# L11

## 4102

2.25.2016

abhi shelat

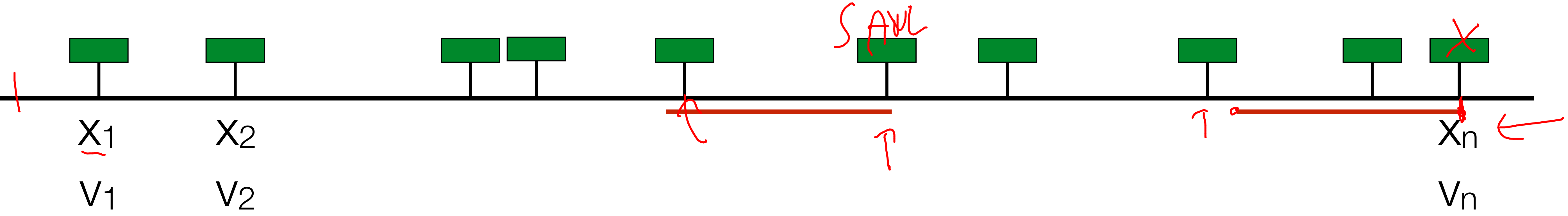
# Billboard problem

$(x_1, x_2, x_3 \ldots x_n)$ $(v_1, v_2 .. v_n)$

↳ mile markers ↖ viewership

$v_i$ = # people that view billboard @ $x_i$.

I-95



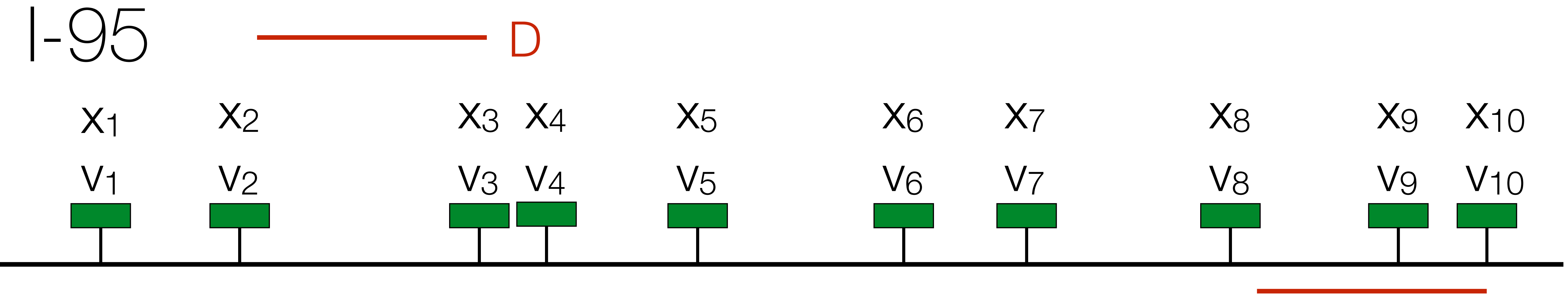
← locations

distance parameter

D

Cannot place ads that are closer than D miles apart

goal: is to maximize viewership for an acceptable campaign



Input is $((x_1, \ldots, x_n)(v_1, \ldots, v_n), D)$

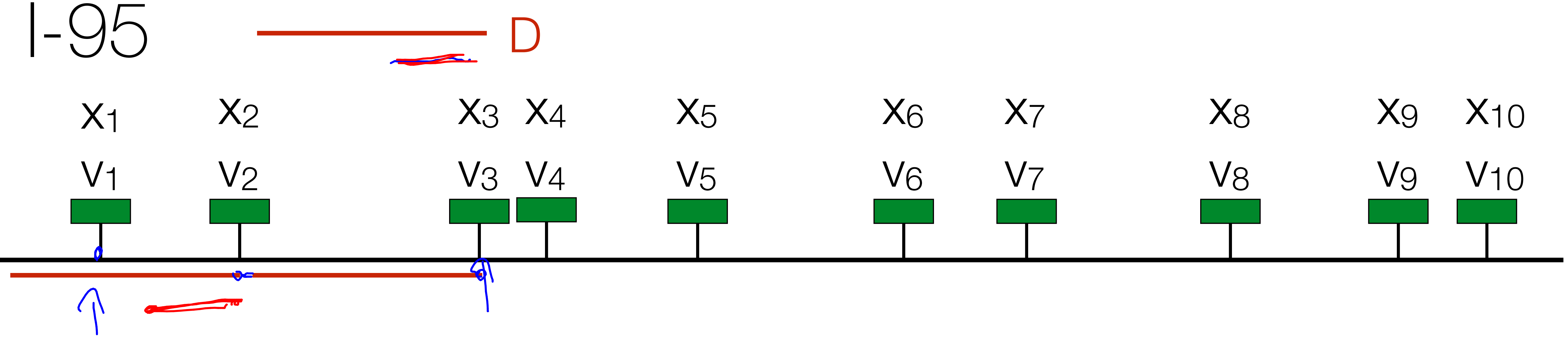
$\underline{\text{Best}_n}$ = max viewership for an acceptable campaign that ~~uses~~ considers the first n billboards

$$\text{Best}_j = \max \begin{cases} \text{Best}_{j-1} \\ V_j + \text{BesT}_{\text{closest billbaard that is D away}} \end{cases}$$

billboad @ $x_j$

buddy(j)

# I-95


D
x1 x2 x3 x4 x5 x6 x7 x8 x9 x10
v1 v2 v3 v4 v5 v6 v7 v8 v9 v10

$\text{Best}_1 = v_j = v_1$

$\text{Best}_2 = \max \begin{cases} \text{Best}_1 \\ v_2 + \underbrace{\text{Best}_{\text{buddy}(2)}}_{} = v_2 \end{cases}$ 0

b/c 2 does not have a buddy

$\text{Best}_3 = \max \begin{cases} \text{Best}_2 \\ v_3 + \text{Best}_{\text{buddy}(3)} = v_3 + \text{Best}_1 \end{cases}$

# Billboard Problem

$$\text{BEST}_j = \max\begin{cases}\text{BEST}_{j-1}\\ v_j + \text{BEST}_{cl(j)}\end{cases}$$

closest = buddy

```
best[0] = 0
for i=1 to n
    cl=i-1
    while (dist(x[cl],x[i]) < D)   cl=cl-1;

    best[i] = max { best[i-1]
                  { v[i]+ best[cl]
return best[n]
```

zero check for cl.

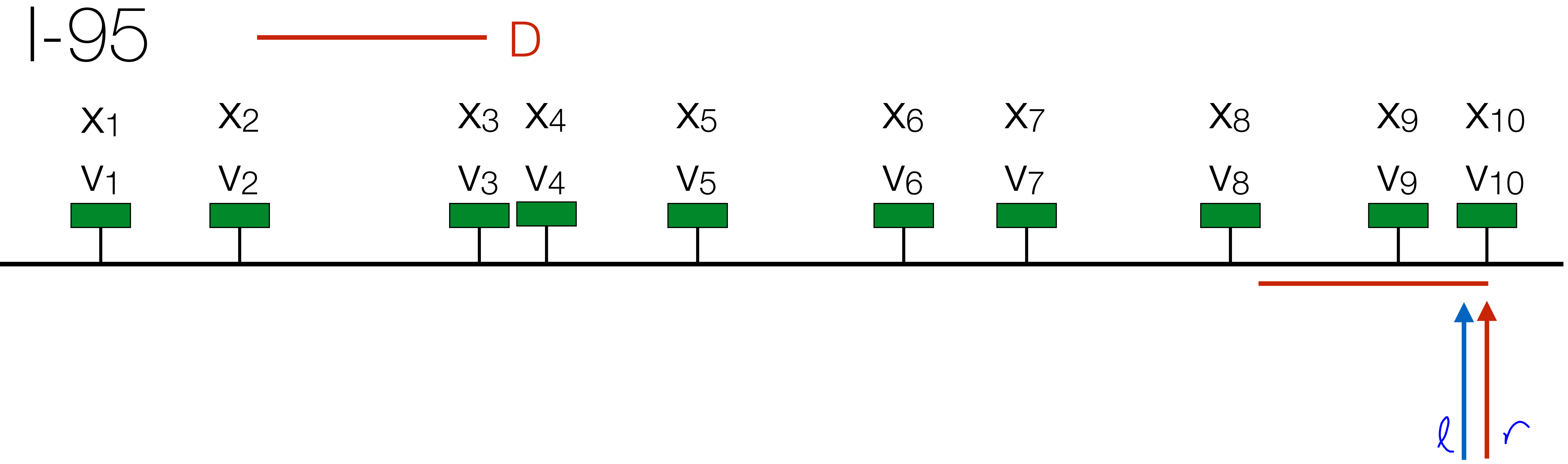
# Billboard Problem

$$\text{BEST}_j = \max \begin{cases} \text{BEST}_{j-1} \\ v_j + \text{BEST}_{cl(j)} \end{cases}$$

RUNTIME : $\Theta(n^2)$

```
best[0] = 0
for i=1 to n  ← n iterations
    cl = i-1
    while( (x[i]-x[cl])< D  && cl>0) cl=cl-1  ← Θ(n)
    best[i] = max(best[i-1], vj+best[cl])
                             vi
return best[n]
```

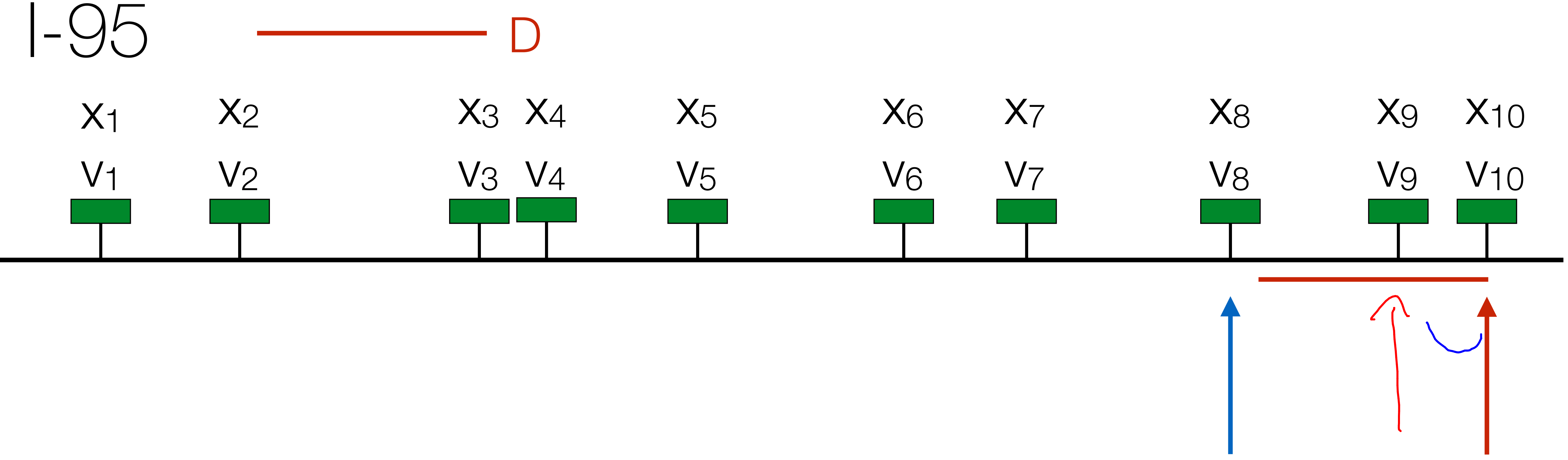
*fix slide


I-95
D
$x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $x_6$ $x_7$ $x_8$ $x_9$ $x_{10}$
$v_1$ $v_2$ $v_3$ $v_4$ $v_5$ $v_6$ $v_7$ $v_8$ $v_9$ $v_{10}$
ℓ
r

Pre-process to find every board's buddy.

right = n, left = n

# I-95



b[10]=8

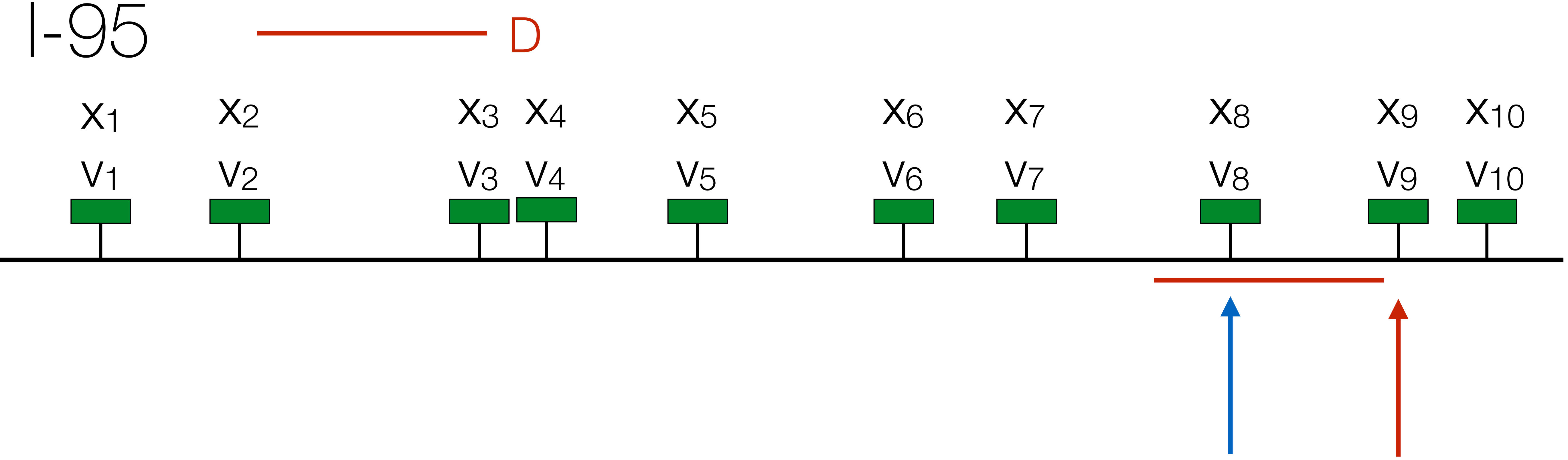
Pre-process to find every board's buddy.

right = n, left = n

move left until dist(x[right], x[left]) > D

buddy[right] = left

# I-95


D
$x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $x_6$ $x_7$ $x_8$ $x_9$ $x_{10}$
$v_1$ $v_2$ $v_3$ $v_4$ $v_5$ $v_6$ $v_7$ $v_8$ $v_9$ $v_{10}$

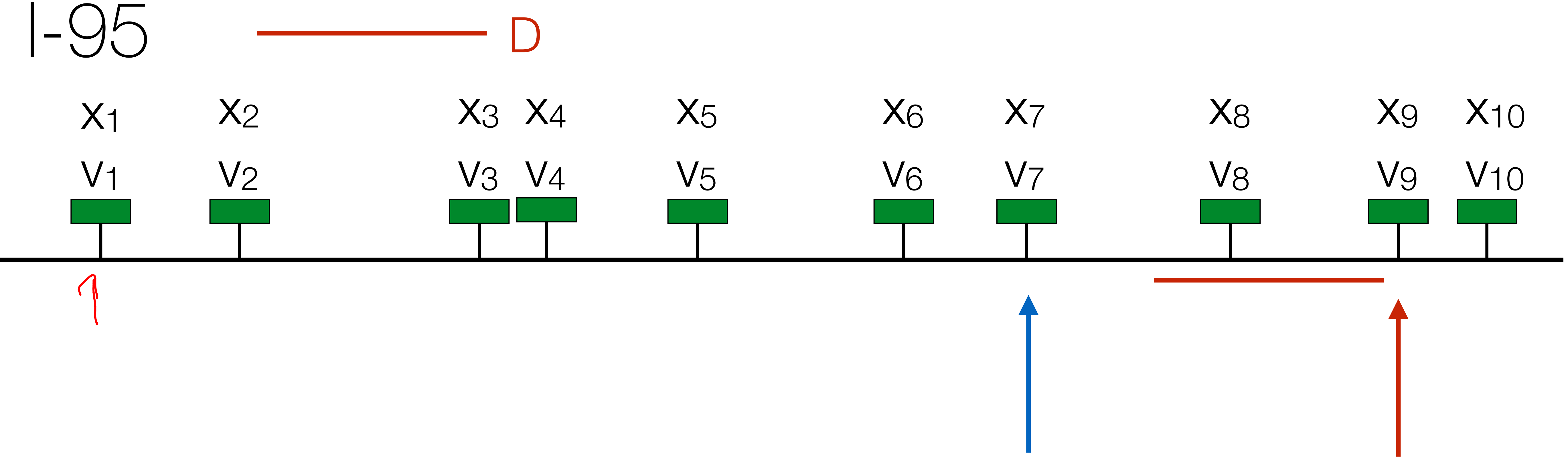
b[10]=8

Pre-process to find every board’s buddy.

```
right = n, left = n

→move left until dist(x[right], x[left]) > D
 buddy[right] = left
 move right to right
```

I-95


D
x1 x2 x3 x4 x5 x6 x7 x8 x9 x10
v1 v2 v3 v4 v5 v6 v7 v8 v9 v10
1

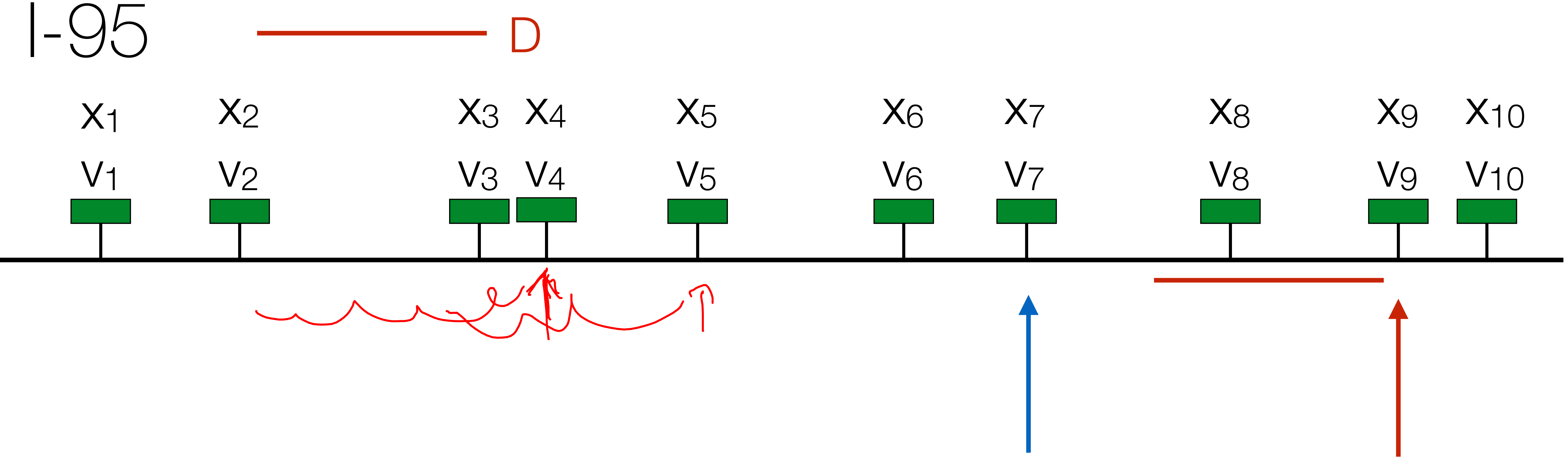
Pre-process to find every board's buddy.

```
right = n, left = n
while right and left are valid
    move left until dist(x[right], x[left]) > D
    buddy[right] = left
    move right to right
```

b[10]=8

b[9) = 7

# I-95

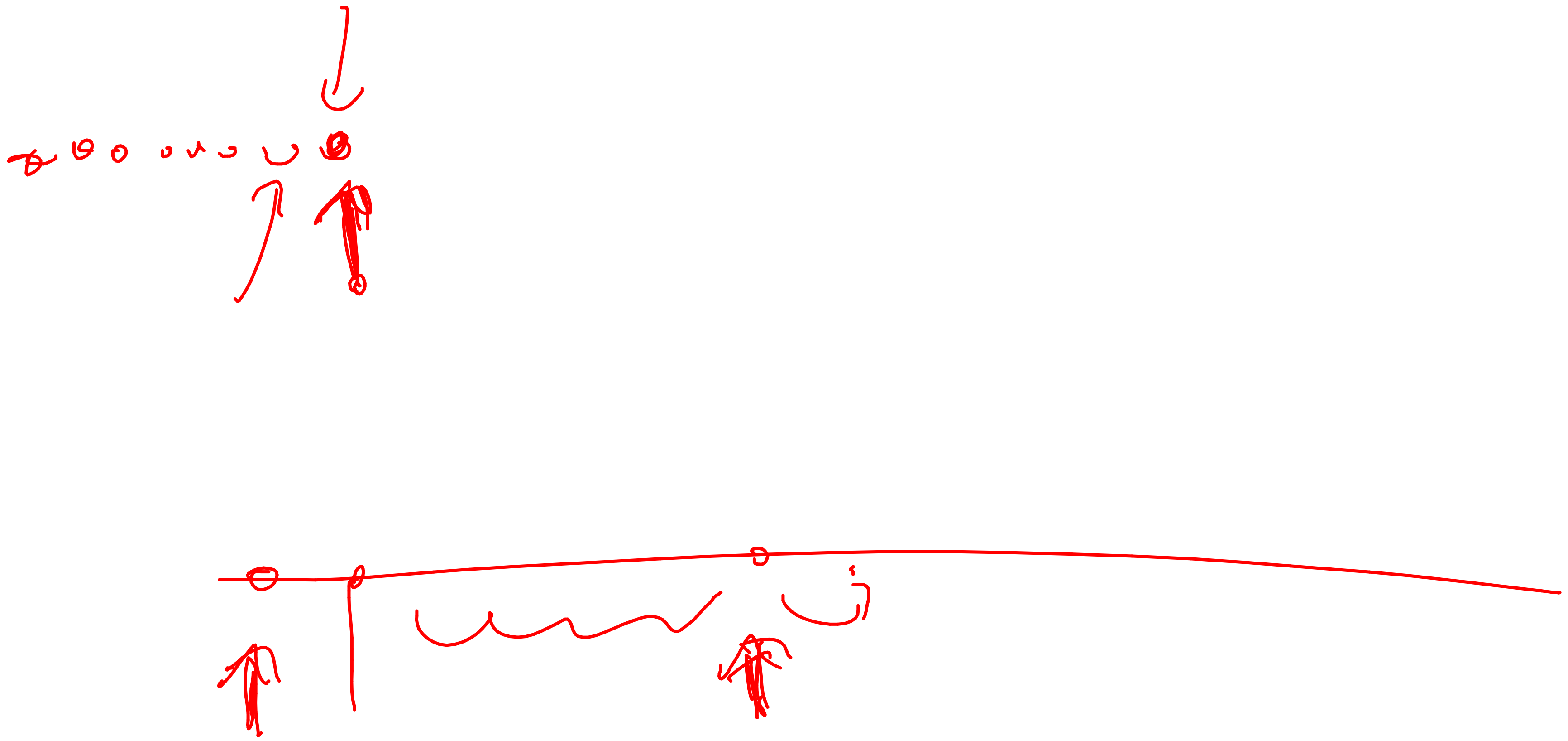

D
$x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $x_6$ $x_7$ $x_8$ $x_9$ $x_{10}$
$v_1$ $v_2$ $v_3$ $v_4$ $v_5$ $v_6$ $v_7$ $v_8$ $v_9$ $v_{10}$

b[10]=8

Pre-process to find every board's buddy.

```
    right = n, left = n
    while right and left are valid
        move left until dist(x[right], x[left]) > D
        buddy[right] = left
        move right to ~~right~~ the left
    handle any leftover right
```

fix !!



# Better Billboard

$$\text{BEST}_j = \max \begin{cases} \text{BEST}_{j-1} \\ v_j + \text{BEST}_{cl(j)} \end{cases}$$

```
<Preprocess buddies> — Θ(n)
best[0] = 0
for i=1 to n  ←  n
   cl = i-1
            r, ℓ
   ~~while( (x[i]-x[cl])< D  && cl>0) cl=cl-1~~ ✗
   best[i] = max(best[i-1], v[j]+best[buddy[i]])   Θ(1)
return best[n]
```

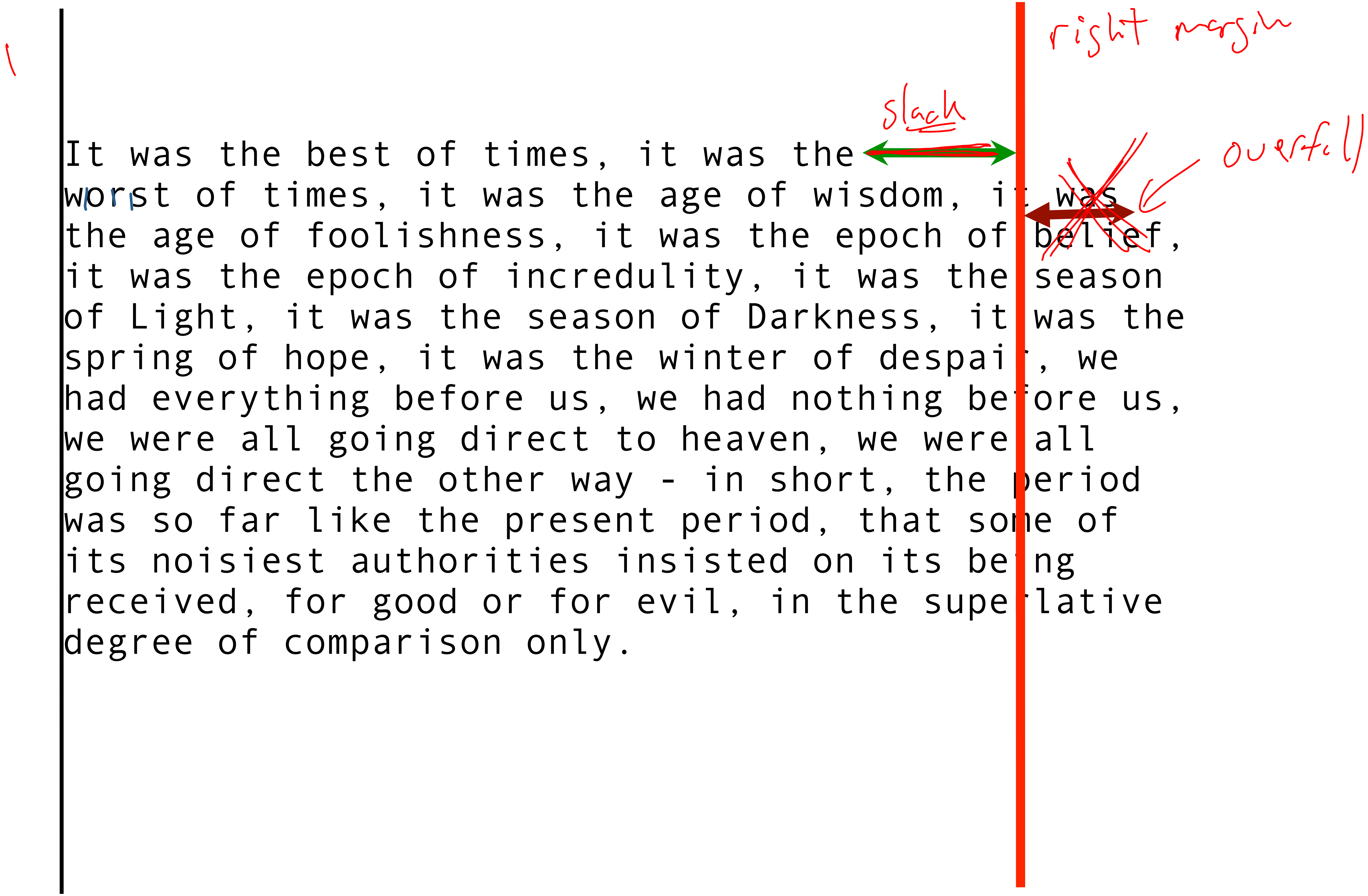
runtime = Θ(n)

fix

# 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.

fix



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

greedy

```
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.
```

minimize the overall slack

penalty for slack is

$slack^2$

______________ is.... slack

# Second rule of typesetting

avoid big ugly whitespaces (slack)

do not typeset in margin ✗

do not typeset in margin ✗

typeset every word ✗

minimize the slack

between margin and last word on a line

one paragraph at a time

greedy typeset

| | | |
|---|---|---|
| It was the best of times, it was the worst | 0 | 0 |
| of times, it was the age of wisdom, it was | 0 | 0 |
| the age of foolishness, it was the epoch | 2 | 4 |
| of belief, it was the epoch of | 12 | 144 |
| incredulity, it was the season of Light, | 2 | 4 |
| it was the season of Darkness, it was the | 1 | 1 |
| spring of hope, it was the winter of | 6 | 36 |
| despair, we had everything before us, we | 2 | 4 |
| had nothing before us, we were all going | 2 | 4 |
| direct to heaven, we were all going direct | 0 | 0 |
| | | 197 |

|  |  |  |
|---|---|---|
| It was the best of times, it was the | 6 | 36 |
| worst of times, it was the age of wisdom, | 1 | 1 |
| it was the age of foolishness, it was the | 1 | 1 |
| epoch of belief, it was the epoch of | 6 | 36 |
| incredulity, it was the season of Light, | 2 | 4 |
| it was the season of Darkness, it was the | 1 | 1 |
| spring of hope, it was the winter of | 6 | 36 |
| despair, we had everything before us, we | 2 | 4 |
| had nothing before us, we were all going | 2 | 4 |
| direct to heaven, we were all going direct | 0 | 0 |
|  |  | 123 |

# Typesetting problem

input: $\underbrace{W = \{w_1, w_2, w_3, \ldots, w_n\}}_{\text{sequence of word lengths}}$ $M \leftarrow$ margin

output: $\underset{\text{lines}}{\underline{L}} = (\underline{w_1}, \ldots, w_{\ell_1}), (\underline{w_{\ell_1+1}}, \ldots, w_{\ell_2}), \ldots, (\underline{w_{\ell_x}}_{+1}, \ldots, w_n)$

such that all words typeset

$$\sum_{i \in \text{each line}} w_i \leq M$$

minimize

$$\sum_{j \in \text{lines}} \left(M - \left(\underset{\text{line } j}{\text{sum of words on}}\right) - \underset{\text{line } j}{\#\text{of words on}} + 1\right)^2$$

# Typesetting problem

input: $W = \{w_1, w_2, w_3, \ldots, w_n\}$ $\quad M$

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

$$c_i = \left( \sum_{j=\ell_i+1}^{\ell_{i+1}} |w_j| \right) + \underbrace{(\ell_{i+1} - \ell_i - 1)}_{\text{spaces b/w words}}$$

length of the $i^{th}$ line in the typesetting

$\ell_{i+1}$

$j = \ell_i + 1$

such that $\quad c_i \leq M \quad \forall i$

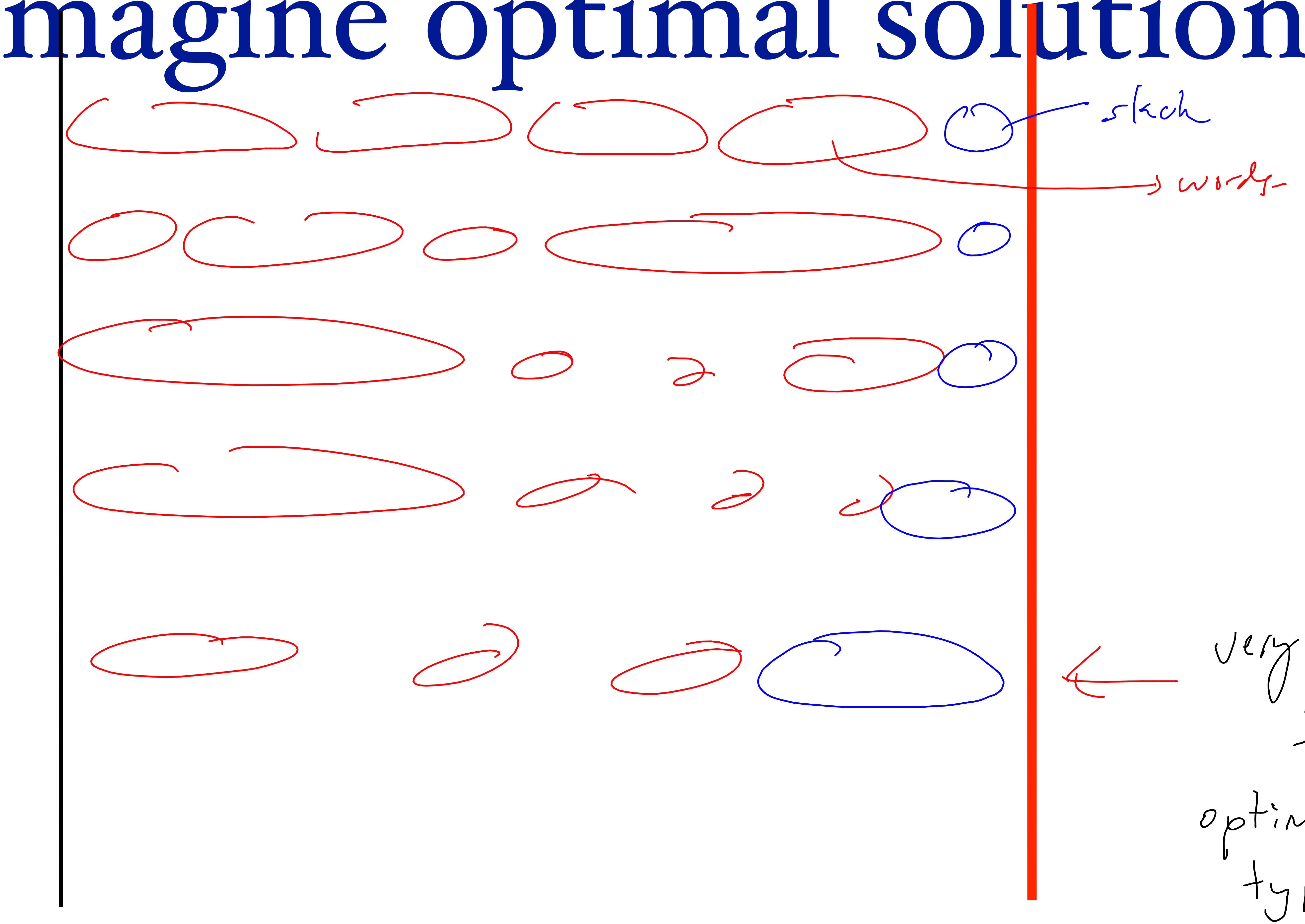
$$\min \sum (M - c_i)^2$$

# how to solve

define the right variable:

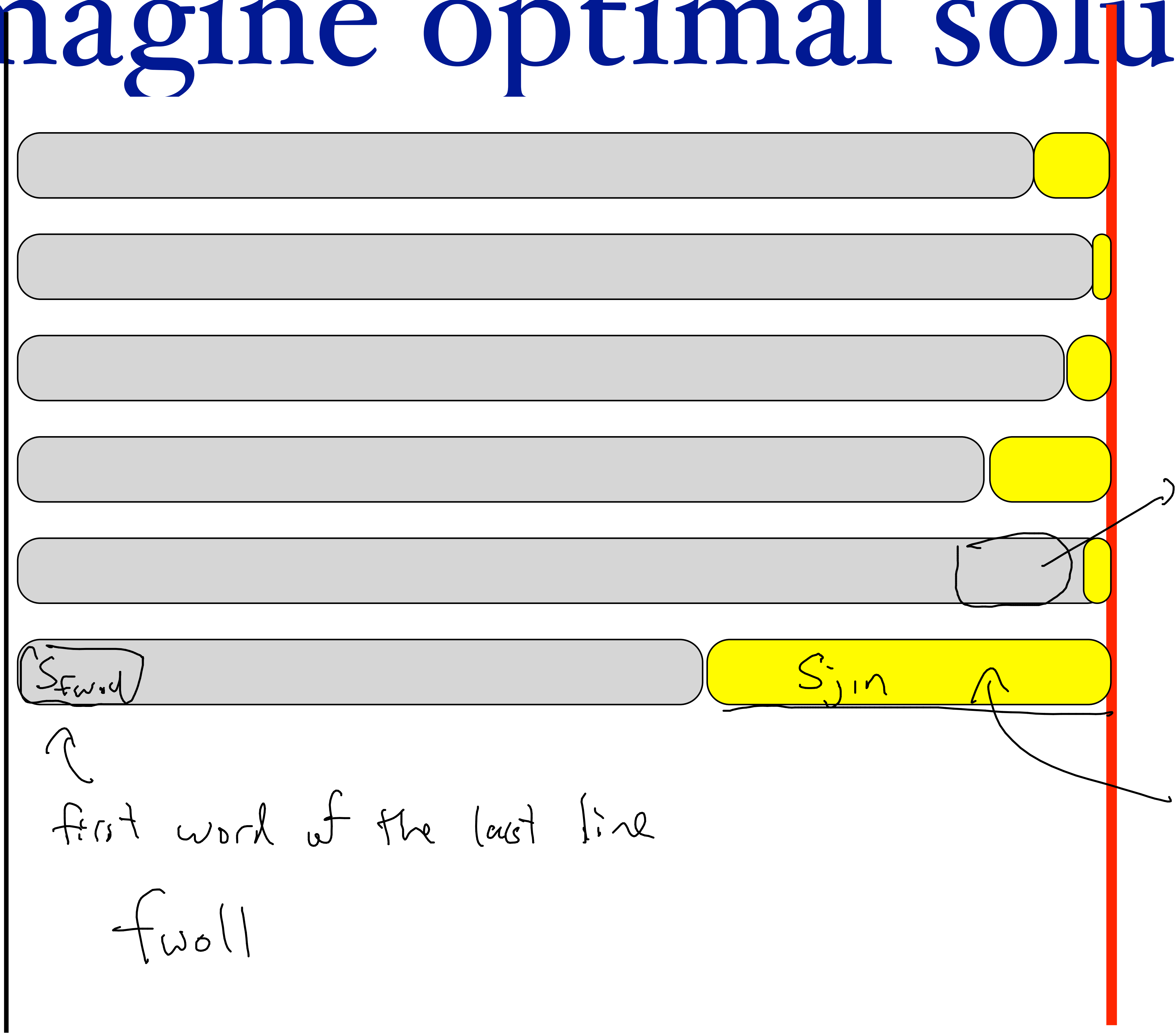
$Best_n$: minimum penalty for typesetting the first $n$ words

# imagine optimal solution



very last line that the optimal solution typesets

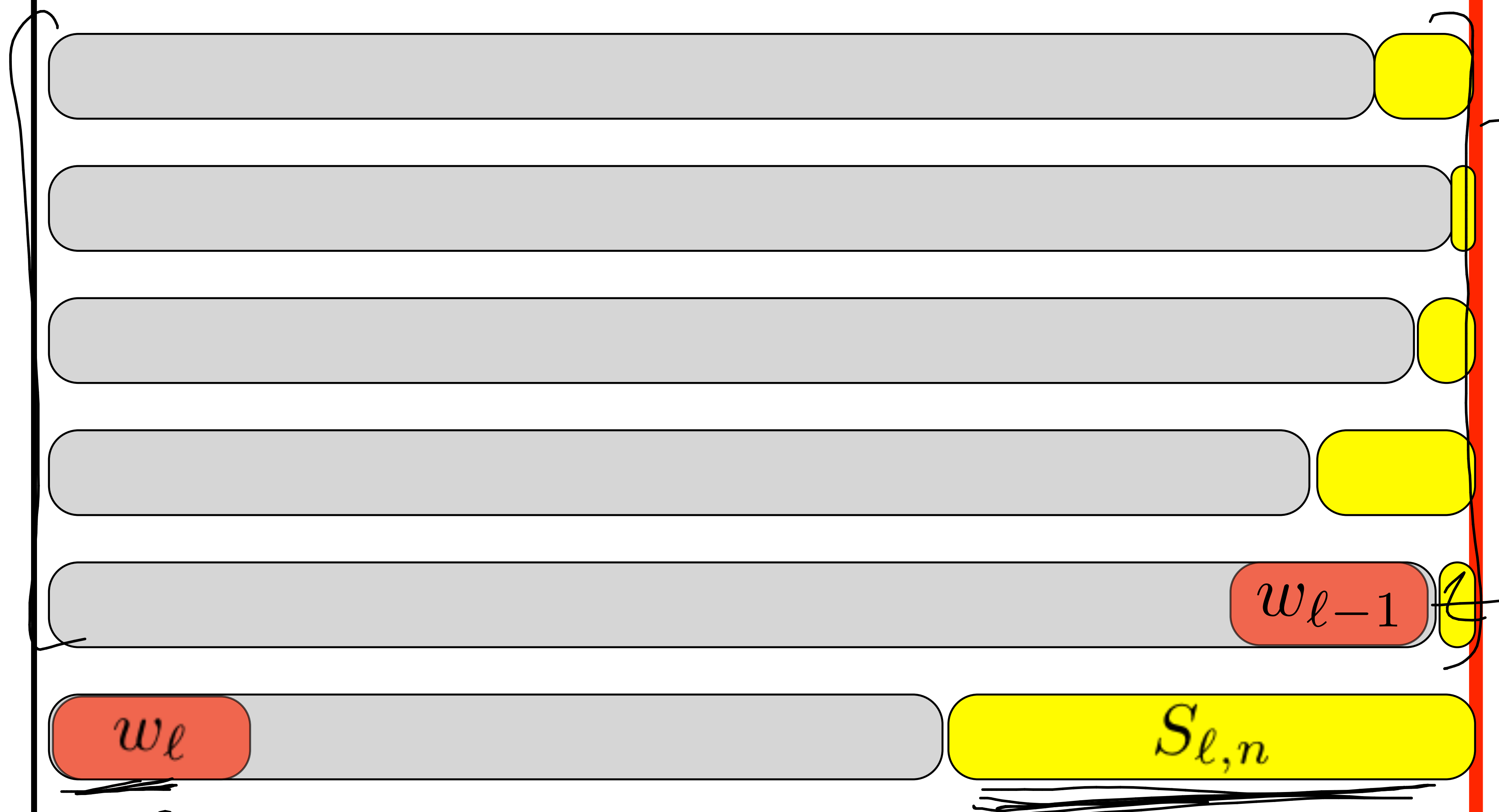
# imagine optimal solution



$S_{fword}$
$S_{j,n}$
last word before fword
last line
first word of the last line
fwoll
slack induced by typesetting word j as fword for first n words.

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

# imagine optimal solution



$\text{Best}_n = \text{Best}_{\text{fwoll}^*-1} + S_{\text{fwoll},n}^2$

$\text{Best}_{\text{fwoll}-1}$ i.e. penalty for typesetting the earlier part of paragraph.

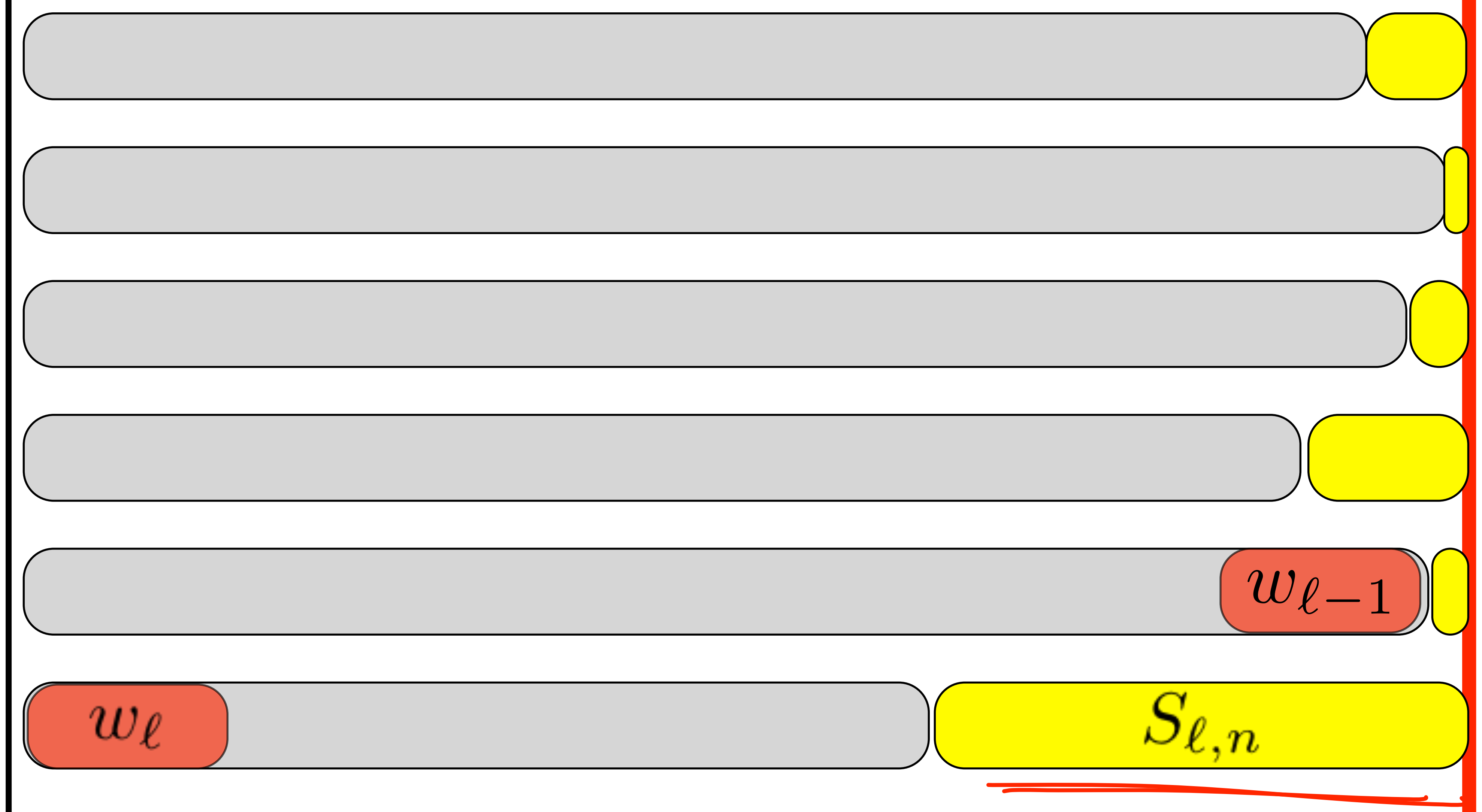
last line

fwoll is $w_\ell^*$

optimal

slack when line starts with $w_\ell$

# imagine optimal solution



last line

fwoll is $w_\ell$

slack when line starts with $w_\ell$

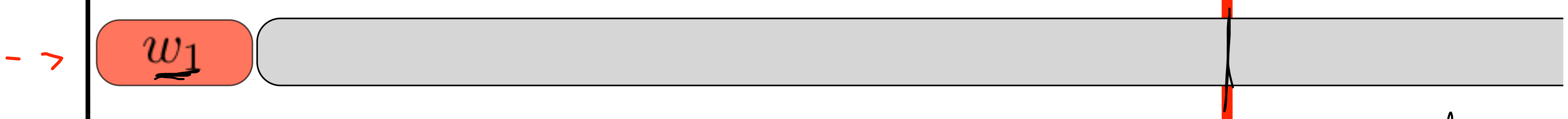
$$\mathrm{BEST}_n = \mathrm{BEST}_{\ell-1} + S_{\ell,n}^2$$

# how many candidates are there for the fwoll?

$$\text{Best}_n = \min_{f=1}^{n} \left\{ \underline{\text{Best}_{f-1}} + \underline{(S_{f,n})^2} \right.$$

candidates ?!

# is $w_1$ fwoll?

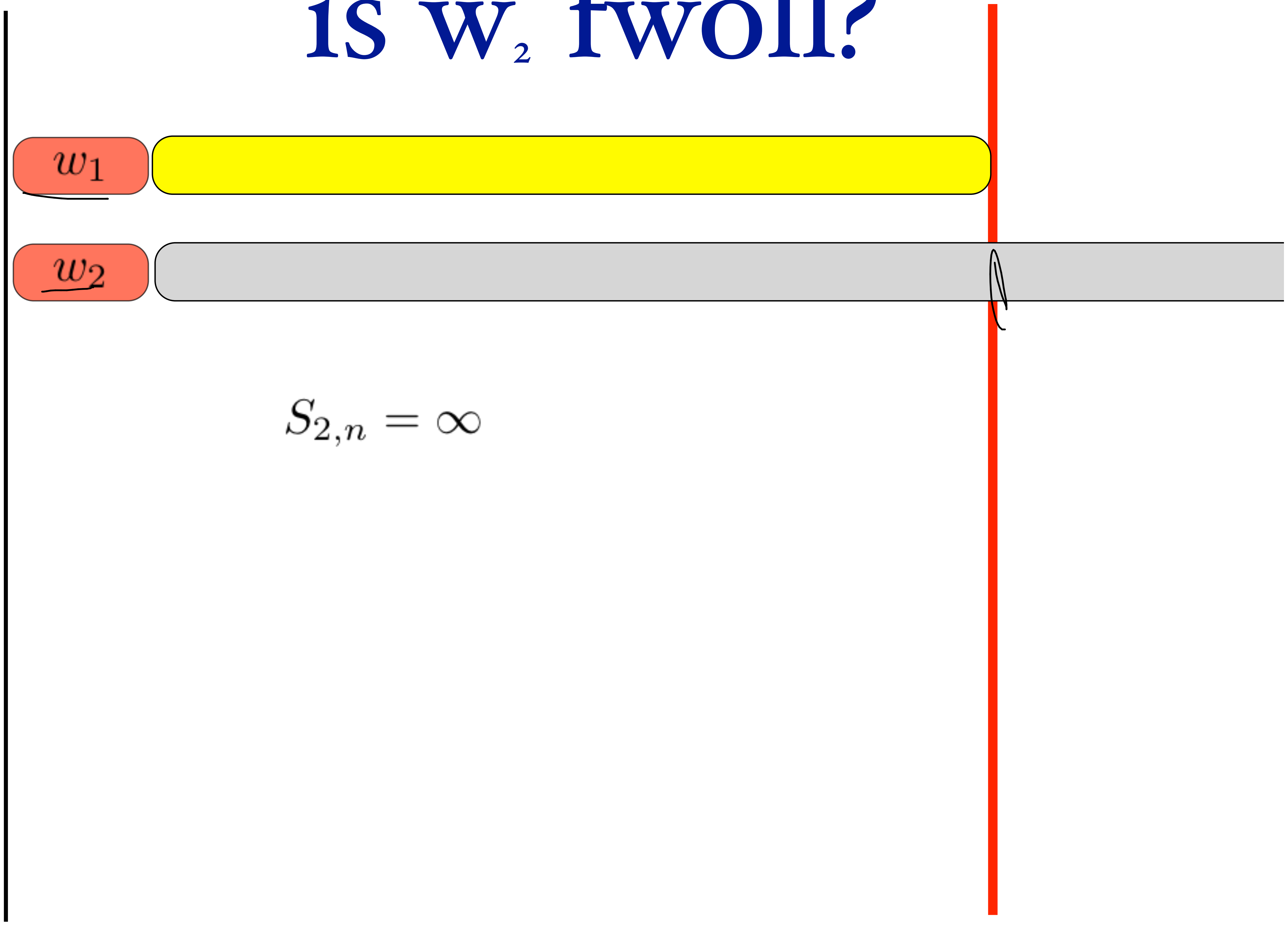


exceed margin if I try to typeset all words on 1 line.

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

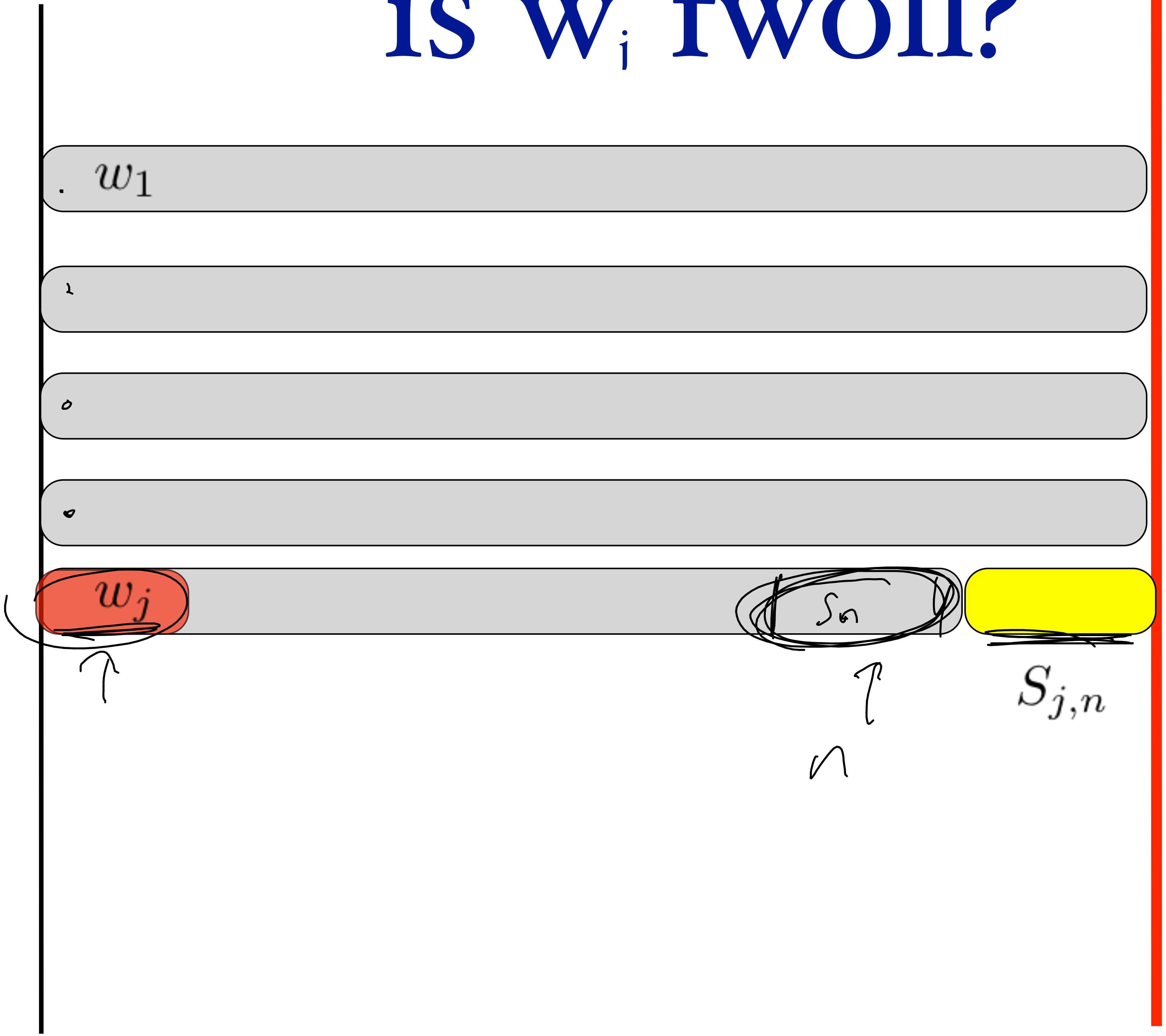
define $S_{1,n} = \infty$ if this happens

# is $w_2$ fwoll?



$$S_{2,n} = \infty$$

# is w$_j$ fwoll?



# which word is fwoll?

$$\mathrm{BEST}_n = \min \left\{ \vphantom{\begin{matrix}a\\b\\c\\d\\e\\f\end{matrix}} \right.$$

# which word is fwoll?

$$\mathrm{BEST}_n = \underline{\min} \left\{ \begin{array}{l} \mathrm{BEST}_0 + S^2_{1,n} \\ \mathrm{BEST}_1 + S^2_{2,n} \\ \mathrm{BEST}_2 + S^2_{3,n} \\ \cdots \\ \mathrm{BEST}_{\ell-1} + S^2_{\ell,n} \\ \\ \mathrm{BesT}_{n-2} + S^2_{n-1,n} \\ \underline{\mathrm{BEST}_{n-1}} + \underline{S^2_{n,n}} \end{array} \right. = \min_{f=1} \left\{ \mathrm{BesT}_{f-1} + (S_{f,n})^2 \right.$$

# typesetting algorithm

# typesetting algorithm

(make table for $\underline{S_{i,j}}$) → slack for typesetting word i as fwoll among first j words

$\underline{\text{for } \mathtt{i=1} \text{ to } \mathtt{n}}$

$$\text{best}[\mathtt{i}] = \min_{j}\{\ \underline{\text{best}[\mathtt{j}]} + \underline{\text{s}[\mathtt{j+1}][\mathtt{i}]^2}\ \}$$

# typesetting algorithm

make table for $S_{i,j}$

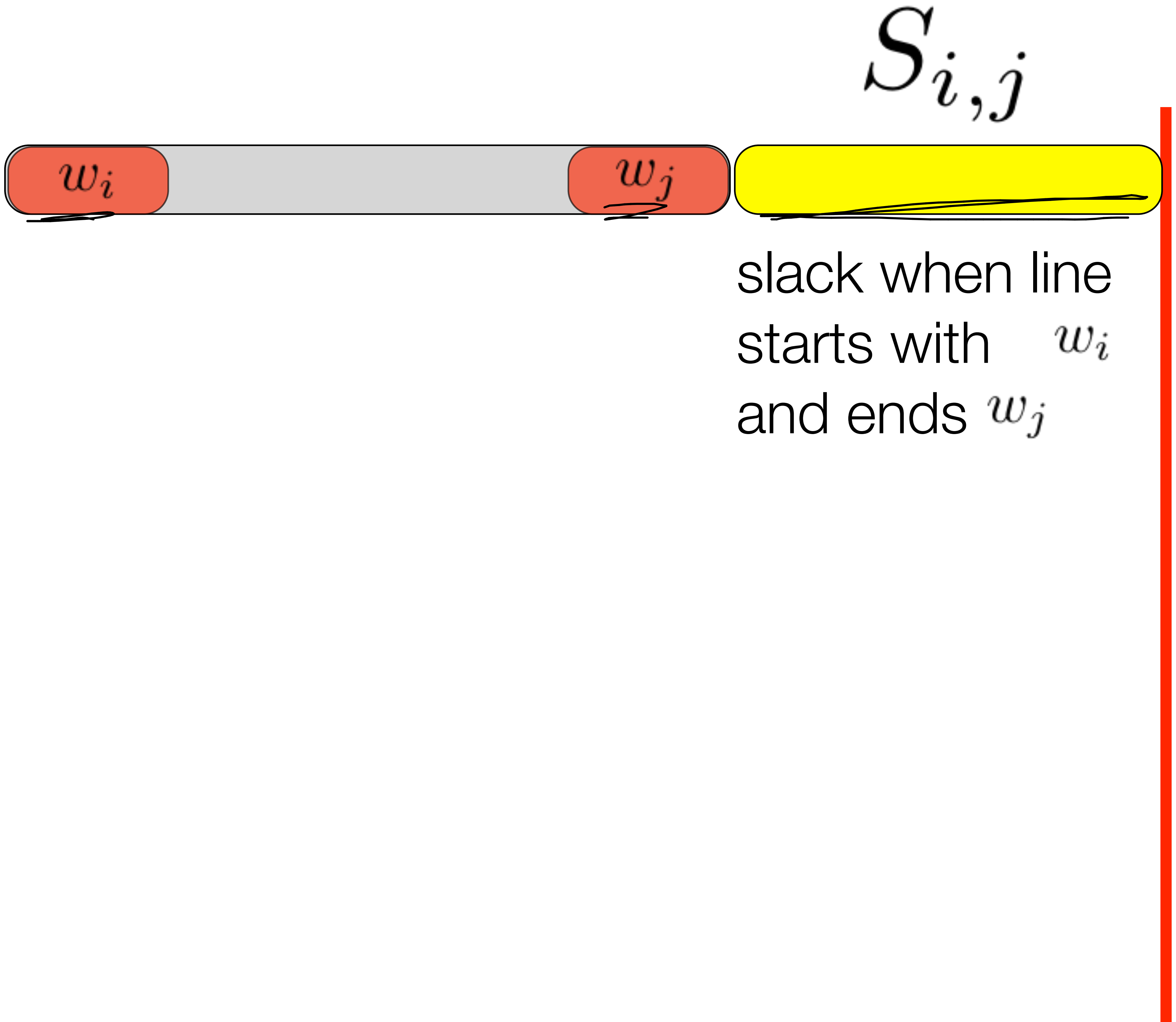
```
for i=1 to n
    best[i] = min{ best[j] + s[j+1][i]^2 }
```

best
choices correspond to

twal

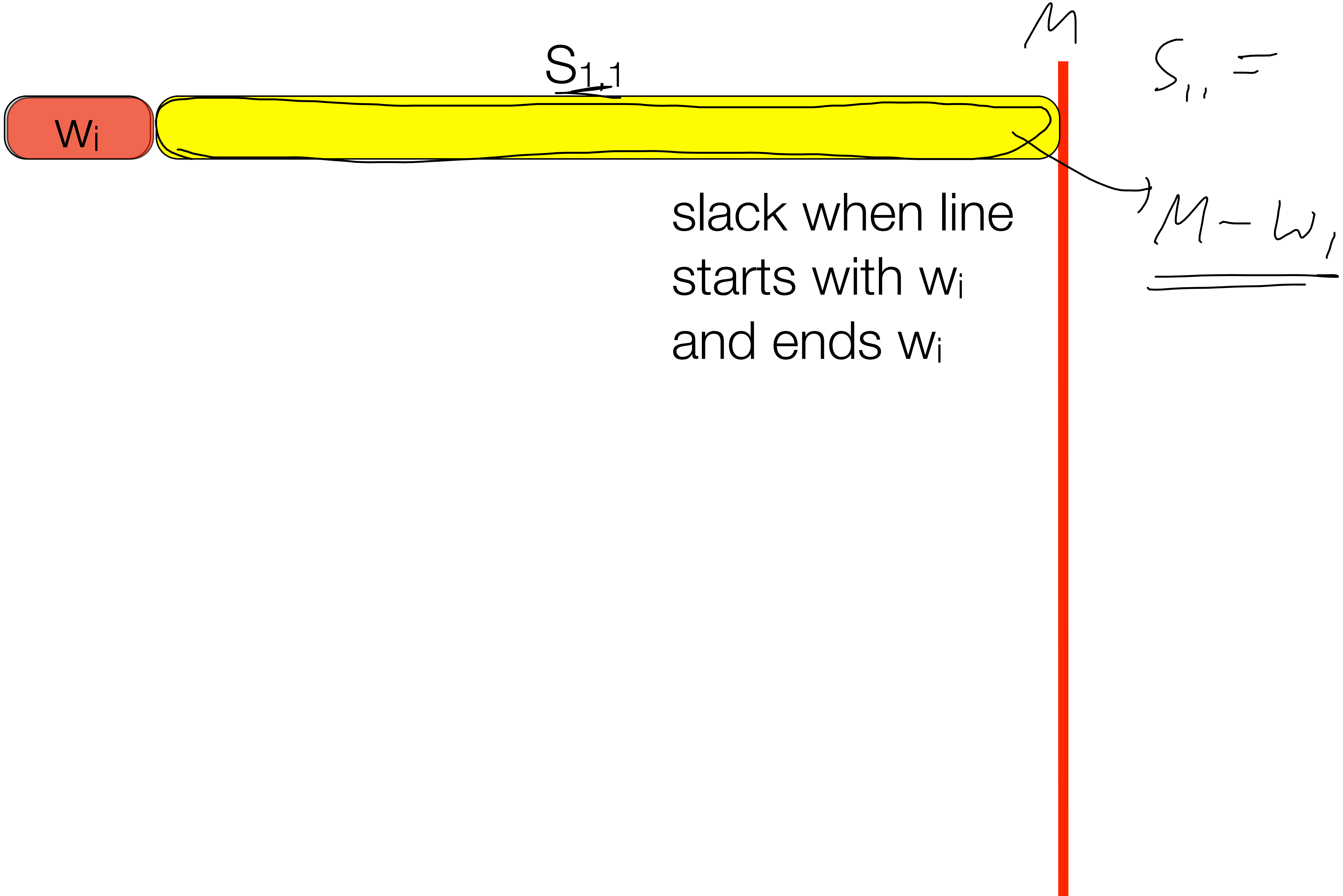
```
// compute best_0,...,best_n
int best[] = new int[n+1];
int choice[] = new int[n+1];
best[0] = 0;
for(int i=1;i<=n;i++) {
    int min = infty;
    int ch  = 0;
    for(int j=0;j<i;j++) {
        int t = best[j] + S[j+1][i]*S[j+1][i];
        if (t<min) { min = t; ch = j;}
    }
    best[i] = min;
    choice[i] = ch;
}
```

# how to compute $S_{i,j}$



slack when line starts with $w_i$ and ends $w_j$

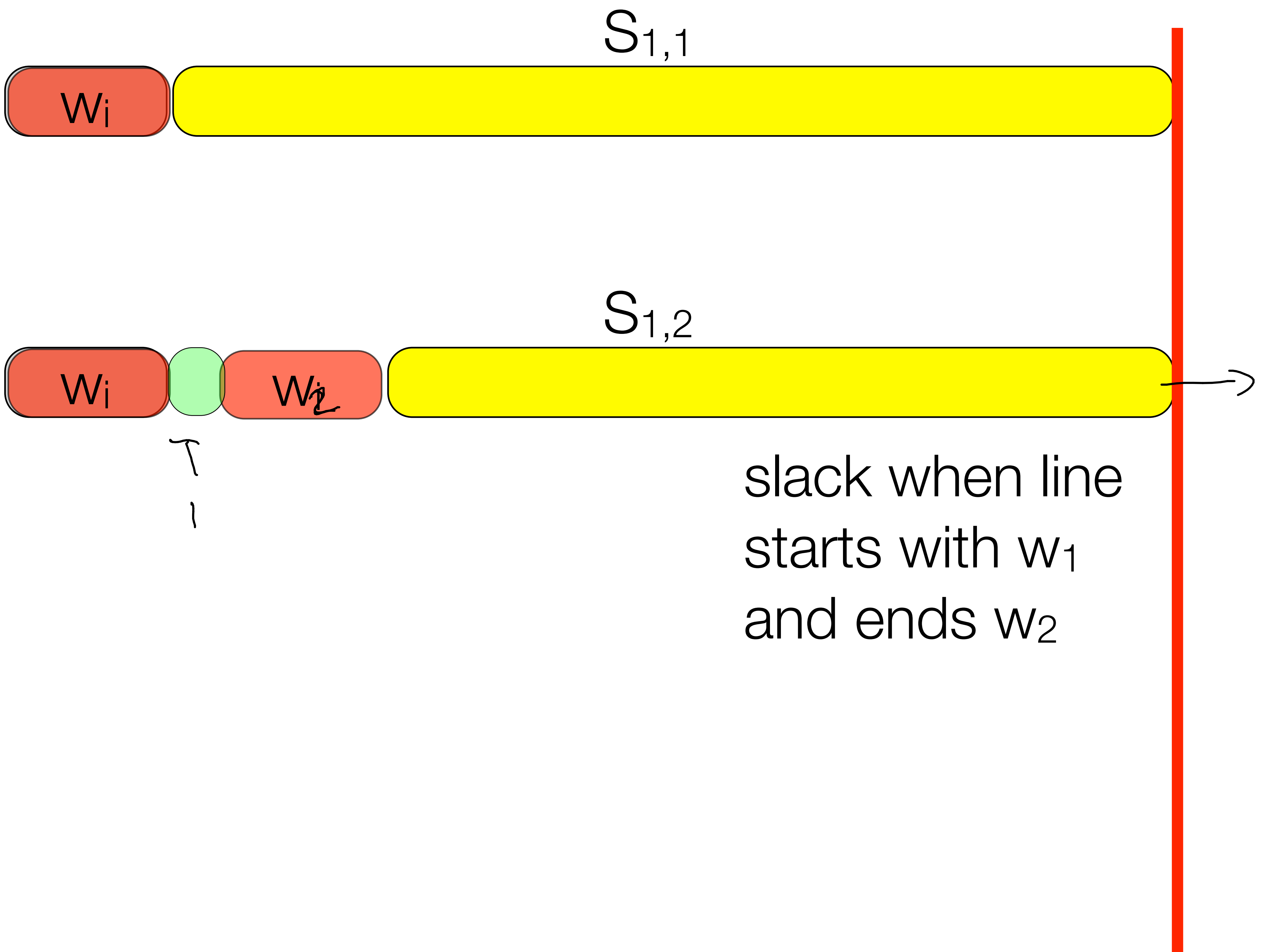
# Simplest case



slack when line starts with $w_i$ and ends $w_i$

$S_{11} = M - w_1$

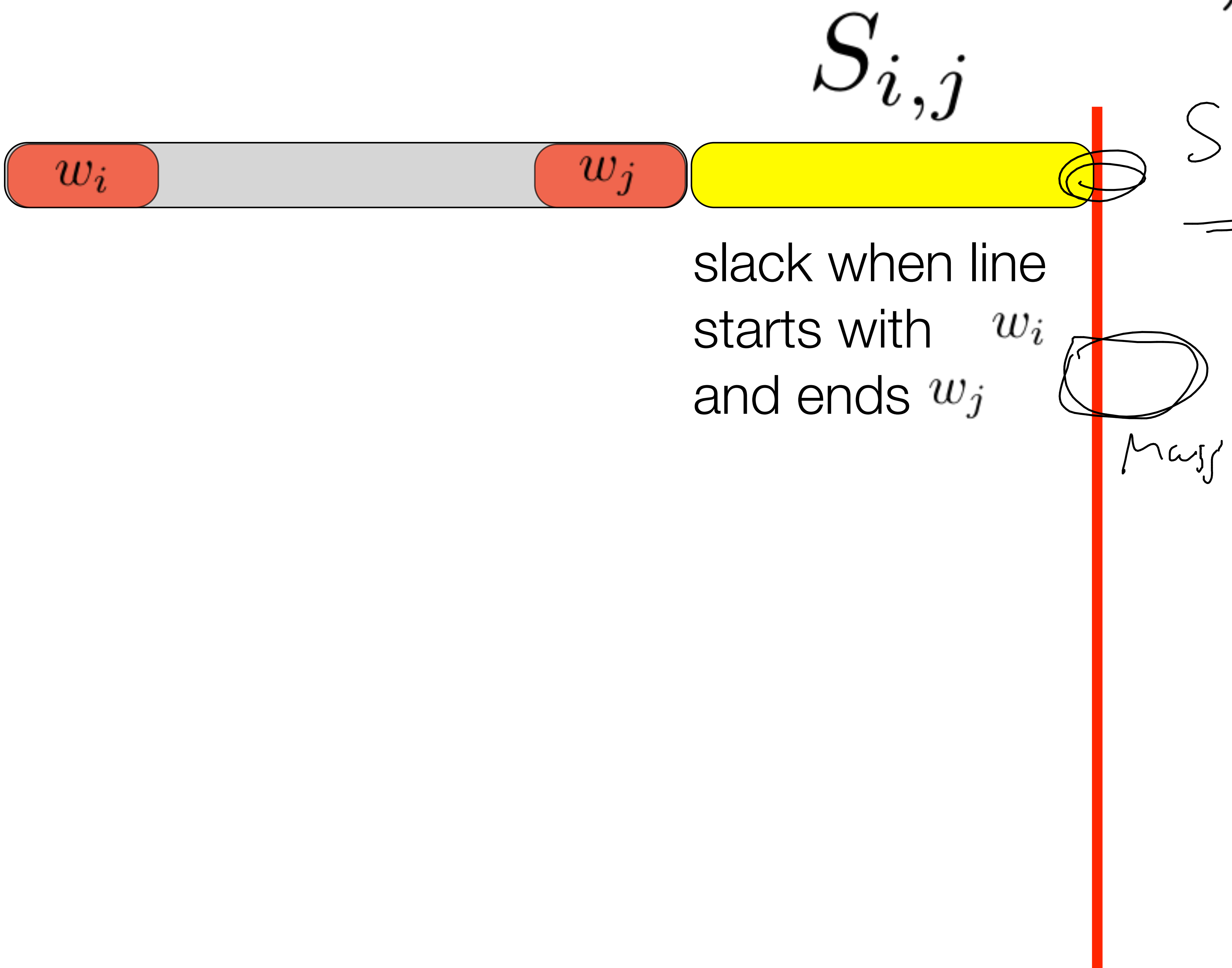
# Simplest case



$\rightarrow S_{12} = S_{11} - w_2 - 1$

slack when line
starts with $w_1$
and ends $w_2$

# how to compute $S_{i,j}$



slack when line starts with $w_i$ and ends $w_j$

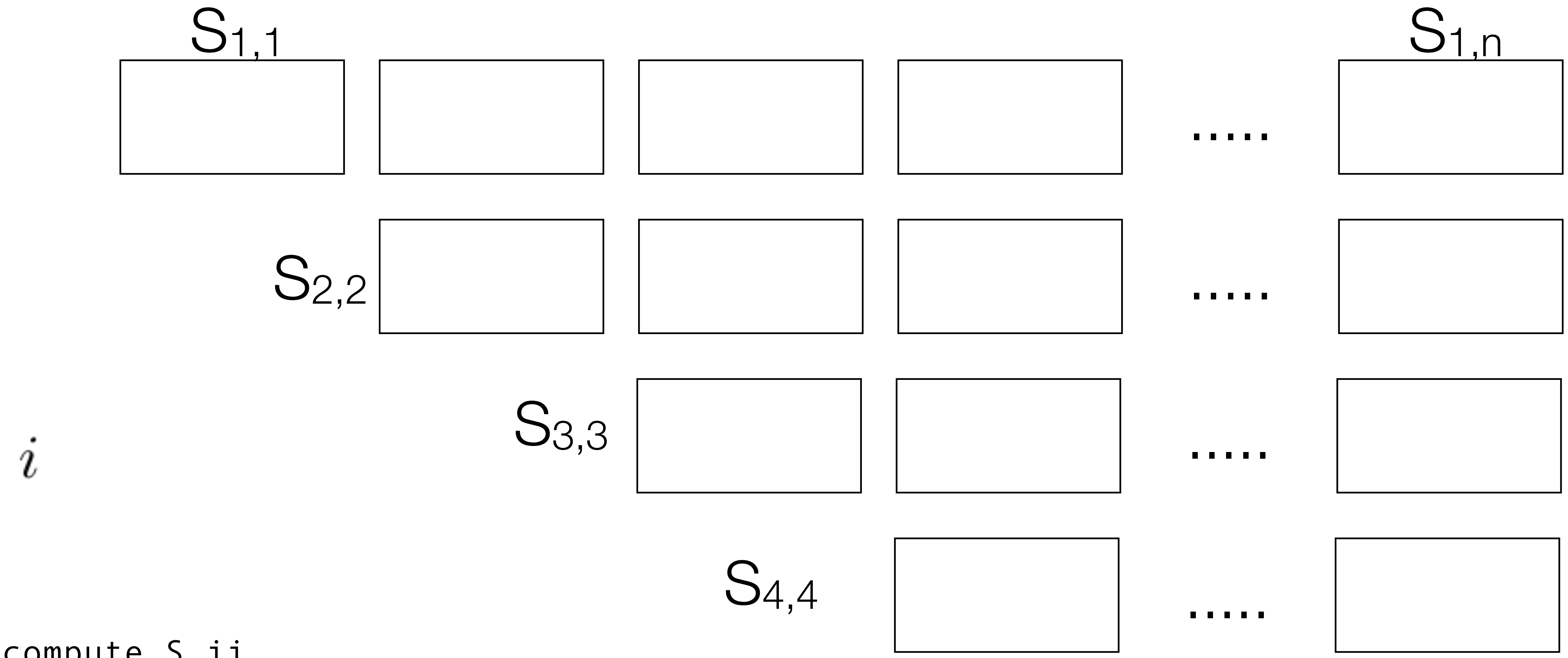
$$S_{ij} = S_{i,j-1} - w_j - 1$$

if $S_{ij} < 0$,

set

$S_{ij} = \infty$

# How to compute S



$S_{1,1}$

$S_{1,n}$

.....

$S_{2,2}$

.....

$i$

$S_{3,3}$

.....

$S_{4,4}$

.....

```
// 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
```

```
  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 $S_{i,j}$

|  | 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 |  |  |  |  |  |  |  |  |

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

$M = 42$

slack

$$\begin{cases} \underline{S_{i,i}} = M - |w_i| \\ \underline{S_{i,j}} = S_{i,j-1} - 1 - |w_j| \end{cases}$$

$S_{1,1} = 42 - 2 = 40$

$S_{1,2} = 40 - 3 - 1 = 36$

$S_{1,3} = 36 - 3 - 1 = 32$

$S_{2,2} = 42 - 3 = 39$

# 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

$M = 42$

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

$$S_{i,j} = S_{i,j-1} - 1 - |w_j|$$

# 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 |

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

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| best | 0 | 1600 | 1256 | | | | | | | | | |

$$\text{Best}_1 = \text{Best}_0 + (S_{1,1})^2 = 40^2$$

$$\text{Best}_2 = \min \begin{cases} \text{best}_0 + (S_{1,2})^2 & 36^2 \\ \text{best}_1 + (S_{2,2})^2 & 1600 + 39^2 \end{cases}$$

$$\text{Best}_3 = \min \begin{cases} \underline{B_0 + S_{1,3}{}^2} \\ B_1 + S_{2,3}{}^2 \\ B_2 + S_{3,3}{}^2 \end{cases}$$

$$\underline{\underline{\text{BEST}_i}} = \min_{j=0}^{i-1}\left\{\text{BEST}_j + \underline{S_{j+1,i}}^2\right\}$$

| $S_{ij}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11<br>∞ | 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

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| best | 0 | 1600 | | | | | | | | | | |

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

| | 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

|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| best | 0 | 1600 | 1296 |  |  |  |  |  |  |  |  |  |

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

|  | 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 |

# Running time

make table for $S_{i,j}$

$\Theta(n^2)$

```
for i=1 to n
    best[i] = min{ best[j] + s[j+1][i]^2 }
```