L8 5800

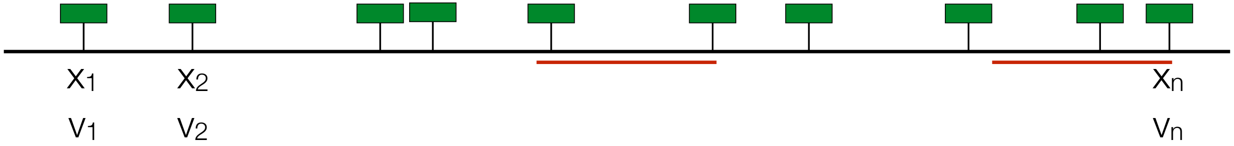
feb 11/14 2022

shelat

Billboard problem

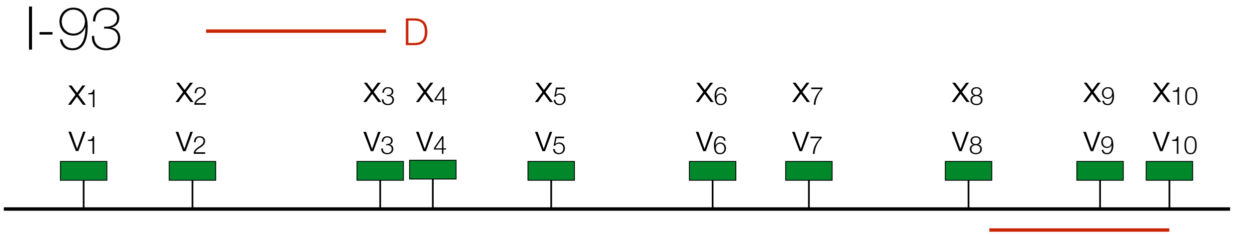
i'm lovin' it
LETTUCE.TOMATO.MAYO.
ON THE DOUBLE.
new
daily double® $1 99
For a limited time only. Prices and participation may vary. ©2012 McDonald's.

I-93



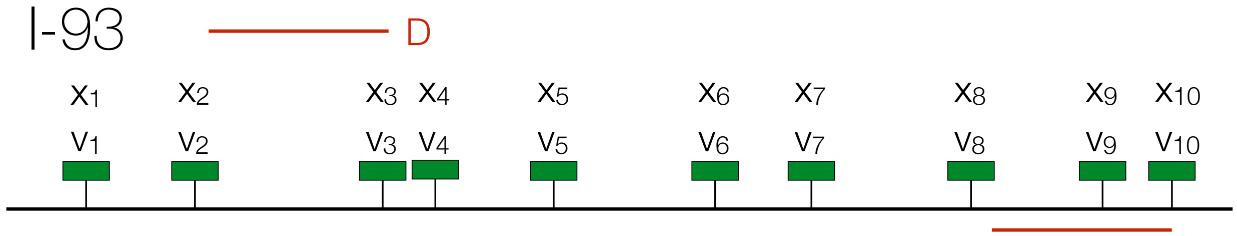
D

distance parameter
Cannot place ads that are closer than D miles apart


I-93
D
x1 x2 x3 x4 x5 x6 x7 x8 x9 x10
v1 v2 v3 v4 v5 v6 v7 v8 v9 v10

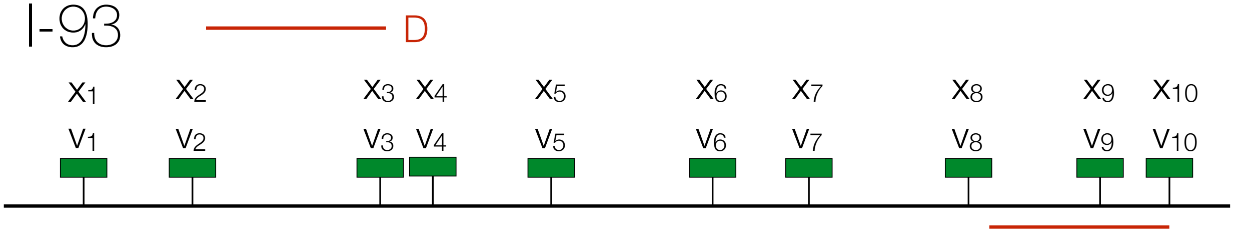
Input is $((x_1, \ldots, x_n)(v_1, \ldots, v_n), D)$

Best$_n$ =



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

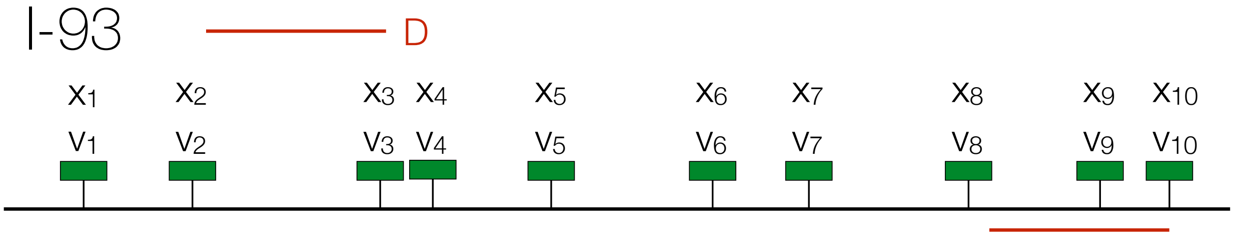
Best$_n$ = Max viewers for a campaign that uses billboards {1…n} with separation D.



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

Best_n = Max viewers for a campaign that uses billboards {1…n} with separation D.

$$Best_n =$$



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

Best$_n$ = Max viewers for a campaign that uses billboards {1…n} with separation D.

$$Best_n = \max \begin{cases} Best_{n-1} \\ v_n + Best_{closest_D(n)} \end{cases}$$

Familiar?

Familiar?

$$Best_n =$$

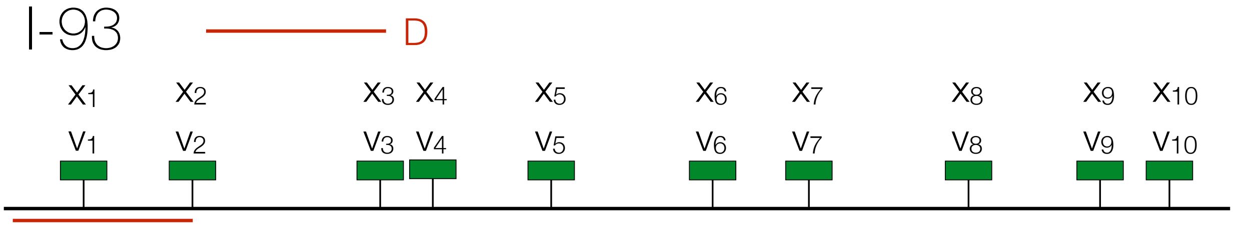
Familiar?

$$Best_n = \max \begin{cases} Best_{n-1} \\ v_n + Best_{closest_D(n)} \end{cases}$$

Familiar?

$$Best_n = \max \left\{ \begin{array}{c} Best_{n-1} \\ v_n + Best_{closest_D(n)} \end{array} \right.$$

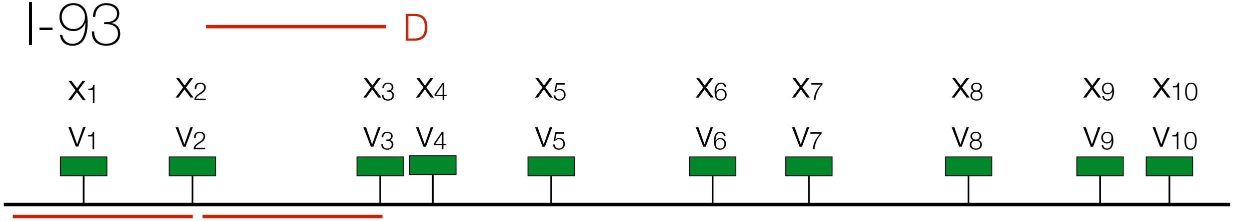
This equation is very similar to the log-cutter equation, with one difference.
We cannot simply use the price to pick the sub-problem, we have to use D:



$Best_1 =$

$Best_2 =$

I-93


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}

$\text{Best}_1 =$

$\text{Best}_2 =$

$\text{Best}_3 =$

Billboard Problem

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

```
best[0] = 0
for i=1 to n



return best[n]
```

Billboard Problem

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

```
best[0] = 0
for i=1 to n
    cl = i-1
    while( (x[i]-x[cl])< D  && cl>0) cl=cl-1
    best[i] = max(best[i-1], $v_i$+best[cl])
return best[n]
```

Billboard Problem

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

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

This line can take $\Theta(i)$ steps in the worst case.

Running time (worst case): $\Theta(n^2)$

Billboard Problem

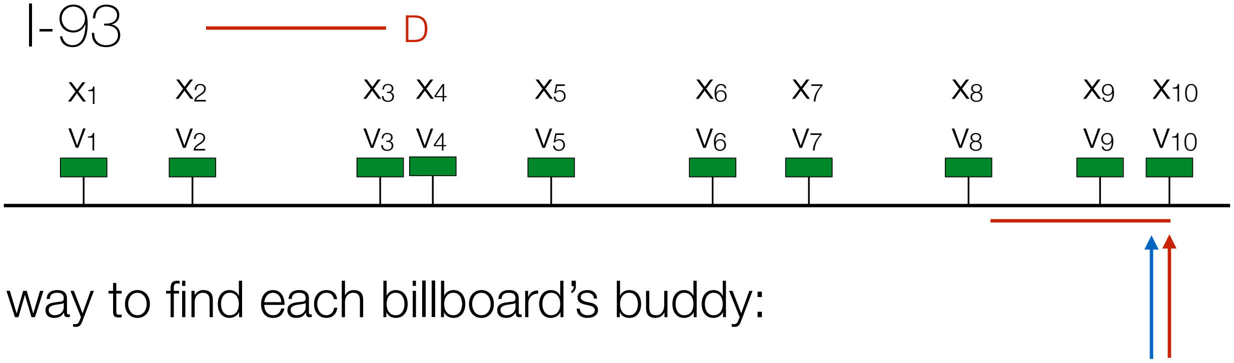
$$\mathrm{BEST}_j = \max \begin{cases} \mathrm{BEST}_{j-1} \\ v_j + \mathrm{BEST}_{cl(j)} \end{cases}$$

```
best[0] = 0
for i=1 to n
    cl = i-1
    while( (x[i]-x[cl])< D  && cl>0) cl=cl-1
    best[i] = max(best[i-1], $v_i$+best[cl])
return best[n]
```

This line can take $\Theta(i)$ steps in the worst case.

How can we improve?

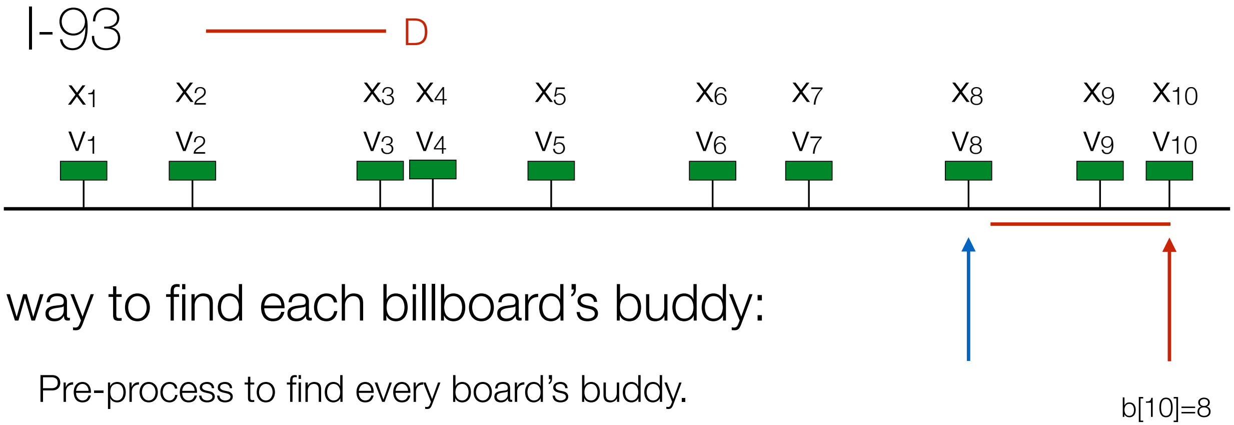
Running time (worst case): $\Theta(n^2)$


I-93
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}

Faster way to find each billboard's buddy:

Pre-process to find every board's buddy.

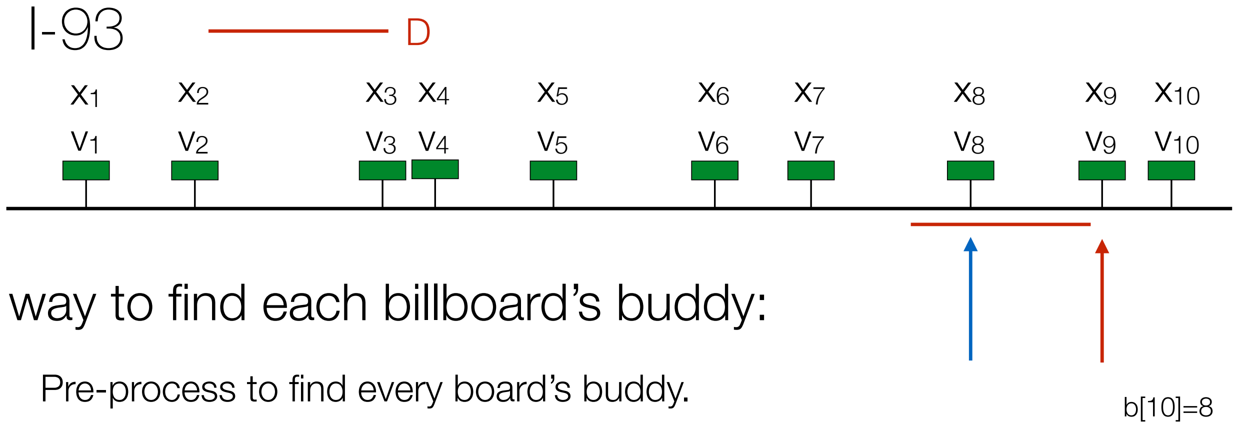
right = n, left = n


I-93
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

Faster way to find each billboard’s buddy:

Pre-process to find every board’s buddy.

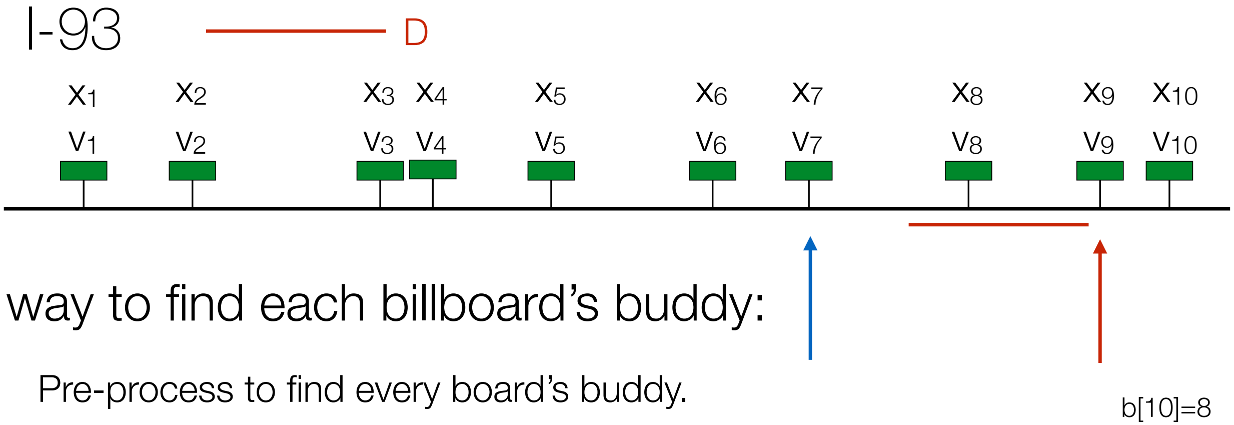
```
right = n, left = n


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


I-93
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

Faster way to find each billboard's buddy:

Pre-process to find every board's buddy.

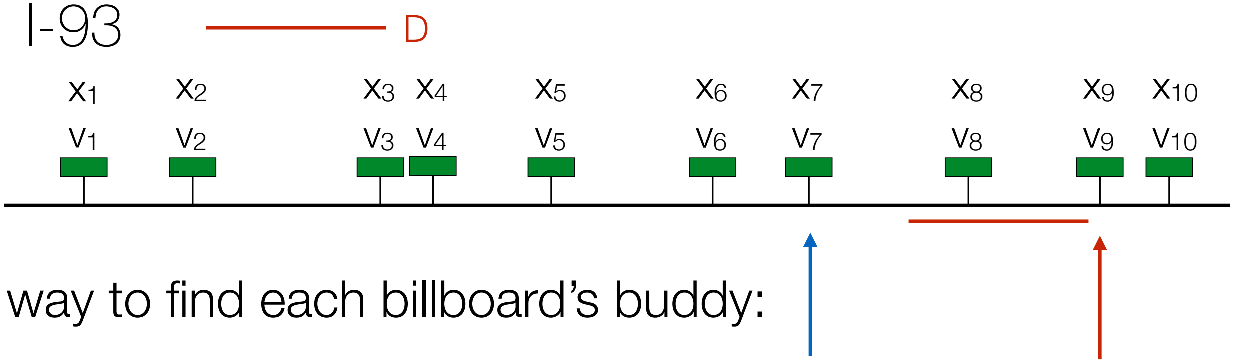
```
right = n, left = n


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



Faster way to find each billboard’s buddy:

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
```


I-93
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}

Faster way to find each billboard's buddy:

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

handle all of the remaining buddies for right
```

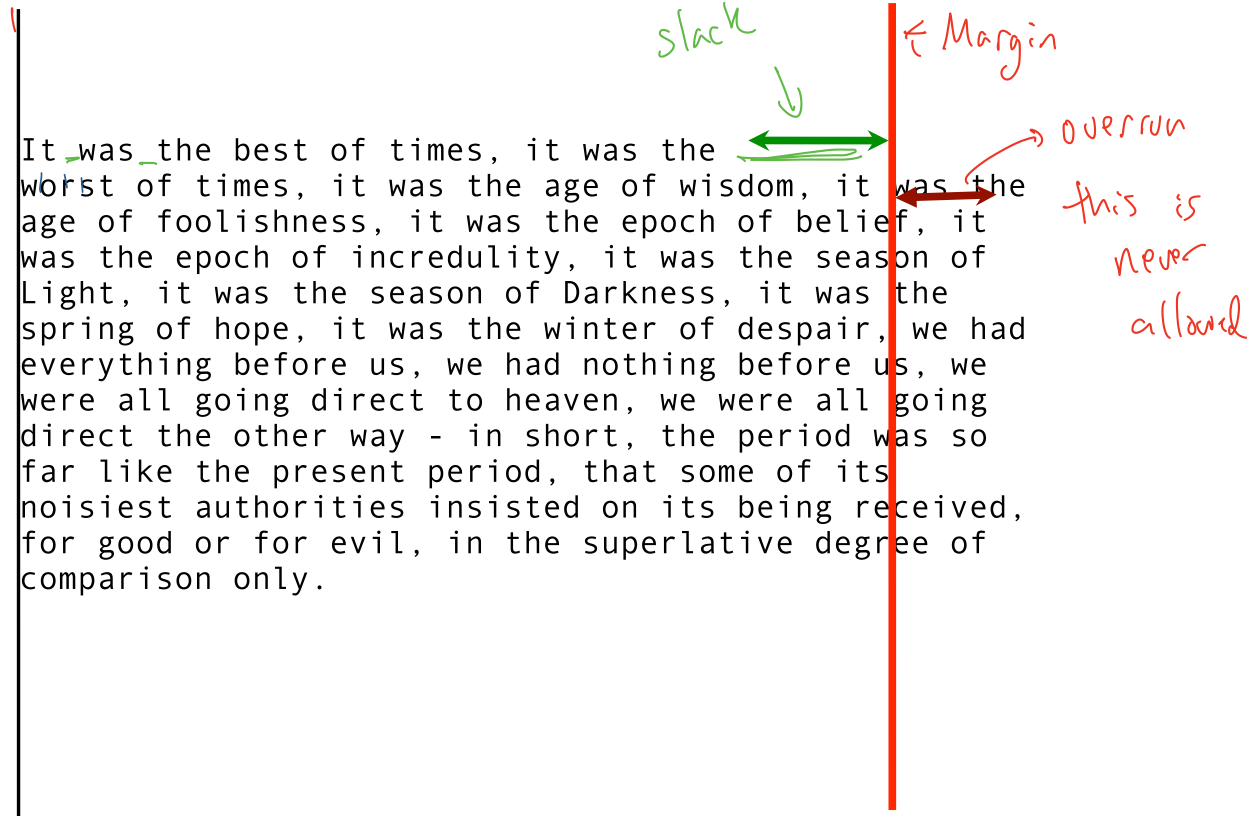
Better Billboard

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

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

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.



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 alg fails

```
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.
```

the score for a paragraph is the

________ is

sum of slack^2.

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.

	$(\text{slack})^2$
0	0
0	0
2	4
12	144
2	4
1	1
6	36
2	4
2	4
	197

__________ is

Penalty is the square of the total slack.

```
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
the other way - in short, the period was
so far like the present period, that some        197
of its noisiest authorities insisted on
its being received, for good or for evil,
in the superlative degree of comparison only
```

Greedy fails: The first two lines are perfect, but the 4th line has large slack.

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
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		
		123

A better solution evens out the slack between the first and 4th line.

Typesetting problem

input: list of words $W = (w_1, w_2, w_3 \dots w_n)$, Margin M

length of the

output: list of words on each line

such that produce No overruns into the margin

and minimize the $\sum(\text{slack})^2$.

Typesetting problem

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

output: $L = (w_1, \ldots, w_{\ell_1 - 1}), (w_{\ell_1}, \ldots, w_{\ell_2 - 1}), (w_{\ell_2}, \ldots, w_{\ell_3 - 1}), \ldots (w_{\ell_k}, \ldots, w_n)$

such that

Typesetting problem

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

Length of each word

output: $L = (w_1, \ldots, w_{\ell_1 - 1}), (\overset{6}{w}_{\ell_1}, \ldots, w_{\ell_2 - 1}), (\overset{10}{w}_{\ell_2}, \ldots, w_{\ell_3 - 1}), \ldots (w_{\ell_k}, \ldots, w_n)$

First words of each line

such that

$$c_i = \left(\sum_{j=\ell_i}^{\ell_{i+1}-1} w_j \right) + (\ell_{i+1} - \ell_i - 1)$$

Chars on each line

Space between the words

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

No line over margin

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

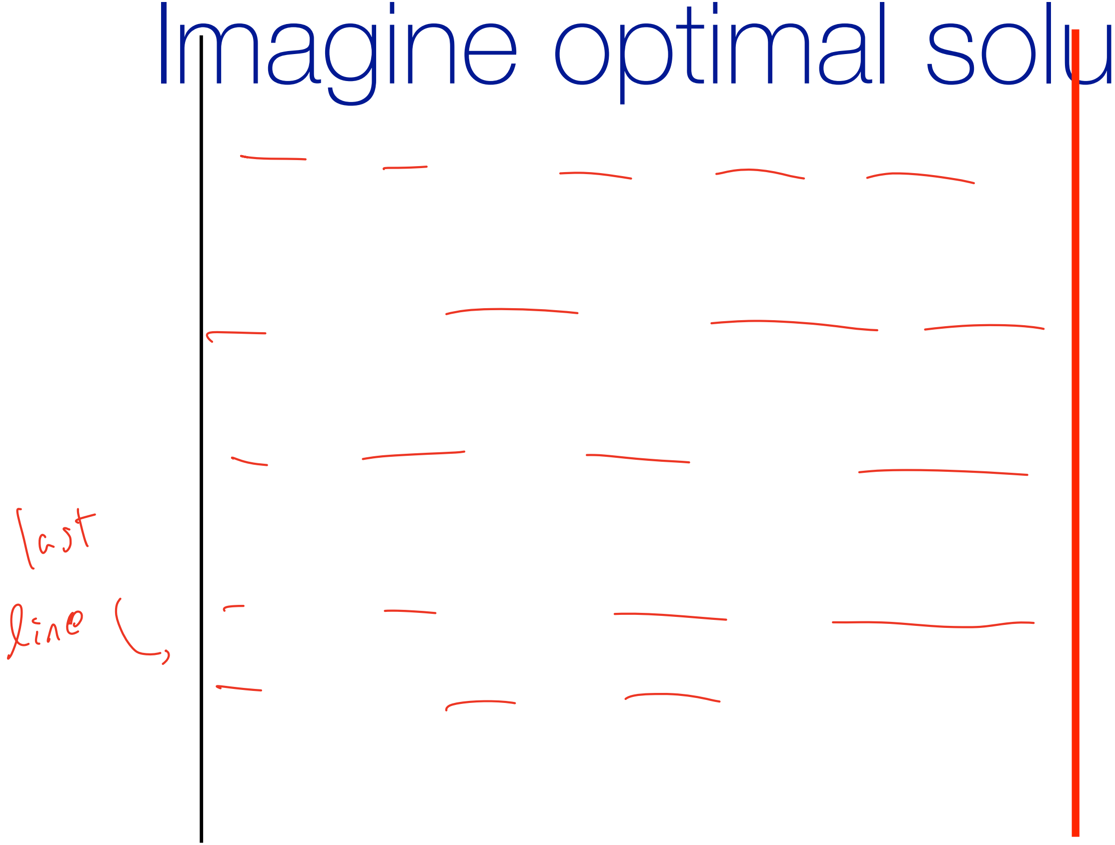
Minimize the slack²

how to solve

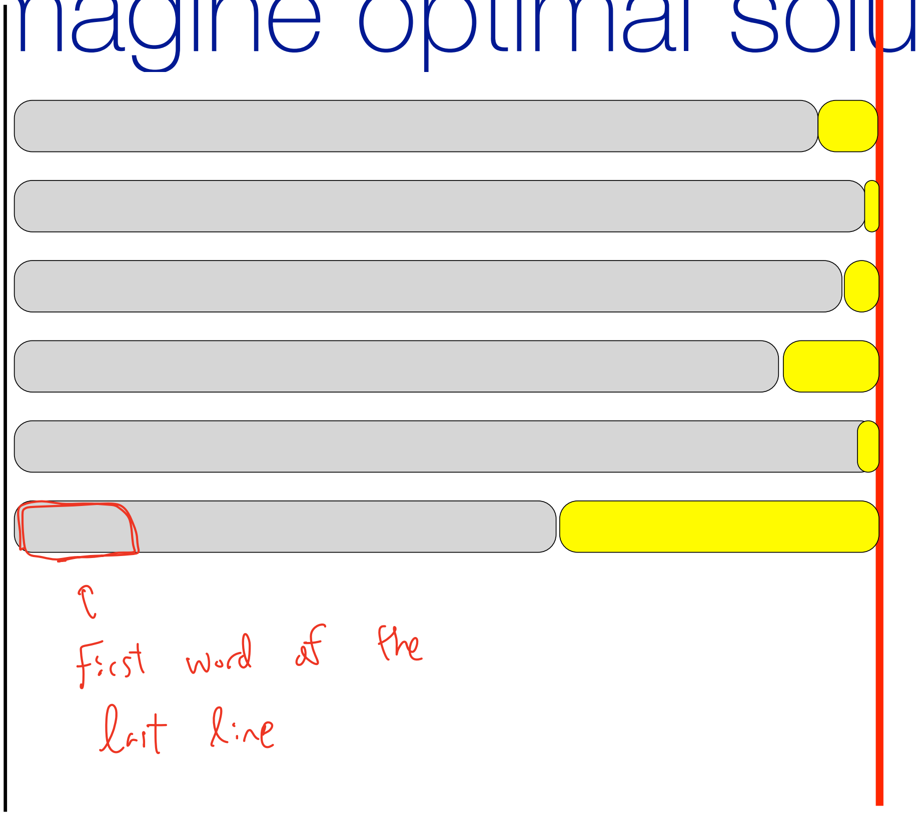
define the right variable:

Best_n: minimum penalty (sum of slack^2) for typesetting the first n words of the input.

Imagine optimal solution



Imagine optimal solution

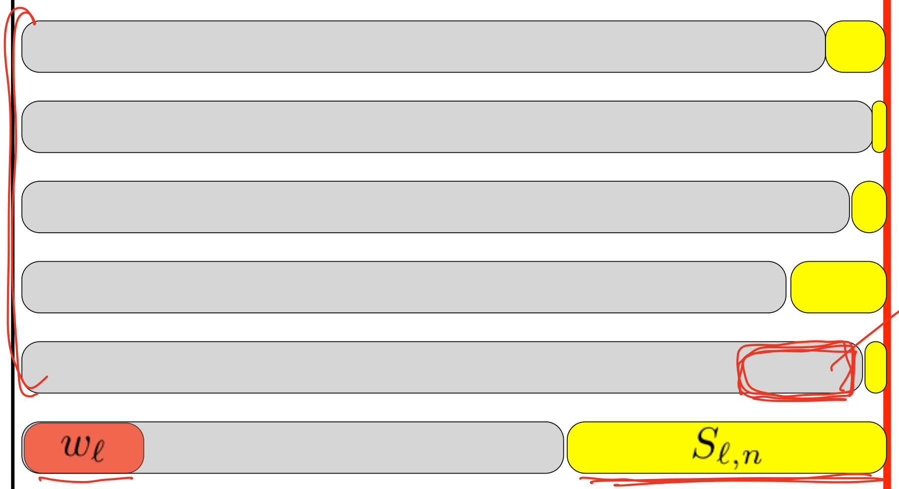


First word of the last line

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

Imagine optimal solution

min penalty if this



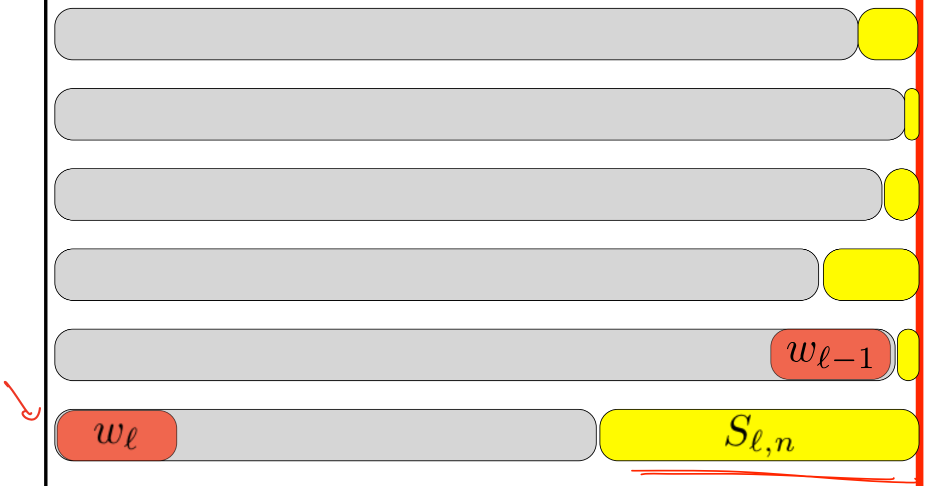
last line

fwoll is w_ℓ

slack when line starts with w_ℓ ends with w_n.

$$\text{Best}_n = \text{Best}_{w_{l-1}} + (S_{l,n})^2$$

Imagine optimal solution



last line

fwoll is w_ℓ

slack when line starts with w_ℓ

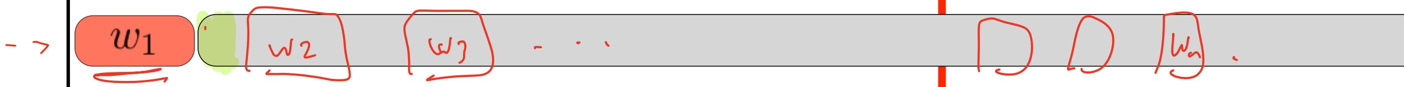
$$\underline{\mathrm{BEST}_n} = \underline{\mathrm{BEST}_{\ell-1}} + \underline{S^2_{\ell,n}}$$

How many candidates are there for the fwoll?

n

$1, 2, 3 \ldots n$

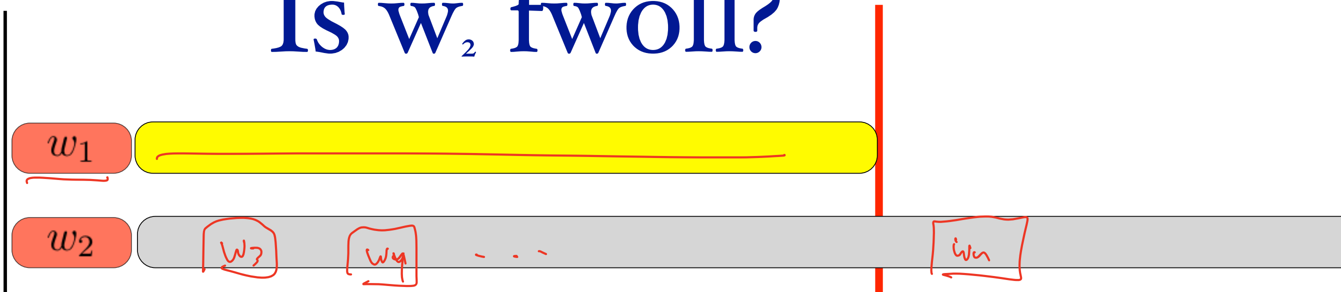
Is w_1 fwoll?



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

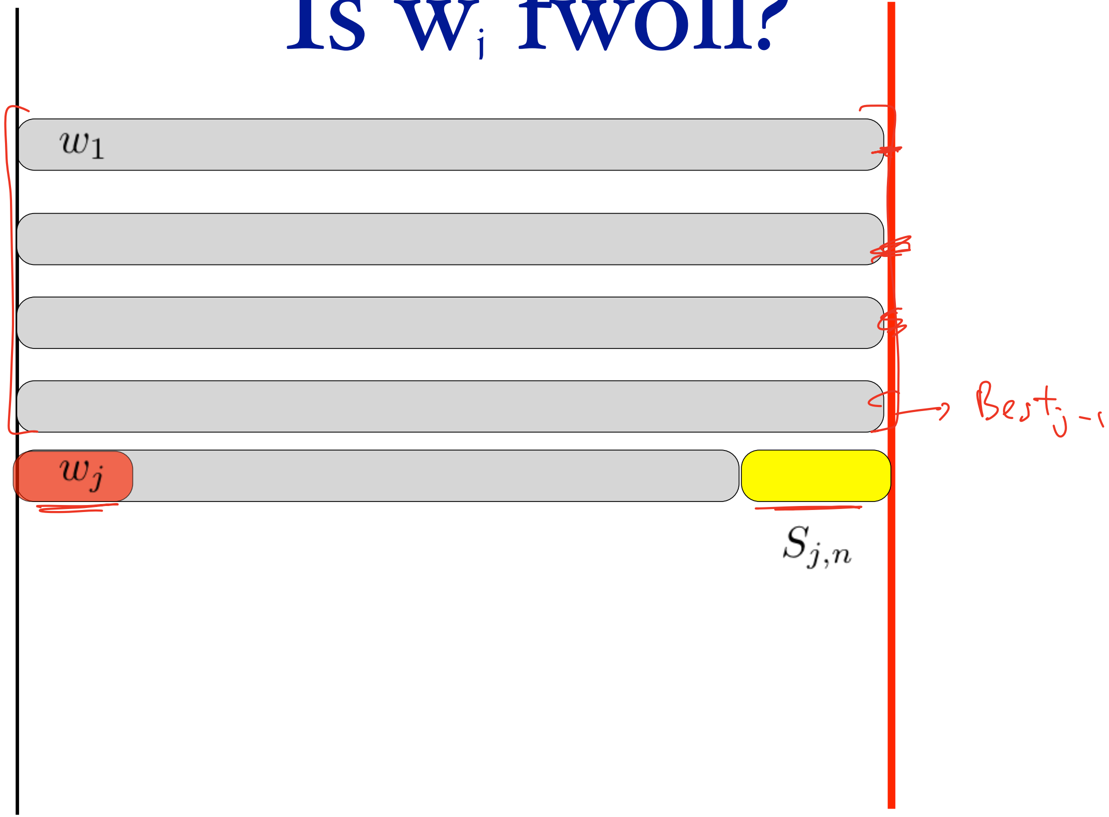
define $S_{1,n} = \infty$ if this happens

Is w_2 fwoll?



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

Is w$_j$ fwoll?



Which word is fwoll?

this equation governs the solution.

$$\underline{\underline{\mathrm{BEST}_n}} = \underline{\min} \left\{ \begin{array}{l} Best_0 + (S_{1,n})^2 \\ Best_1 + (S_{2,n})^2 \\ Best_2 + (S_{3,n})^2 \\ \vdots \\ Best_{n-1} + (S_{n,n})^2 \end{array} \right.$$

$$\sum_{i=0}^{n}$$

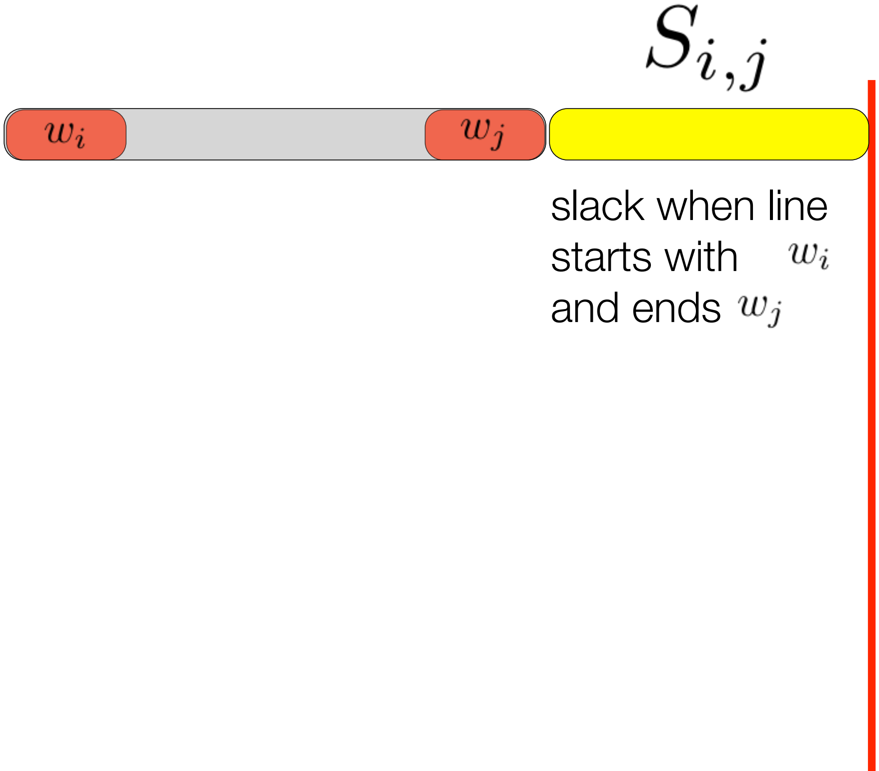
alternatively:

$$Best_n = \underline{\underline{\min_{k=0}^{n-1}}} \left\{ \underline{Best_k} + (\underset{\uparrow}{S_{k+1,n}})^2 \right\}$$

Which word is fwoll?

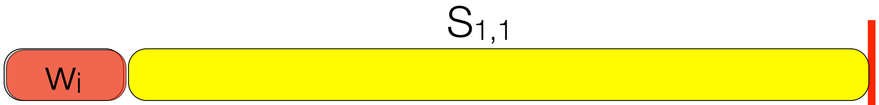
$$\mathrm{BEST}_n = \min \begin{cases} \mathrm{BEST}_0 + S_{1,n}^2 \\ \mathrm{BEST}_1 + S_{2,n}^2 \\ \mathrm{BEST}_2 + S_{3,n}^2 \\ \quad \dots \\ \mathrm{BEST}_{\ell-1} + S_{\ell,n}^2 \\ \\ \quad \dots \\ \\ \mathrm{BEST}_{n-1} + S_{n,n}^2 \end{cases}$$

How to compute $S_{i,j}$



slack when line starts with w_i and ends w_j

Simplest case

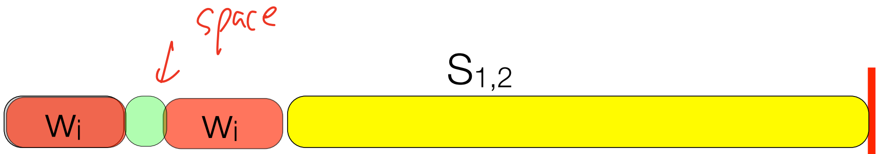


slack when line
starts with w_i
and ends w_i

$$S_{1,1} = M - w_1$$

$$S_{j,j} = M - w_j$$

Simplest case



slack when line
starts with w_i
and ends w_2

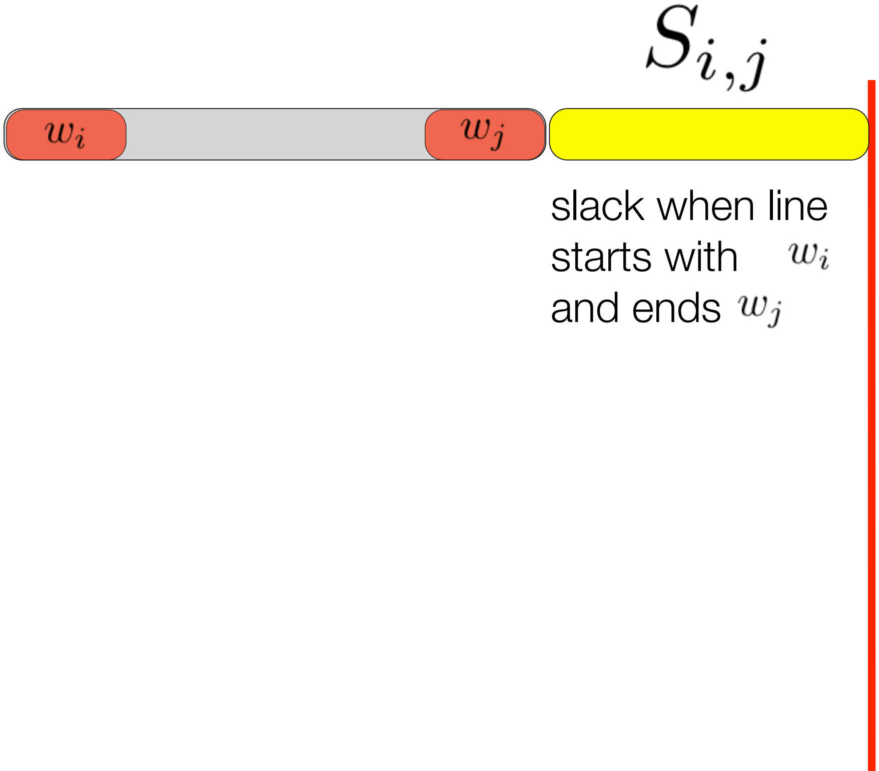
$$S_{1,2} = \underbrace{M - w_1}_{} - 1 - w_2$$

$$= S_{1,1} - w_2 - 1$$

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

true as long
as
$S_{i,j}$ is
positive.

how to compute $S_{i,j}$



slack when line starts with w_i and ends w_j

	$S_{1,1}$					$S_{1,n}$
	$M - w_1$	$S_{1,1} - w_2 - 1$	$S_{1,2} - w_3 - 1$		.....	
	$S_{2,2}$	$M - w_2$			.....	
i		$S_{3,3}$	$M - w_3$		.....	
			$S_{4,4}$	$M - w_4$	.....	

```
int infty = M*M*2;

// 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; }
        }
    }
}
```

once the line crosses the margin, then all slack values become ∞.

Typesetting algorithm

1 make table for $S_{i,j}$

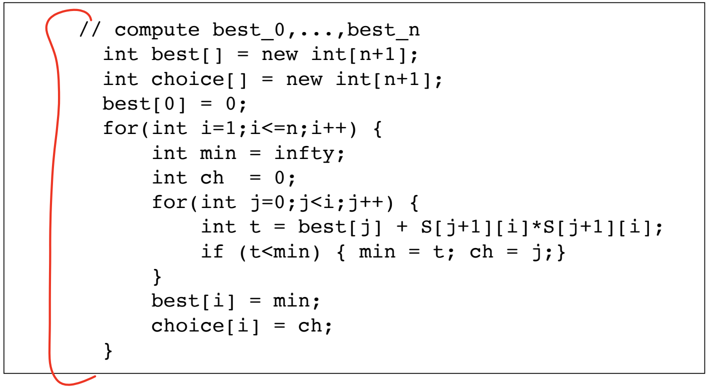
```
for $i = 1 \dots n$
    compute $Best_i$ using the equation
        $Best_i = \min_{j=0}^{i-1} \left\{ Best_j + (S_{j+1,i})^2 \right.$
```

Typesetting algorithm

make table for $S_{i,j}$

for `i=1` to `n`

best[`i`] = min{ best[`j`] + s[`j+1`][`i`]^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;
 }
```

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
```

length of words of our input

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

$S_{1,\dots}$

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

$S_{1,1} = 42 - 2$. $S_{1,2} = 40 - 3 - 1$ $S_{1,3} = 36 - 3 - 1$

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

$M = 42$

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

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

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	∞	∞
3			39	34	31	24	...						

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

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

$S_{2,3} = 39 - 3 - 1$

$S_{2,4} = 35 - 4 - 1$

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

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

$M = 42$

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
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

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

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

second step: compute

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

$$\text{Best}_1 = \min_{0}^{0} \left\{ \text{Best}_0 + (S_{1,1})^2 = 0 + 40^2 = 1600 \right.$$

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

$S_{i..j}$:

	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

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
best	0	1600												
choice	0	0	0	0	0	0	0	0	0	0	0			

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

It

$$\text{Best}_2 = \min_{0}^{1}\begin{cases}\text{Best}_0 + (S_{1,2})^2 = 0 + 36^2 = 1296 \\ \text{Best}_1 + (S_{2,2})^2 = 1600 + 39^2 \Rightarrow 2000\end{cases}$$

$S_{i,j}$	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

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
best	0	1600	1296											
choice	0	0	0	0	0	0	0	0	0	0	0			

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

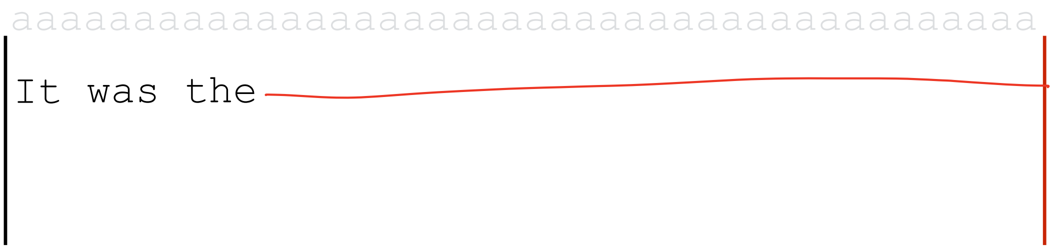
It was

$$\text{Best}_3 = \min \begin{cases} \text{Best}_0 + (S_{1,3})^2 & = 0 + 32^2 = 1024 \\ \text{Best}_1 + (S_{2,3})^2 \\ \text{Best}_2 + (S_{3,3})^2 \end{cases}$$

$S_{i,j}$	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

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
best	0	1600	1296	1024										
choice	0	0	0	0	0	0	0	0	0	0	0			



$S_{i,j}$	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

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
best	0	1600	1296	1024	729	576	289	196	100	36	0			
choice	0	0	0	0	0	0	0	0	0	0	0			

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
It was the best of times, it was the worst
```

$S_{i,j}$

	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

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
best	0	1600	1296	1024	729	576	289	196	100	36	0			
choice	0	0	0	0	0	0	0	0	0	0	0			

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

It was the best of times, it was the worst

↗ 11

of

$\mathrm{Best}_{10} + (S_{11,11})^2 = 0 + 40^2 = \underline{1600}$

Best₁₁ = min {

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
best	0	1600	1296	1024	729	576	289	196	100	36	0			
choice	0	0	0	0	0	0	0	0	0	0	0			

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

It was the best of times, it was the

worst of

$\text{best}_9 + S^2_{10,11}$

$\rightarrow 36 + (S_{10,11})^2 = 36 + (42-8)^2 = 36+1296 = \underline{1332}$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
best	0	1600	1296	1024	729	576	289	196	100	36	0			
choice	0	0	0	0	0	0	0	0	0	0	0			

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

$\text{best}_8 + S^2_{9,11}$

It was the best of times, it was
the worst of

$100 + (42-12)^2 = 100 + 900 = 1000$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
best	0	1600	1296	1024	729	576	289	196	100	36	0	818		
choice	0	0	0	0	0	0	0	0	0	0	0			

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

It was the best of times,

it was the worst of

$\text{best}_6 + S^2_{7,11}$

$289+(23)^2 = 818$

$$\underline{\text{BEST}_{11}} = \min\begin{cases}\text{BEST}_{10} + S^2_{11,11}\\ \text{BEST}_9 + S^2_{10,11}\\ \text{BEST}_8 + S^2_{9,11}\\ \text{BEST}_7 + S^2_{8,11}\\ \text{BEST}_6 + S^2_{7,11}\\ \text{Best}_5 + \ldots\end{cases}$$

This break is the best one for the first 11 words.

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
best	0	1600	1296	1024	729	576	289	196	100	36	0	818	545	
choice	0	0	0	0	0	0	0	0	0	0	0	6	6	

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

It was the best of times,

it was the worst of times, it

14

$\text{best}_6 + S^2_{7,13}$

$289 + 14^2 = 289 + 196 = 485$

$196 +$

$$\text{BEST}_{13} = \min \begin{cases} \text{BEST}_{12} + S^2_{13,13} \\ \text{BEST}_{11} + S^2_{12,13} \\ \ldots \\ \text{BEST}_7 + S^2_{8,13} \\ \text{BEST}_6 + S^2_{7,13} \end{cases}$$

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

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
best	0	1600	1296	1024	729	576	289	196	100	36	0	818	545	
choice	0	0	0	0	0	0	0	0	0	0	0	6	6	

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

$\text{best}_7 + S^2_{8,13}$

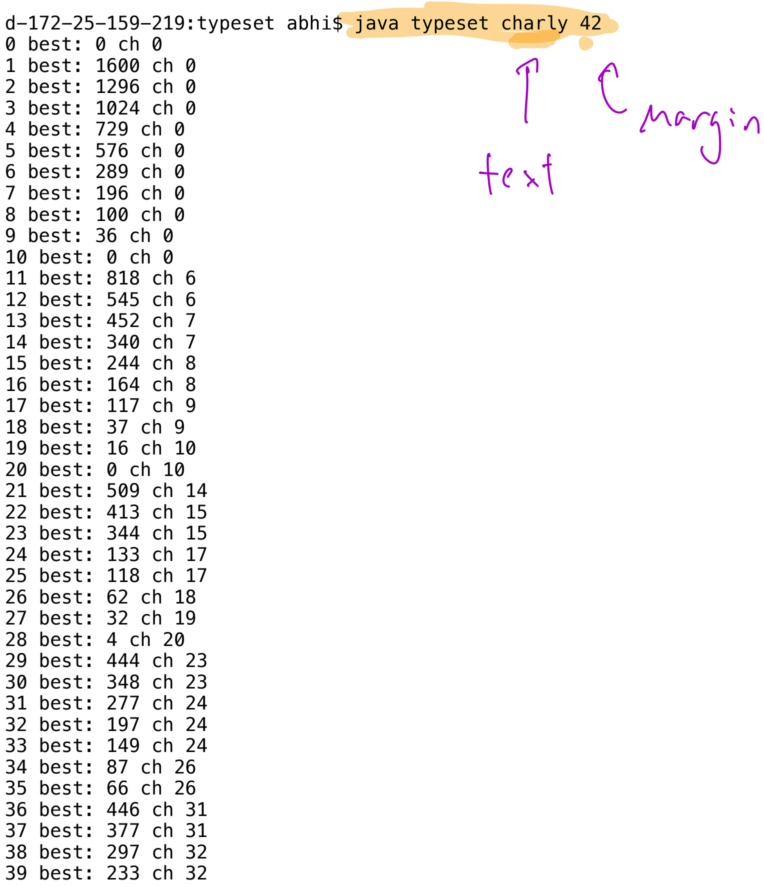
```
It was the best of times, it
was the worst of times, it
```

$\underline{16}$

$$\text{BEST}_{13} = \min \begin{cases} \text{BEST}_{12} + S^2_{13,13} \\ \text{BEST}_{11} + S^2_{12,13} \\ \ldots \\ \text{BEST}_7 + S^2_{8,13} \\ \text{BEST}_6 + S^2_{7,13} \end{cases}$$

$196 + 16^2 = 196 + 256 = \underline{\underline{452}}$

```
0 best: 0 ch 0
1 best: 1600 ch 0        It
2 best: 1296 ch 0        It was
3 best: 1024 ch 0        It was the
4 best: 729 ch 0         It was the best
5 best: 576 ch 0         It was the best of
6 best: 289 ch 0         It was the best of times,
7 best: 196 ch 0         It was the best of times, it
8 best: 100 ch 0         It was the best of times, it was
9 best: 36 ch 0          It was the best of times, it was the
10 best: 0 ch 0          It was the best of times, it was the worst
11 best: 818 ch 6        It was the best of times,\nit was the worst of
12 best: 545 ch 6        It was the best of times,\nit was the worst of times,
13 best: 452 ch 7        It was the best of times, it\nwas the worst of times, it
14 best: 340 ch 7        It was the best of times, it\nwas the worst of times, it was
15 best: 244 ch 8        It was the best of times, it was\nthe worst of times, it was the
16 best: 164 ch 8        It was the best of times, it was\nthe worst of times, it was the age
17 best: 117 ch 9        It was the best of times, it was the\nworst of times, it was the age of
18 best: 37 ch 9         It was the best of times, it was the\nworst of times, it was the age of wisdom,
19 best: 16 ch 10        It was the best of times, it was the worst\nof times, it was the age of wisdom, it
20 best: 0 ch 10         It was the best of times, it was the worst\nof times, it was the age of wisdom, it was
21 best: 509 ch 14       It was the best of times, it\nwas the worst of times, it was\nthe age of wisdom, it was the
22 best: 413 ch 15       It was the best of times, it was\nthe worst of times, it was the\nage of wisdom, it was the age
23 best: 344 ch 15       It was the best of times, it was\nthe worst of times, it was the\nage of wisdom, it was the age of
24 best: 133 ch 17       It was the best of times, it was the\nworst of times, it was the age of\nwisdom, it was the age of foolishness,
25 best: 118 ch 17       It was the best of times, it was the\nworst of times, it was the age of\nwisdom, it was the age of foolishness, it
26 best: 62 ch 18        It was the best of times, it was the\nworst of times, it was the age of wisdom,\nit was the age of foolishness, it was
```



```
d-172-25-159-219:typeset abhi$ java typeset charly 42
0 best: 0 ch 0
1 best: 1600 ch 0
2 best: 1296 ch 0
3 best: 1024 ch 0
4 best: 729 ch 0
5 best: 576 ch 0
6 best: 289 ch 0
7 best: 196 ch 0
8 best: 100 ch 0
9 best: 36 ch 0
10 best: 0 ch 0
11 best: 818 ch 6
12 best: 545 ch 6
13 best: 452 ch 7
14 best: 340 ch 7
15 best: 244 ch 8
16 best: 164 ch 8
17 best: 117 ch 9
18 best: 37 ch 9
19 best: 16 ch 10
20 best: 0 ch 10
21 best: 509 ch 14
22 best: 413 ch 15
23 best: 344 ch 15
24 best: 133 ch 17
25 best: 118 ch 17
26 best: 62 ch 18
27 best: 32 ch 19
28 best: 4 ch 20
29 best: 444 ch 23
30 best: 348 ch 23
31 best: 277 ch 24
32 best: 197 ch 24
33 best: 149 ch 24
34 best: 87 ch 26
35 best: 66 ch 26
36 best: 446 ch 31
37 best: 377 ch 31
38 best: 297 ch 32
39 best: 233 ch 32
```

text

margin

```
// read input

 try {
 BufferedReader bin = new BufferedReader(new FileReader(args[0]));
 String line = bin.readLine();
 String words[] = line.split(" ");
 int n = words.length;
 int M = Integer.parseInt(args[1]);
 int lens[] = new int[n+1];
 for(int i=1;i<=n; i++) {
     lens[i] = words[i-1].length();
     if (lens[i]>M) {
         System.out.println("word too long");
         System.exit(1);
     }
 }

 int infty = M*M*2;

 // 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; }
         }
     }
 }
```

```
// 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;
}
```

$\Theta(n^2)$

```
// backtrack to output linebreaks
int end = n;
int start   = choice[end]+1;
String lines[] = new String[n];
int cnt = 0;
while (end>0) {
    StringBuffer buf = new StringBuffer();
    for(int j=start; j<=end; j++) {
        buf.append(words[j-1] + " ");
    }
    lines[cnt++] = buf.toString();
    end = start-1;
    start = choice[end]+1;
}
```