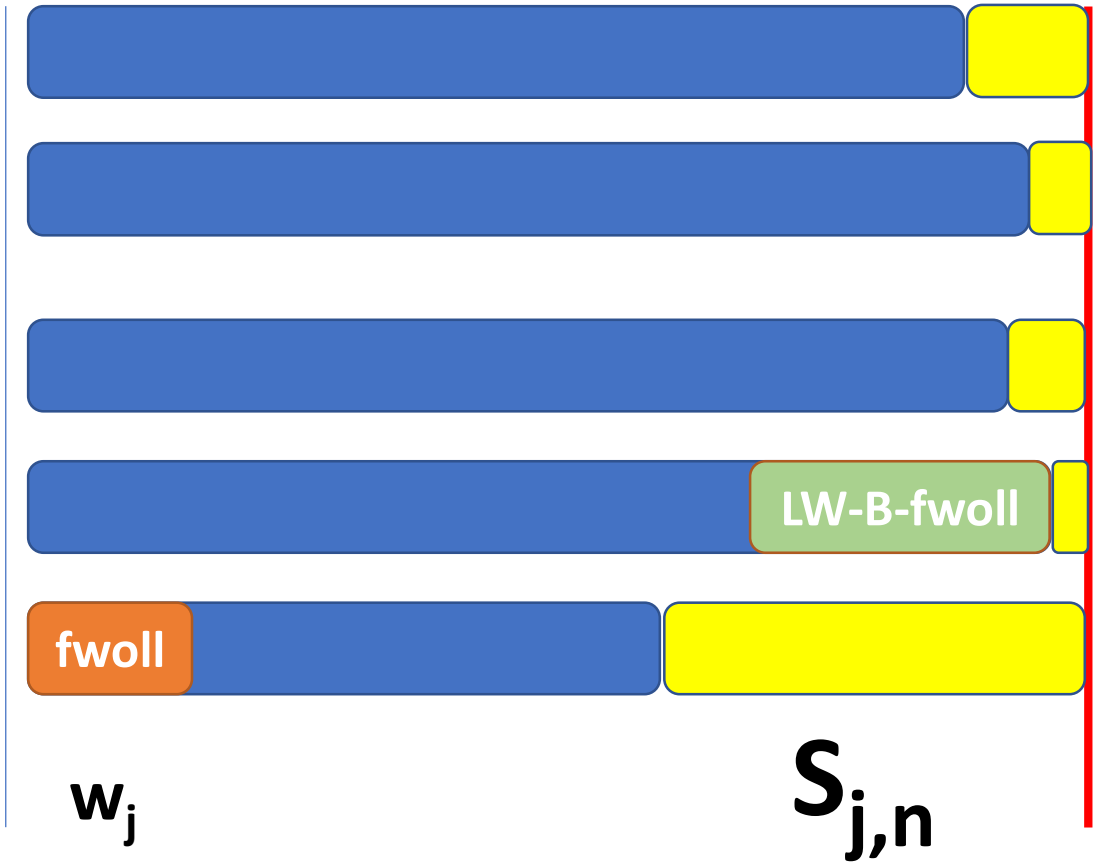
How to solve DP problem (Last Step)



How to solve DP problem (Last Step)



How to solve DP problem (Last Step)

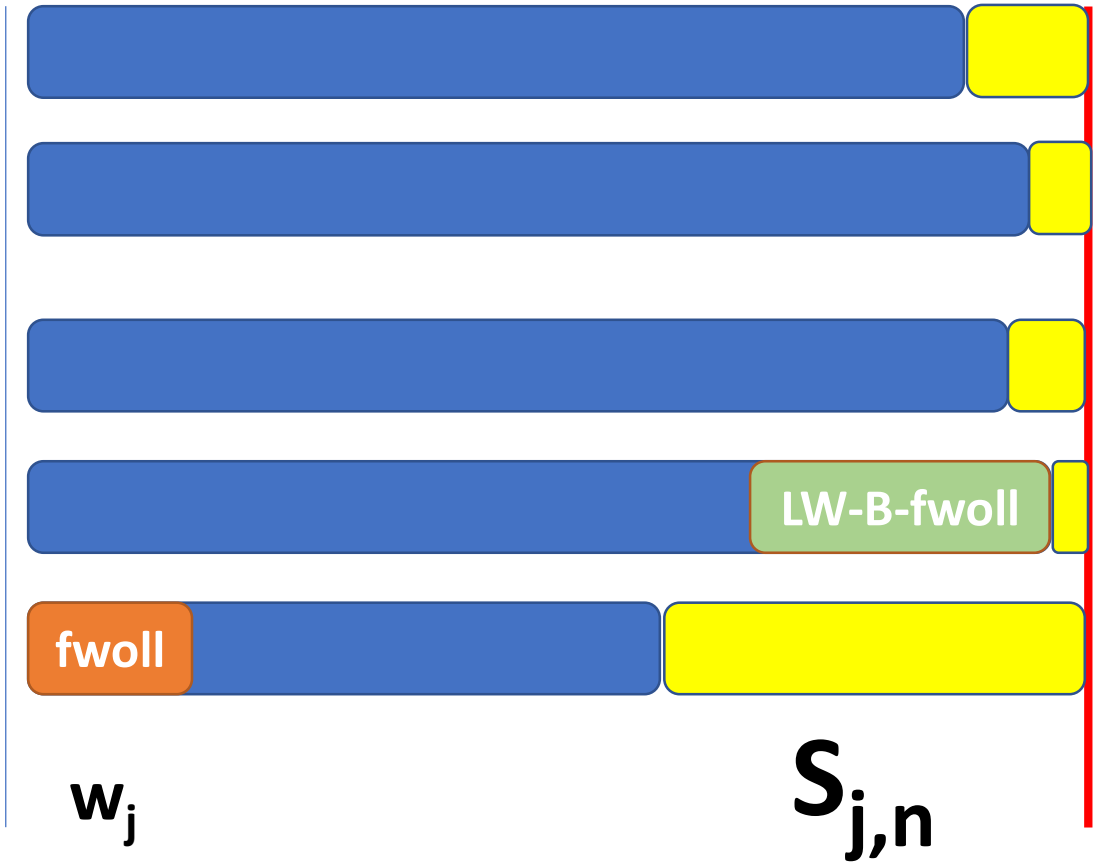


Slack induced by typesetting word **j** as **fwoll** for first **n** words

Some word has to be the

first – word – of – last – line (fwoll)

How to solve DP problem (Last Step)

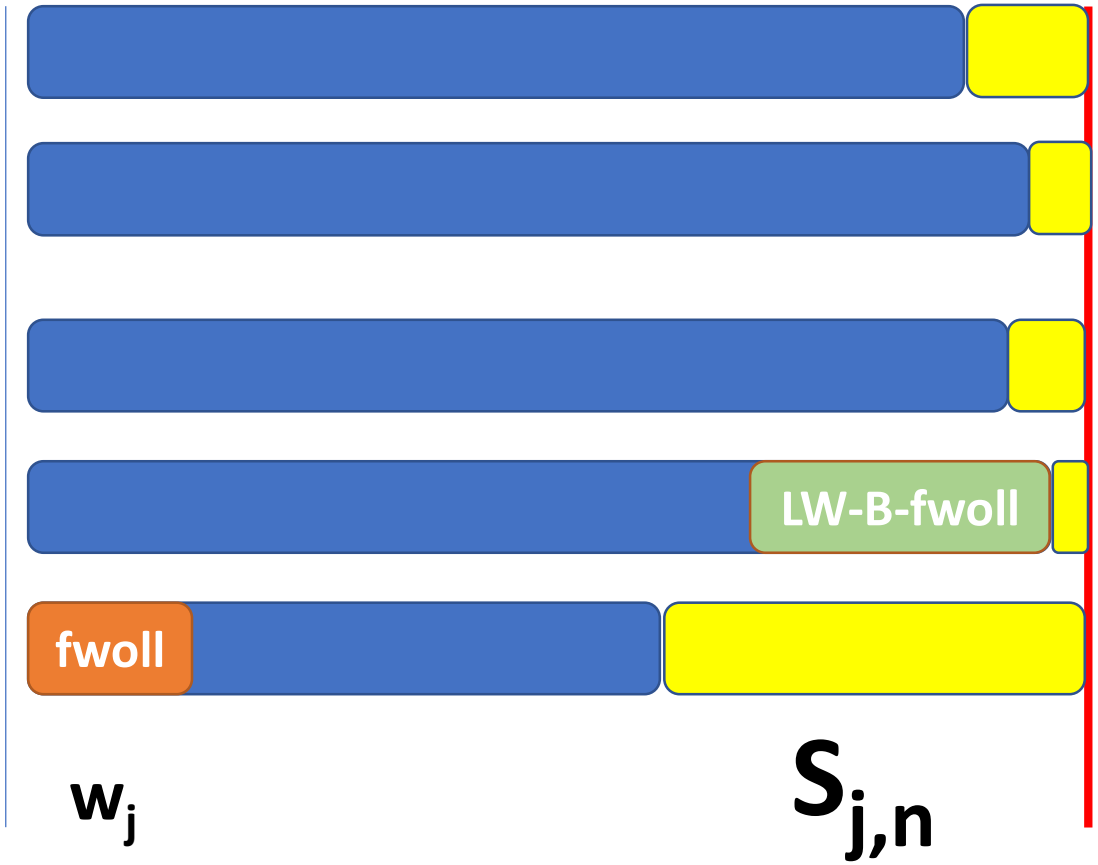


w_{j-1} *Best* $_{fwoll-1}$

Last line

Slack induced by typesetting word **j** as **fwoll** for first **n** words

How to solve DP problem (Last Step)



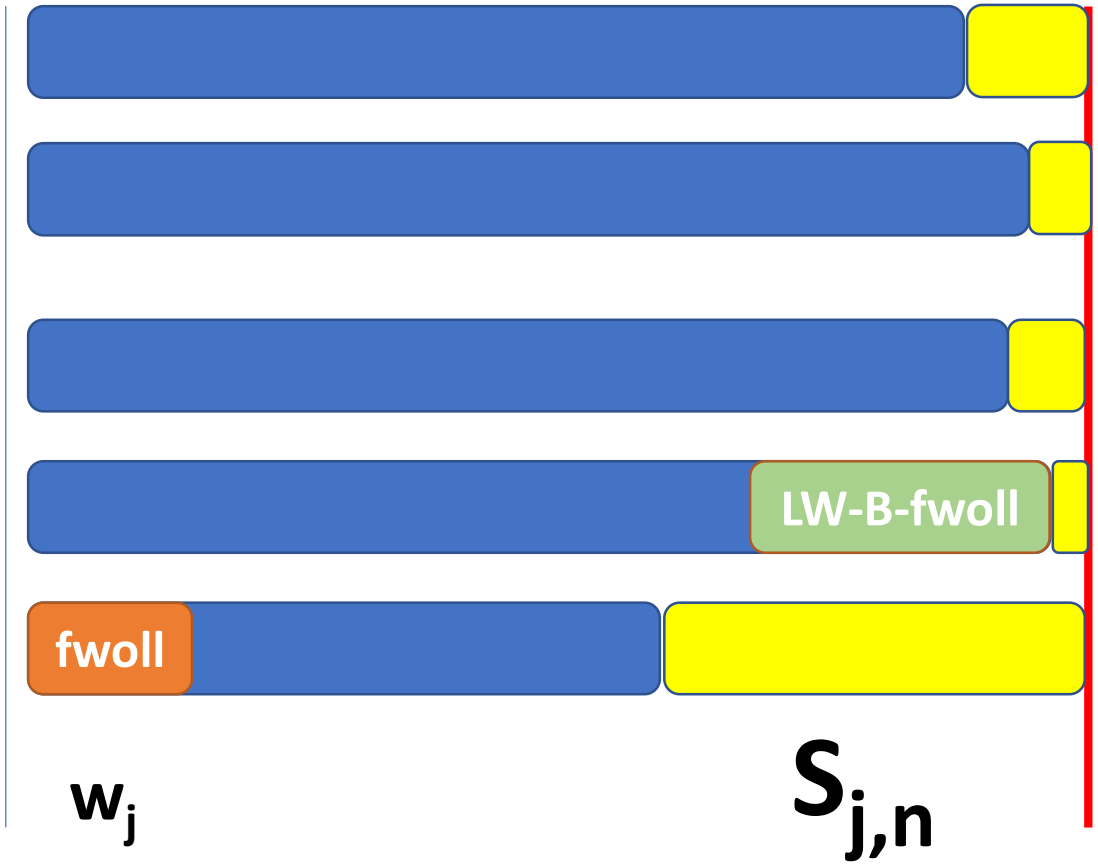
$$Best_n = Best_{fwoll-1} + (S_{fwoll,n})^2$$

w_{j-1} *Best* $_{fwoll-1}$

Last line

Slack induced by typesetting word **j** as **fwoll** for first **n** words

How to solve DP problem (Last Step)



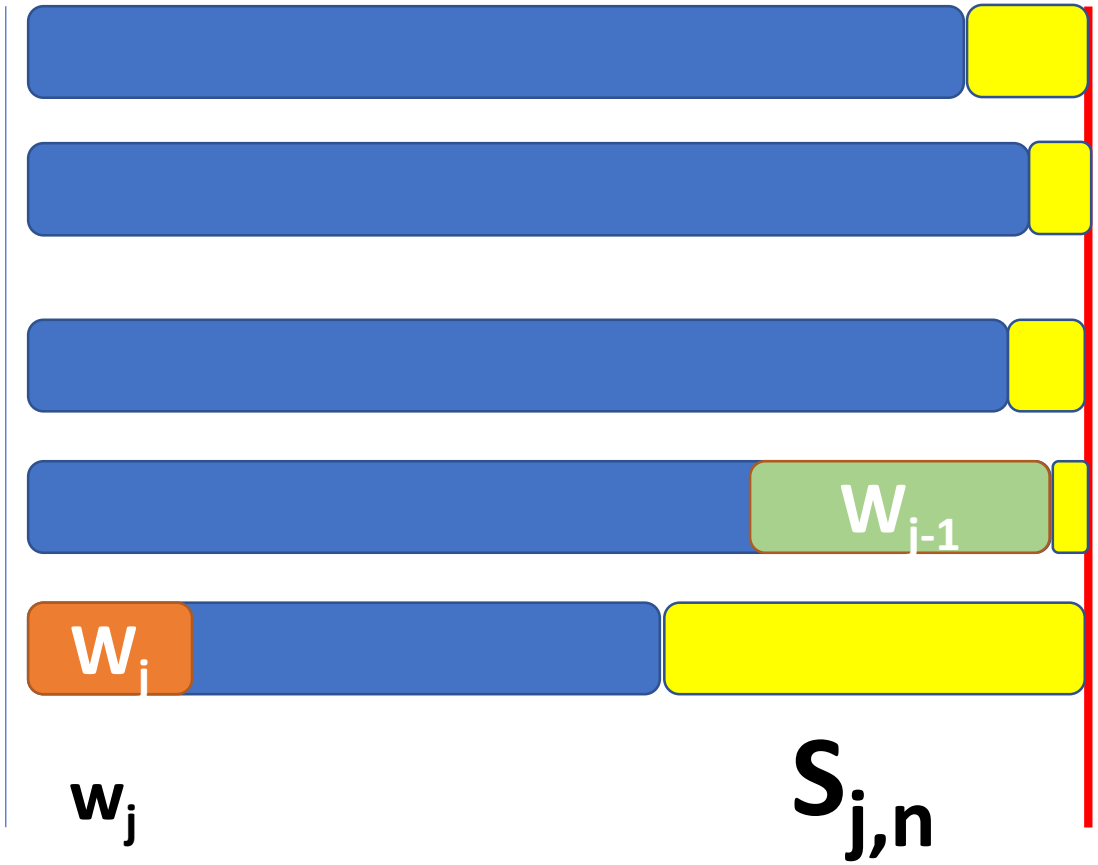
$$Best_n = Best_{fwoll-1} + (S_{fwoll,n})^2$$

w_{j-1} *Best* $_{fwoll-1}$

Last line

Slack induced by typesetting word **j** as **fwoll** for first **n** words

How to solve DP problem (Last Step)



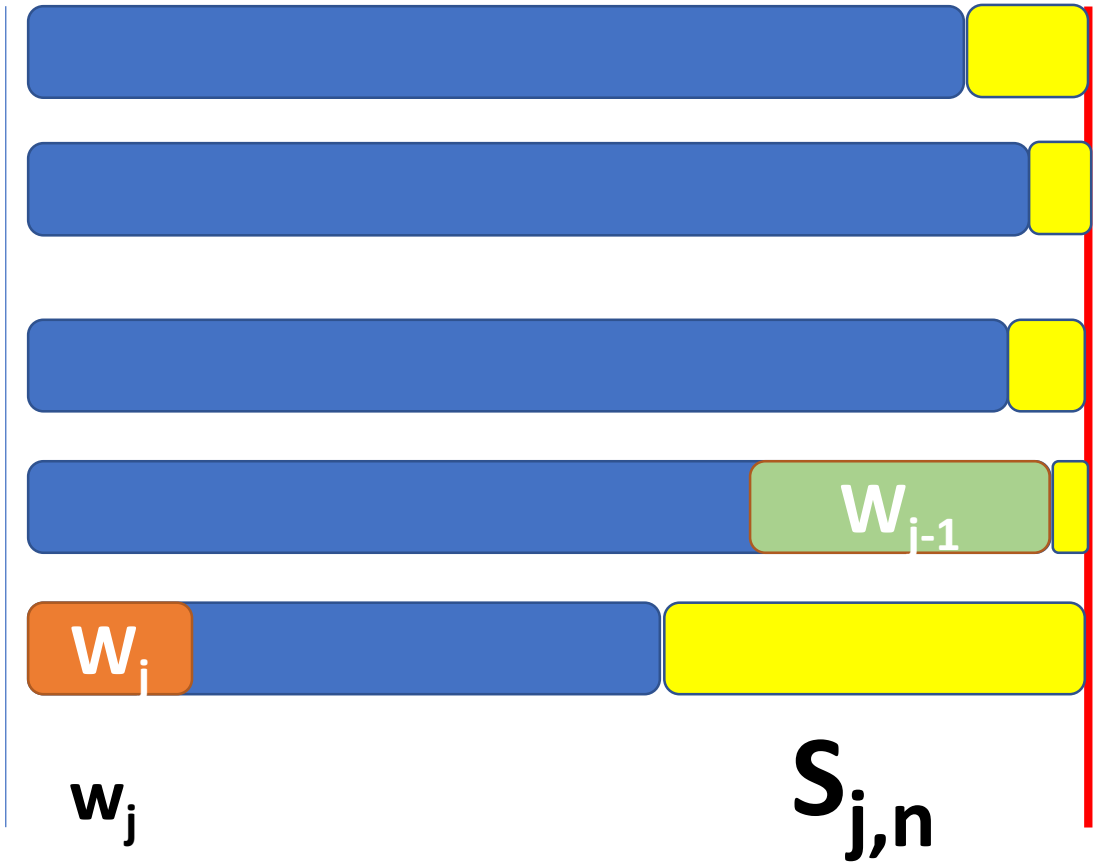
$$\mathbf{Best_n = Best_{j-1} + (S_{j,n})^2}$$

$\mathbf{W_{j-1}}$ *Best* $_{j-1}$

Last line

Slack induced by typesetting word **j** as **fwoll** for first **n** words

How to solve DP problem (Last Step)



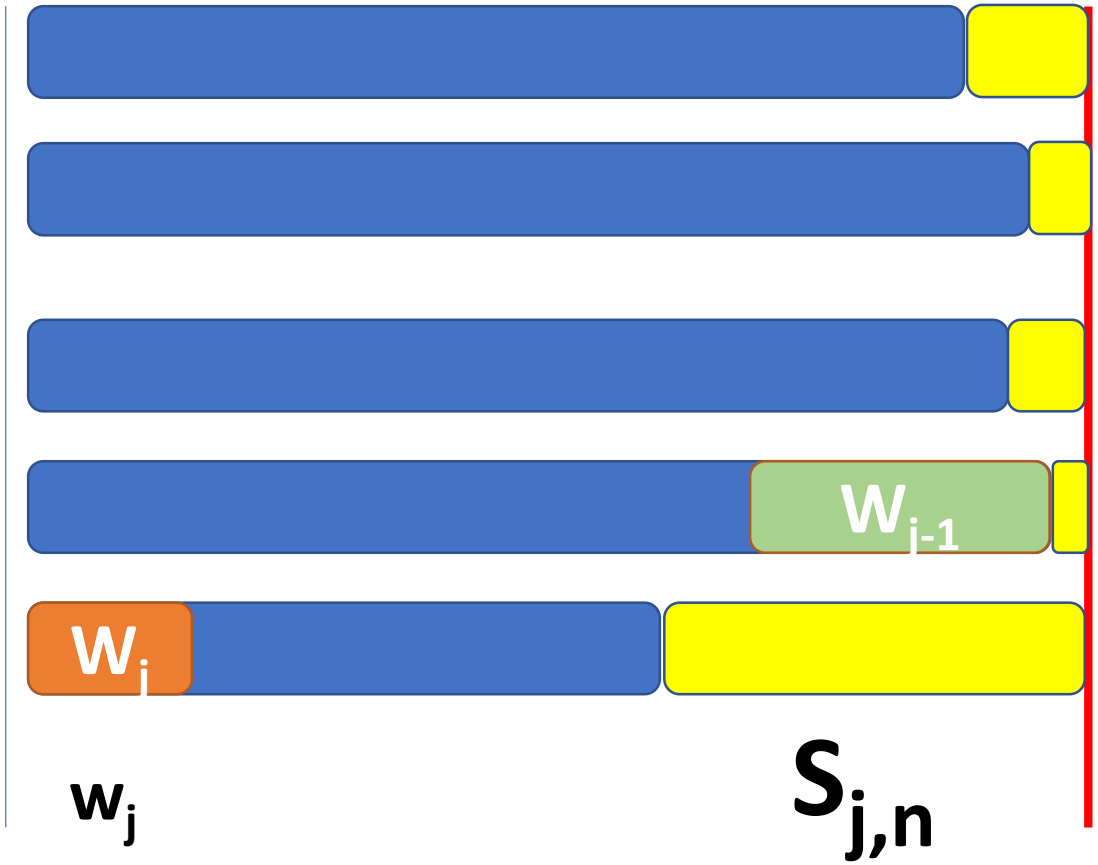
$$Best_n = \min \begin{cases} Best_0 + (S_{1,n})^2 \\ Best_1 + (S_{2,n})^2 \\ Best_2 + (S_{3,n})^2 \\ \cdots\cdots \\ Best_{j-1} + (S_{j,n})^2 \\ \cdots\cdots \\ Best_{n-1} + (S_{n,n})^2 \end{cases}$$

Suppose we have a table $S_{i,j}$:

Stores slack for typesetting word i as fwoll among first j words

For i=1 to n

best[i]=min{best[j]+(s[j+1][i])^2}



$$Best_n = \min \begin{cases} Best_0 + (S_{1,n})^2 \\ Best_1 + (S_{2,n})^2 \\ Best_2 + (S_{3,n})^2 \\ \cdots\cdots \\ Best_{j-1} + (S_{j,n})^2 \\ \cdots\cdots \\ Best_{n-1} + (S_{n,n})^2 \end{cases}$$

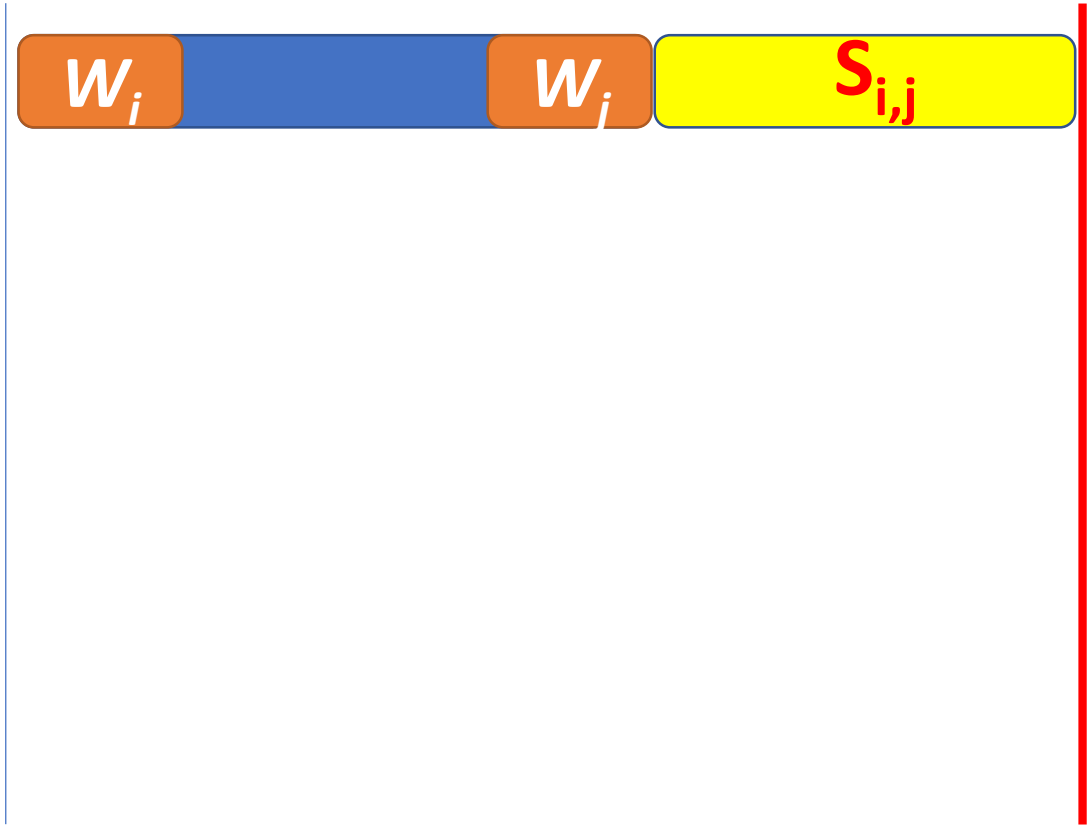
compute best_0,...,best_n

```
int best[] = new int[n+1];
int choice[] = new int[n+1];
best[0] = 0;
for(int i=1;i<=n;i++) {
    int min = infty;
    int ch = 0;
    for(int j=0;j<I;j++){
        int t = best[j] + S[j+1][i]*S[j+1][i];
        If (t<min){
            min=t;
            ch=j;
        }
    }
    best[i]=min;
    choice[i]=ch;
}
```

For i=1 to n

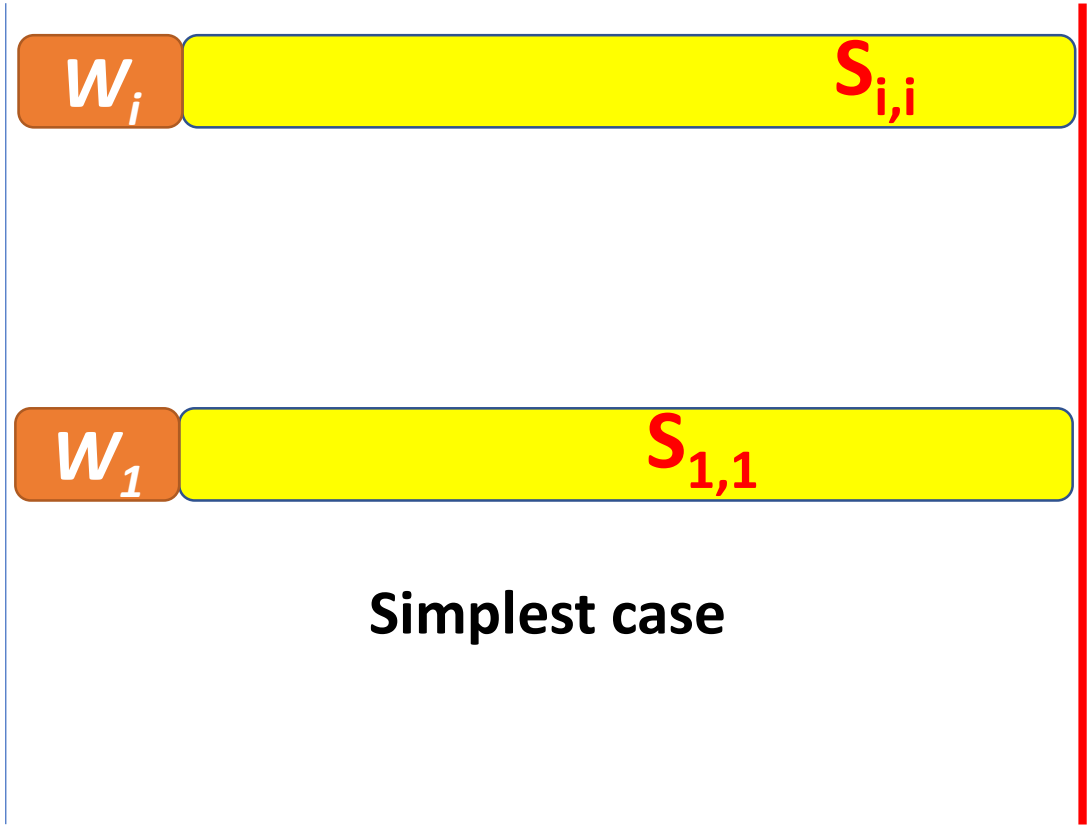
$$best[i]=min\{best[j]+(s[j+1][i])^2\}$$

How to compute $S_{i,j}$



Slack when line starts with w_i and end with w_j

How to compute $S_{i,j}$

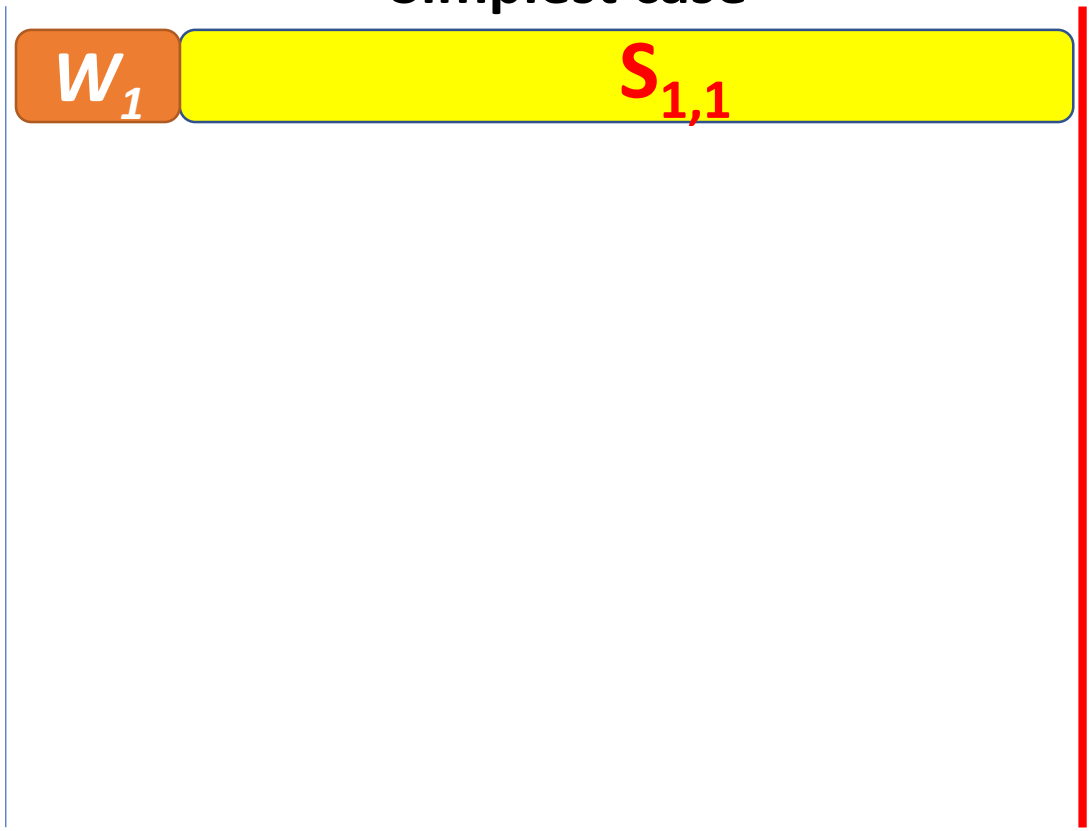


Slack when line starts with w_i and end with w_i

$S_{1,1} = M - W_1$

How to compute $S_{i,j}$

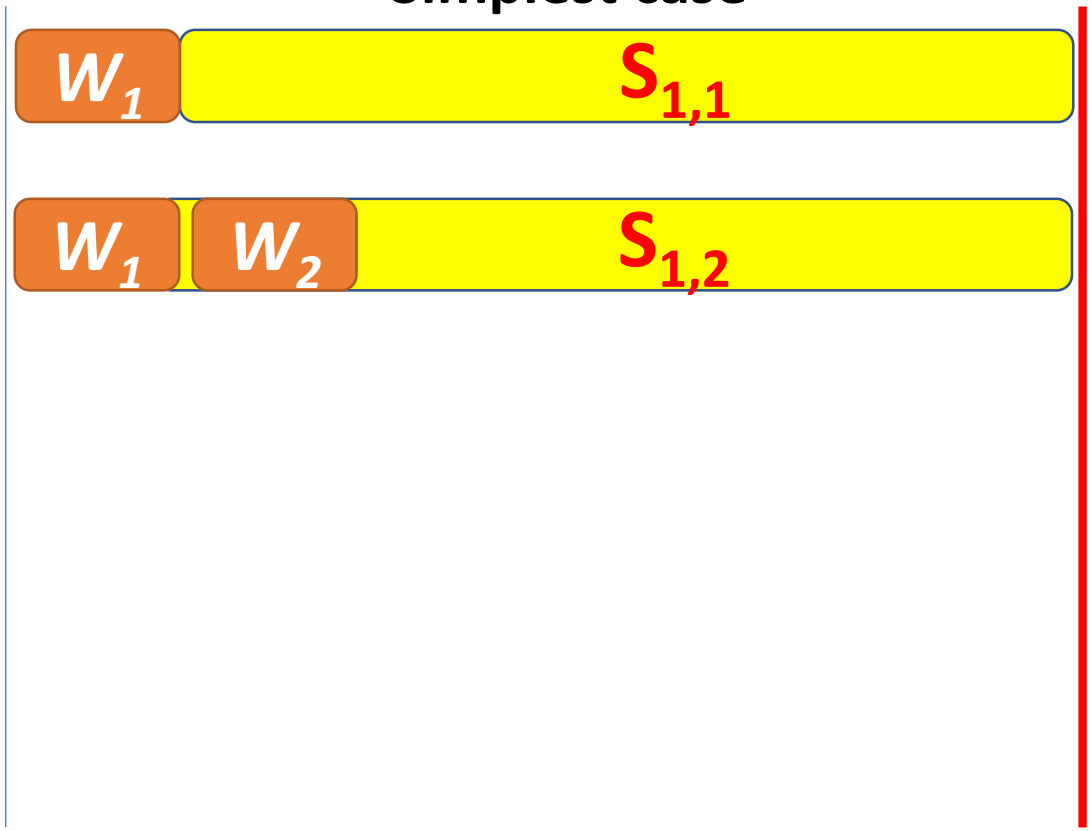
Simplest case



$S_{1,1} = M - W_1$

How to compute $S_{i,j}$

Simplest case

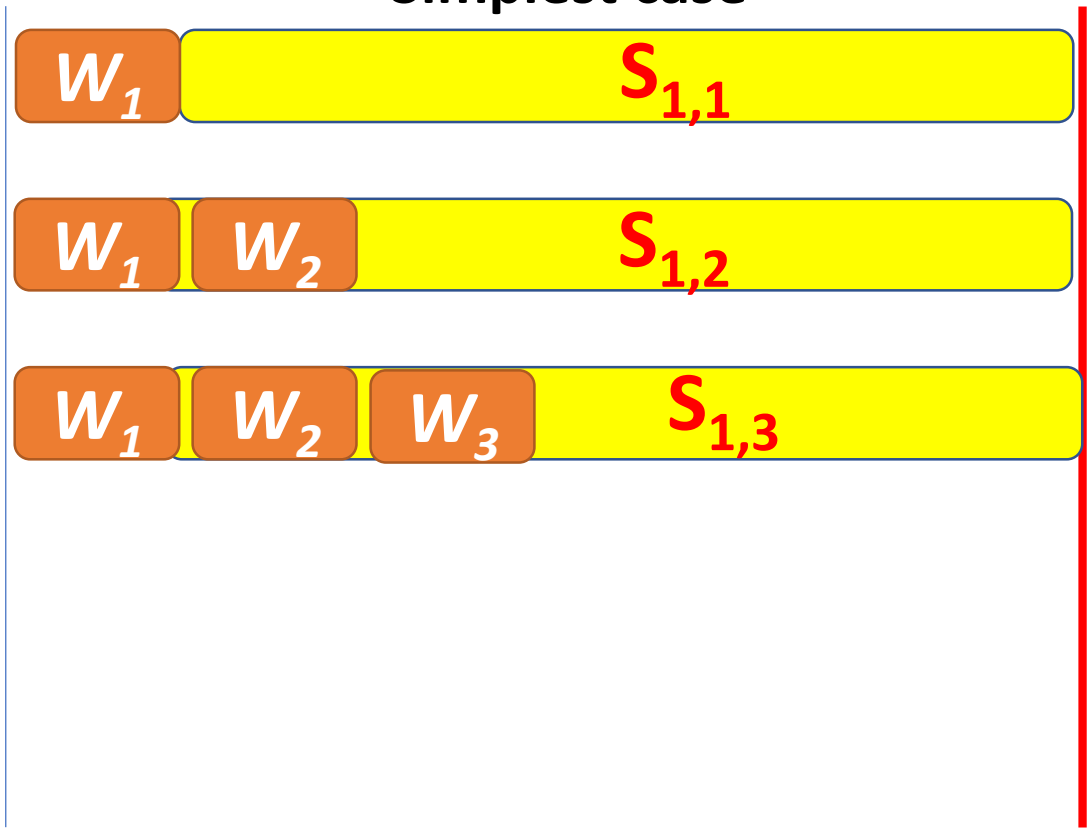


$S_{1,1} = M - W_1$

$S_{1,2} = S_{1,1} - W_2 - 1$

How to compute $S_{i,j}$

Simplest case

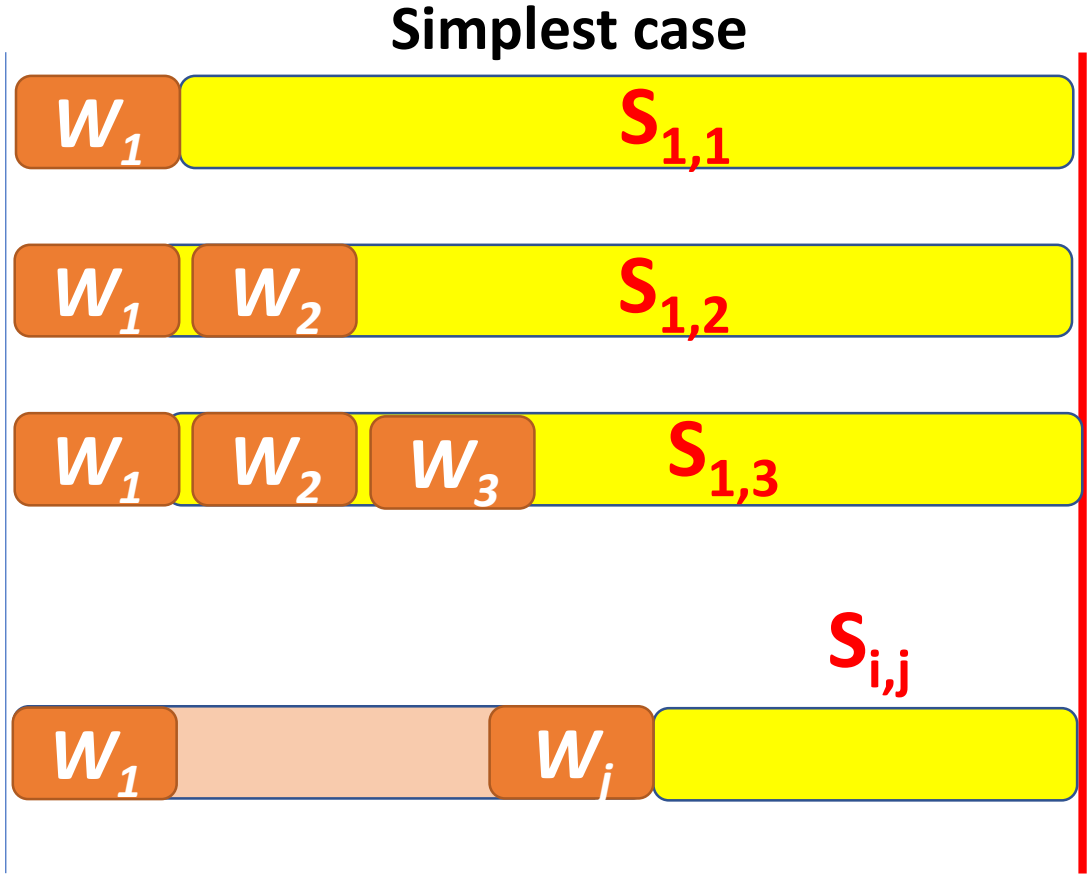


$S_{1,1} = M - W_1$

$S_{1,2} = S_{1,1} - W_2 - 1$

$S_{1,3} = S_{1,2} - W_3 - 1$

How to compute $S_{i,j}$



$S_{1,1} = M - W_1$

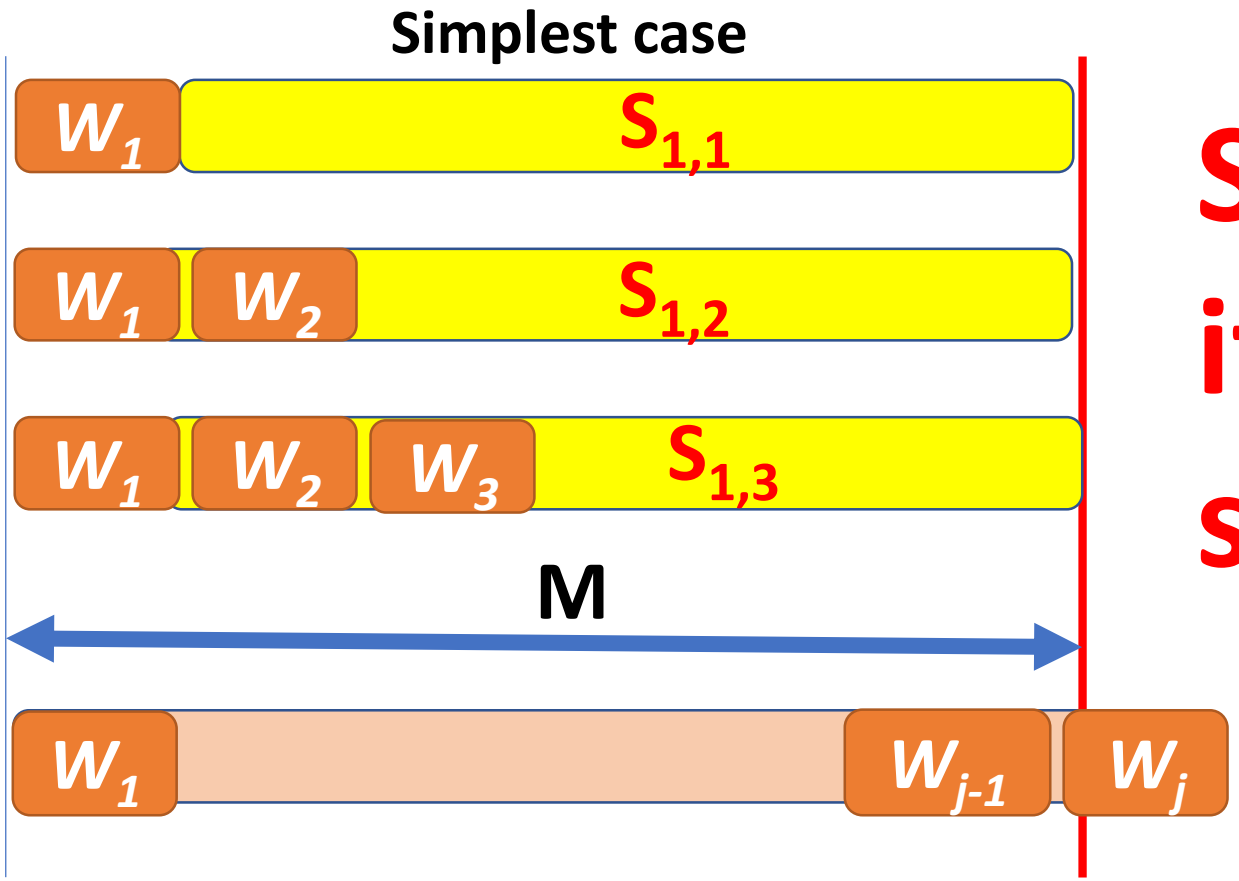
$S_{1,2} = S_{1,1} - W_2 - 1$

$S_{1,3} = S_{1,2} - W_3 - 1$

$S_{i,j} = S_{i,j-1} - W_j - 1$

$S_{i,j}$: Slack when line starts with w_i and end with w_j

How to compute $S_{i,j}$



$$S_{i,j} = S_{i,j-1} - W_j - 1$$

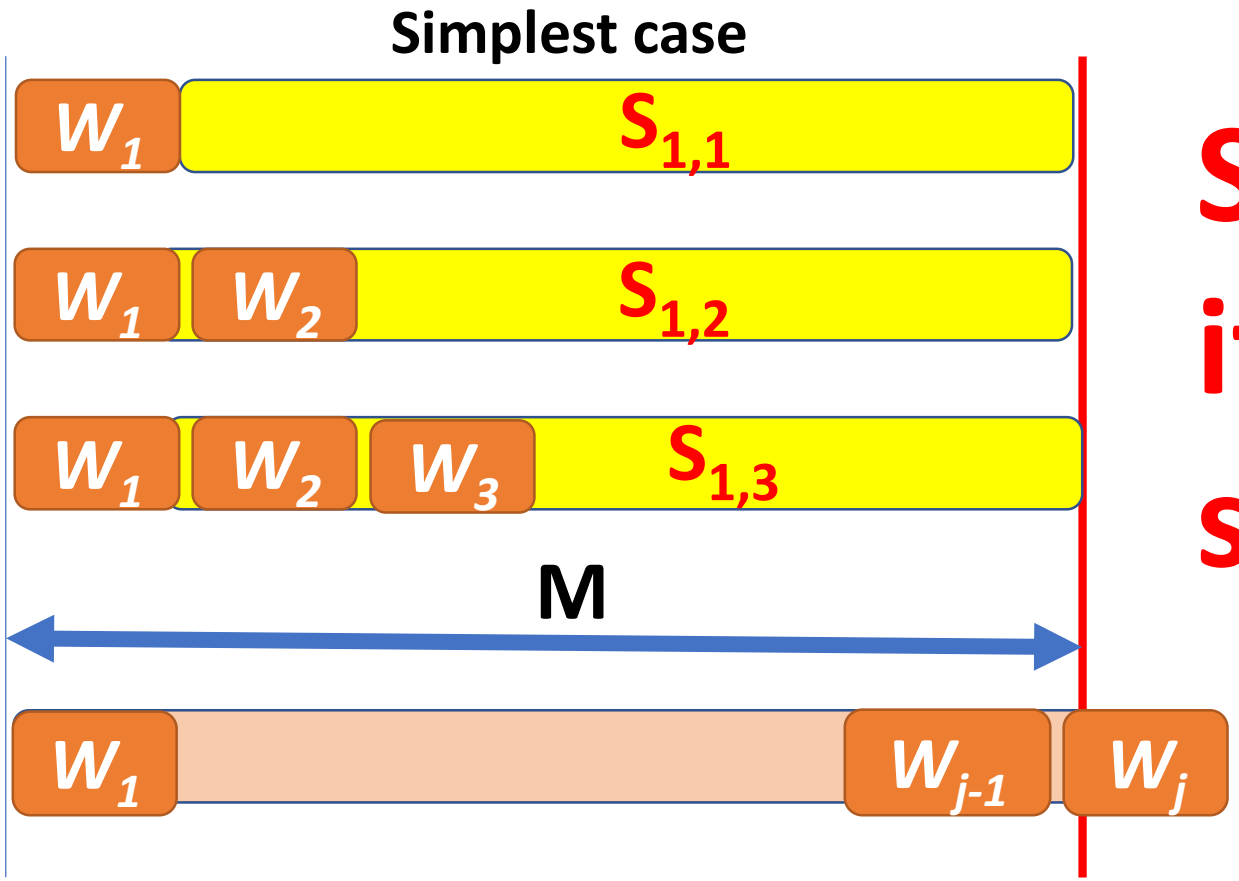
if $S_{i,j} < 0$,

set $S_{i,j} = 0$

$S_{i,j}$ Negative!!!

$S_{i,j}$: Slack when line starts with w_i and end with w_j

How to compute $S_{i,j}$



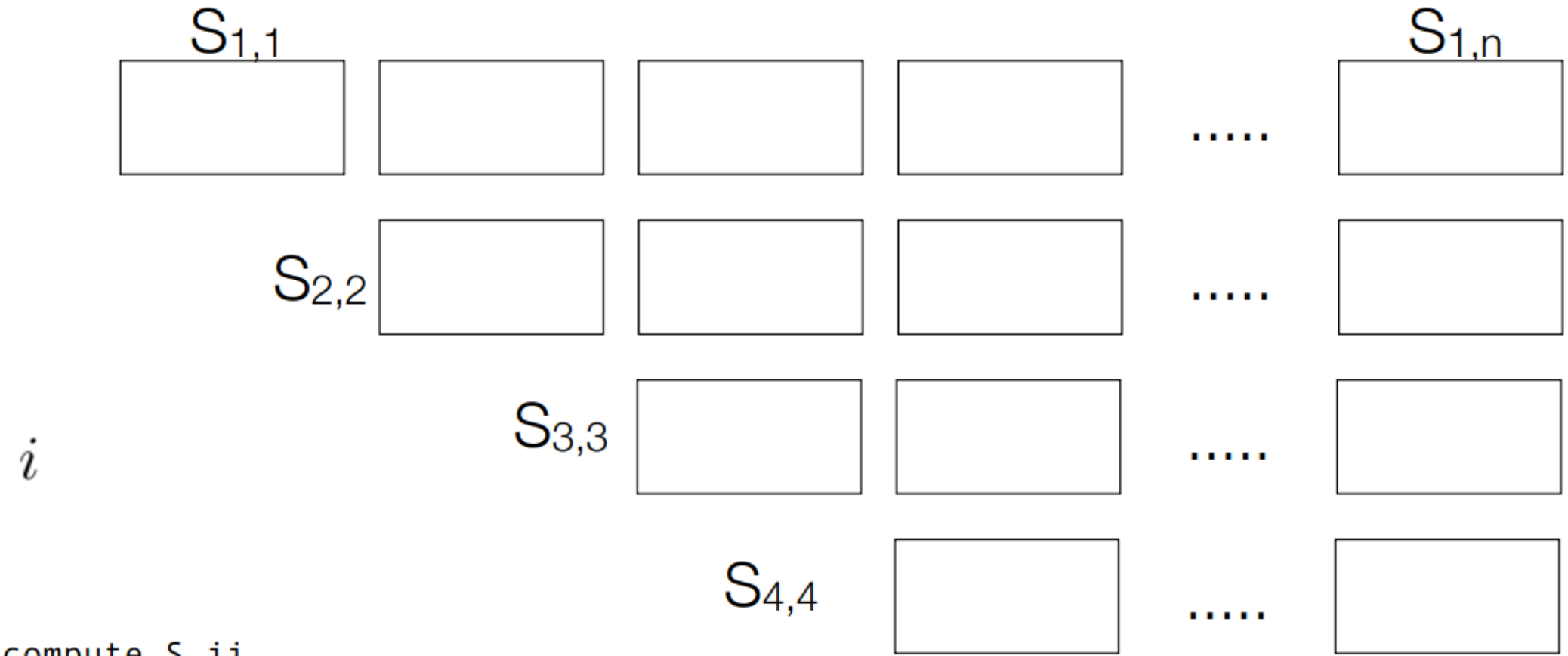
$$S_{i,j} = S_{i,j-1} - W_j - 1$$

if $S_{i,j} < 0$,

set $S_{i,j} = 0$

$S_{i,j}$ Negative!!!

$S_{i,j}$: Slack when line starts with w_i and end with w_j




```
// compute S_ij
int S[][] = new int[n+1][n+1];
for(int i=1;i<=n;i++) {
    S[i][i] = M - lens[i];
    for(int j=i+1; j<=n; j++) {
        S[i][j] = S[i][j-1] - lens[j] - 1;
        if (S[i][j]<0) {
            while(j<=n) { S[i][j++] = infty; }
        }
    }
}
```

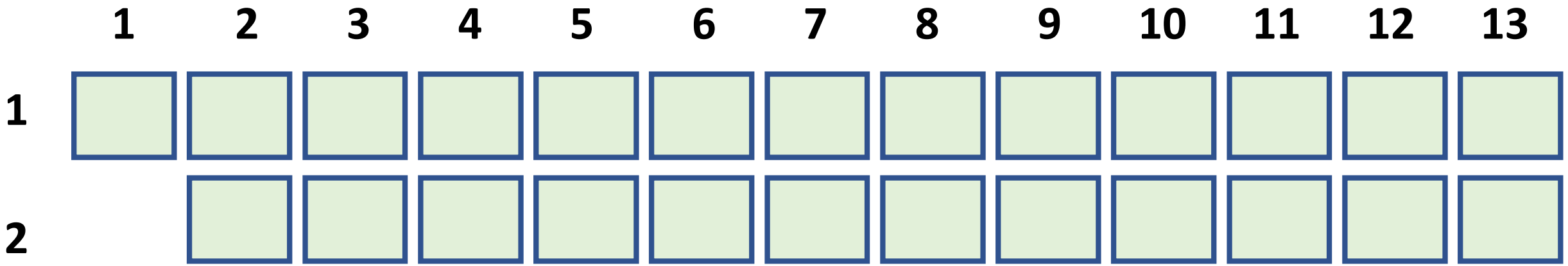
Example

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

M=42

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

$$S_{i,j} = S_{i,j-1} - W_j - 1$$
$$S_{i,j} = M - |w_i|$$



Example

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

M=42

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

$$S_{i,j} = S_{i,j-1} - W_j - 1$$
$$S_{i,j} = M - |w_i|$$

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

Example

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

M=42

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

$S_{i,j} = S_{i,j-1} - W_j - 1$

$S_{i,j} = M - |w_i|$

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

Example

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

M=42

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

$$S_{i,j} = S_{i,j-1} - W_j - 1$$
$$S_{i,j} = M - |w_i|$$

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

Example

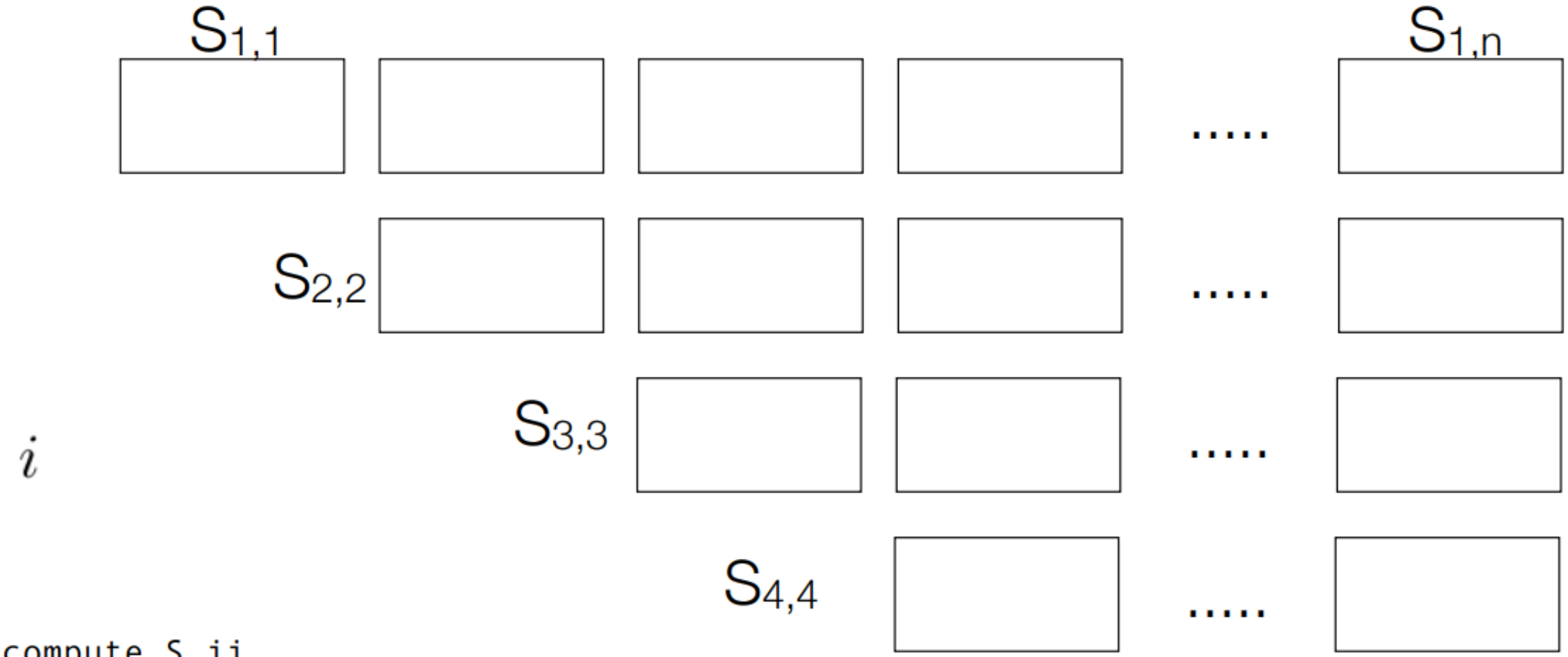
It was the best of times, it was the worst of times; it was the age o
wisdom, it was the age of foolishness; it was the epoch of belief, it
was the epoch of incredulity; it was the season of

M=42

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

$$S_{i,j} = S_{i,j-1} - W_j - 1$$
$$S_{i,j} = M - |w_i|$$

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




```
// compute S_ij
int S[][] = new int[n+1][n+1];
for(int i=1;i<=n;i++) {
    S[i][i] = M - lens[i];
    for(int j=i+1; j<=n; j++) {
        S[i][j] = S[i][j-1] - lens[j] - 1;
        if (S[i][j]<0) {
            while(j<=n) { S[i][j++] = infty; }
        }
    }
}
```

Run-time: $\theta(n^2)$

compute best_0,...,best_n

```
int best[] = new int[n+1];
int choice[] = new int[n+1];
best[0] = 0;
for(int i=1;i<=n;i++) {
    int min = infty;
    int ch = 0;
    for(int j=0;j<I;j++){
        int t = best[j] + S[j+1][i]*S[j+1][i];
        If (t<min){
            min=t;
            ch=j;
        }
    }
    best[i]=min;
    choice[i]=ch;
}
```

For i=1 to n

$$best[i]=min\{best[j]+(s[j+1][i])^2\}$$

Compute Typesetting

$$best[i]=\min_{j=0}^{i-1}\{best[j]+(s[j+1][i])^2\}$$

	0	1	2	3	4
best					

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

Compute Typesetting

$$best[i]=\min_{j=0}^{i-1}\{best[j]+(s[j+1][i])^2\}$$

	0	1	2	3	4
best	0				

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

Compute Typesetting

$$best[i]=\min_{j=0}^{i-1}\{best[j]+(s[j+1][i])^2\}$$

	0	1	2	3	4
best	0				

$$best[1]= best[0]+(s[1][1])^2\} = (40)^2$$

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

Compute Typesetting

$$best[i]=\min_{j=0}^{i-1}\{best[j]+(s[j+1][i])^2\}$$

	0	1	2	3	4
best	0	1600			

$$best[1]= best[0]+(s[1][1])^2\} = (40)^2$$

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

Compute Typesetting

$$best[i]=\min_{j=0}^{i-1}\{best[j]+(s[j+1][i])^2\}$$

	0	1	2	3	4
best	0	1600			

$$best[2]=\min\begin{cases} best[0]+(s[1][2])^2 = (36)^2 \end{cases}$$

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

Compute Typesetting

$$best[i]=\min_{j=0}^{i-1}\{best[j]+(s[j+1][i])^2\}$$

	0	1	2	3	4
best	0	1600			

$$best[2]=\min\begin{cases} best[0]+(s[1][2])^2 = (36)^2 \\ best[1]+(s[2][2])^2 = 1600+(39)^2 \end{cases}$$

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

Compute Typesetting

$$best[i]=\min_{j=0}^{i-1}\{best[j]+(s[j+1][i])^2\}$$

	0	1	2	3	4
best	0	1600	1296		

$$best[2]=\min\begin{cases} best[0]+(s[1][2])^2 = (36)^2 \\ best[1]+(s[2][2])^2 = 1600+(39)^2 \end{cases}$$

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

compute best_0,...,best_n

```
int best[] = new int[n+1];
int choice[] = new int[n+1];
best[0] = 0;
for(int i=1;i<=n;i++) {
    int min = infty;
    int ch = 0;
    for(int j=0;j<I;j++){
        int t = best[j] + S[j+1][i]*S[j+1][i];
        If (t<min){
            min=t;
            ch=j;
        }
    }
    best[i]=min;
    choice[i]=ch;
}
```

For i=1 to n

$$best[i]=min\{best[j]+(s[j+1][i])^2\}$$

Run-time: $\theta(n^2)$