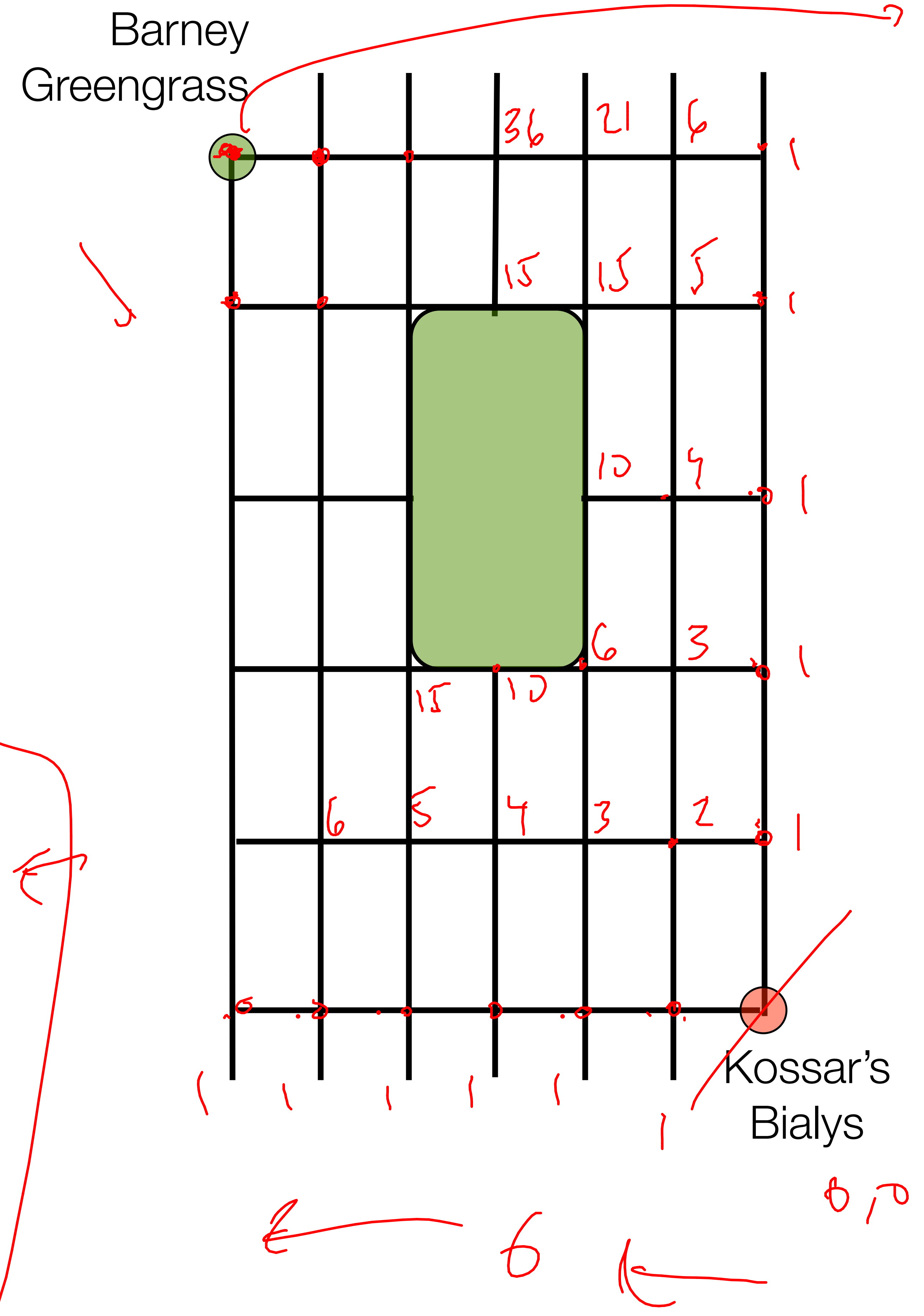
L11

10.1.2013

4102

abhi shelat

How many ways can I get from KB to BG moving only up and left?



$B(i,j)$ = # of ways to get to (i,j)

$= B(i-1,j) + B(i,j-1)$

5

11 moves, 6 of which are ←

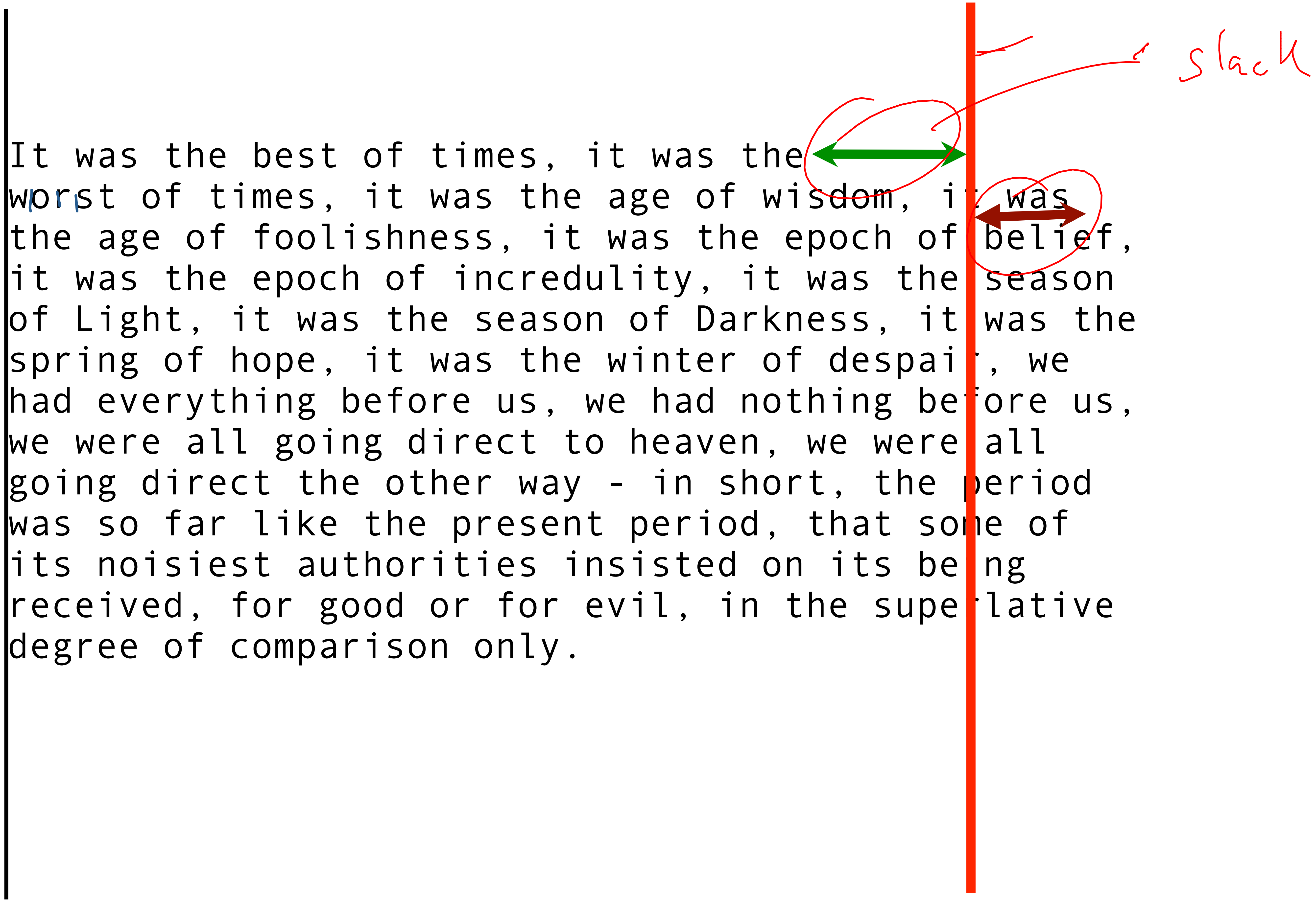
$\binom{11}{6}$

↑ if CP wasn't there.

userid:

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.

do not typeset in margin ✓

do not typeset in margin ✓

typeset every word ✓

minimize the slack^2 ✓

between margin and last word on a line

one paragraph at a time

greedy strategy fails

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 of its		197
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.
```

6	36
1	1
1	1
6	36
2	4
1	1
6	36
2	4
2	4
0	0
	123

Typesetting problem

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

output: $L = (w_1, \ldots, w_{\boxed{\ell_1}}), (w_{\ell_1+1}, \ldots, w_{\boxed{\ell_2}}), \ldots, (w_{\underline{\boxed{\ell_x}+1}}, \ldots, \underline{w_n})$ $\longleftarrow$ line breaks

such that

No line exceeds the margin

$$\underbrace{c_i}_{} < M \longrightarrow \text{characters on line } i.$$

$$\min \sum (M - c_i)^2 \longrightarrow \text{minimize the sum the (slack on each line)}^2$$

Typesetting problem

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

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

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

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

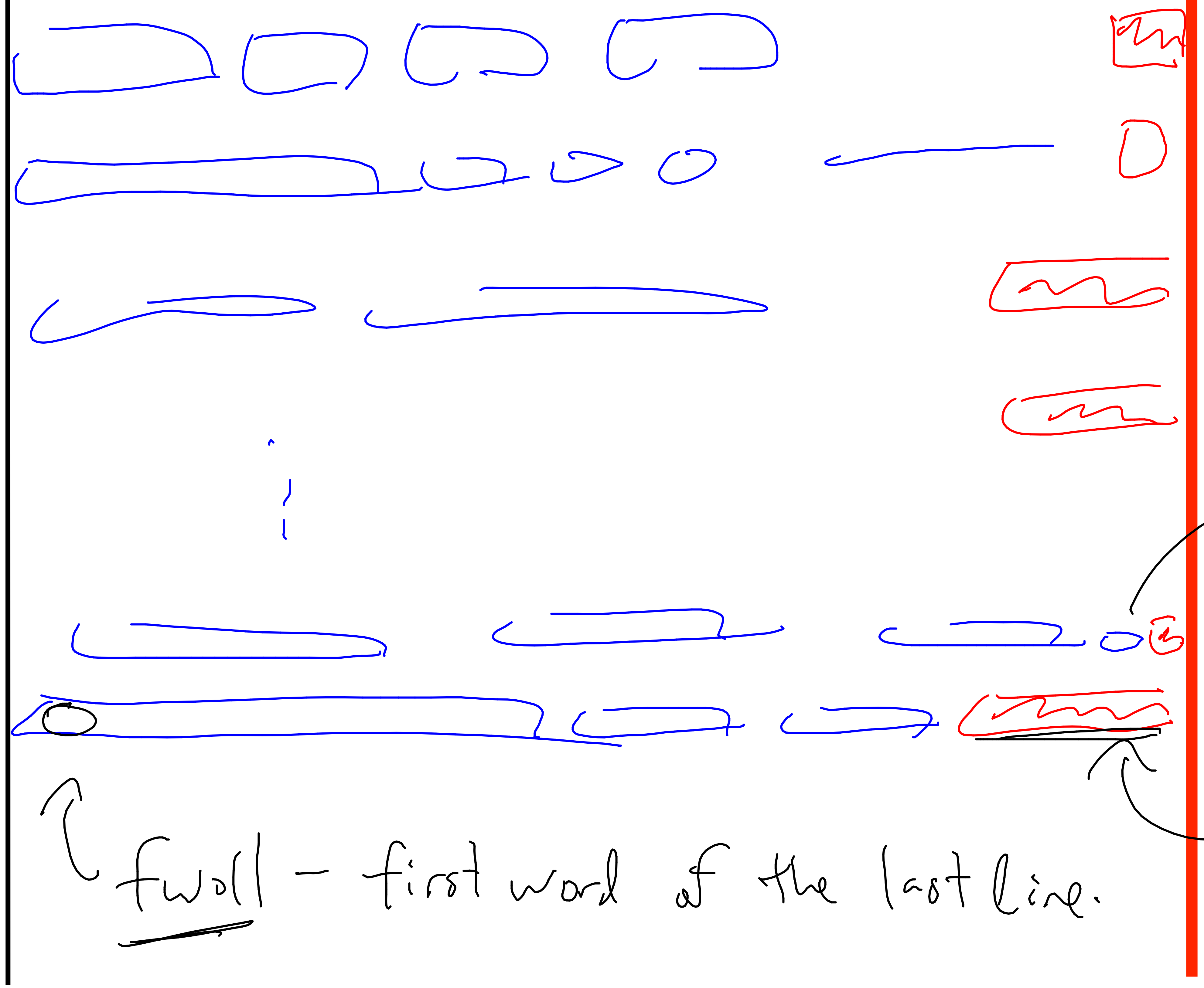
$$\underline{\min \sum (\underline{M - c_i})^2}$$

how to solve

define the right variable:

BestTn : smallest penalty for which the first n words can be typeset.

imagine optimal solution



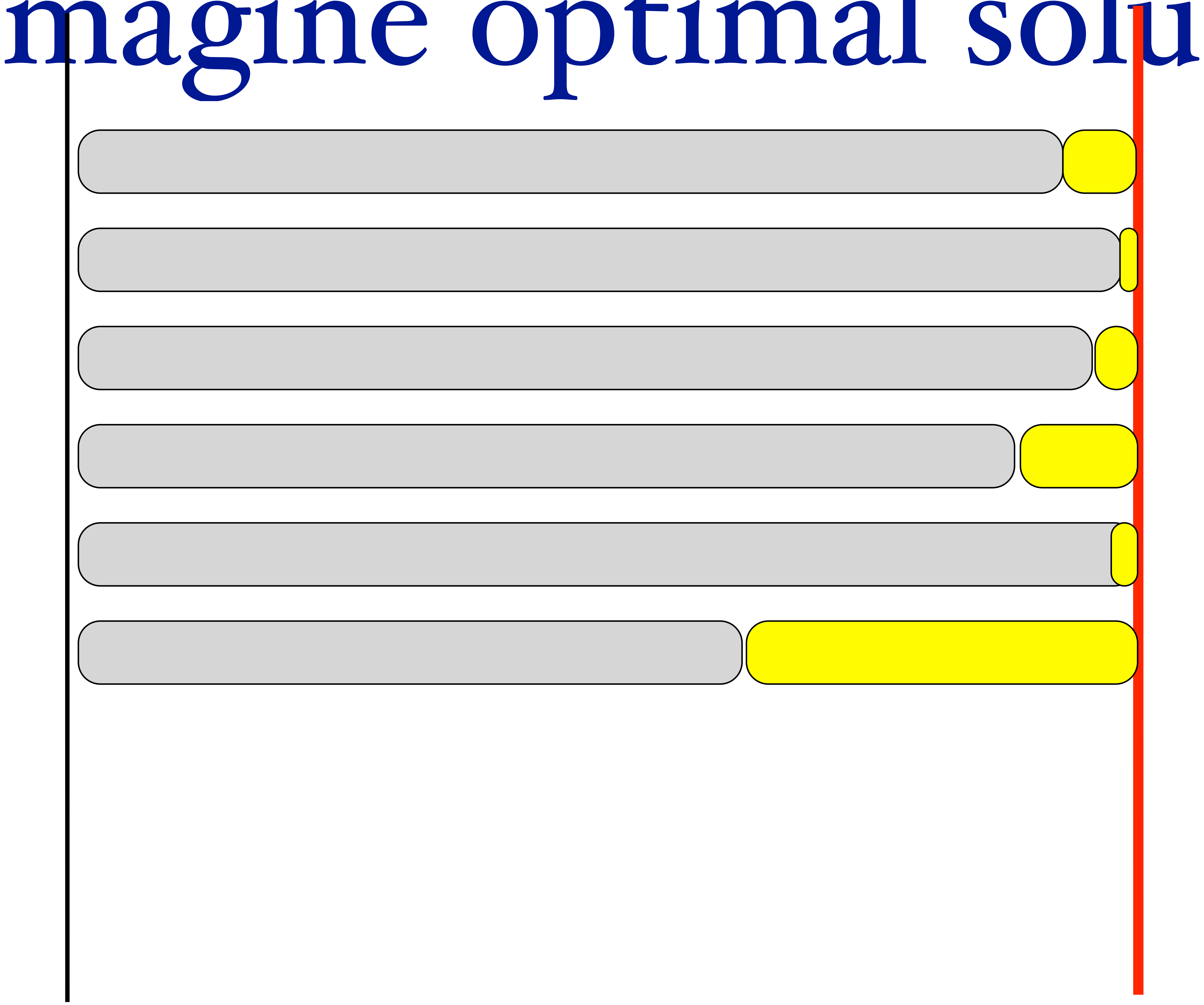
fwoll - first word of the last line.

$$Best_n = Best_{fwoll-1} + (slack_{fwoll})^2$$

fwoll-1

slack for the last line if the last line begins w/ fwoll.

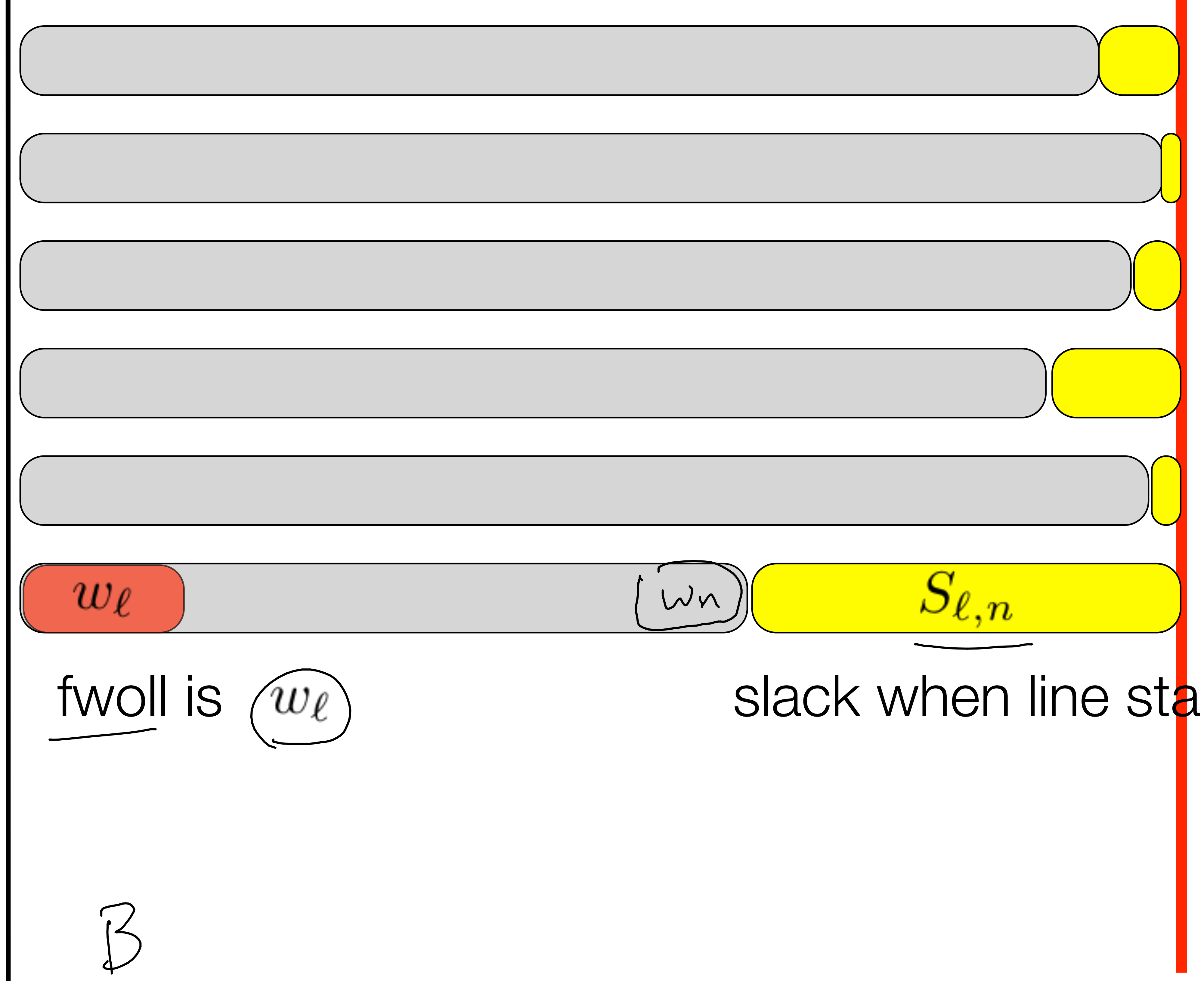
imagine optimal solution



last line

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

imagine optimal solution

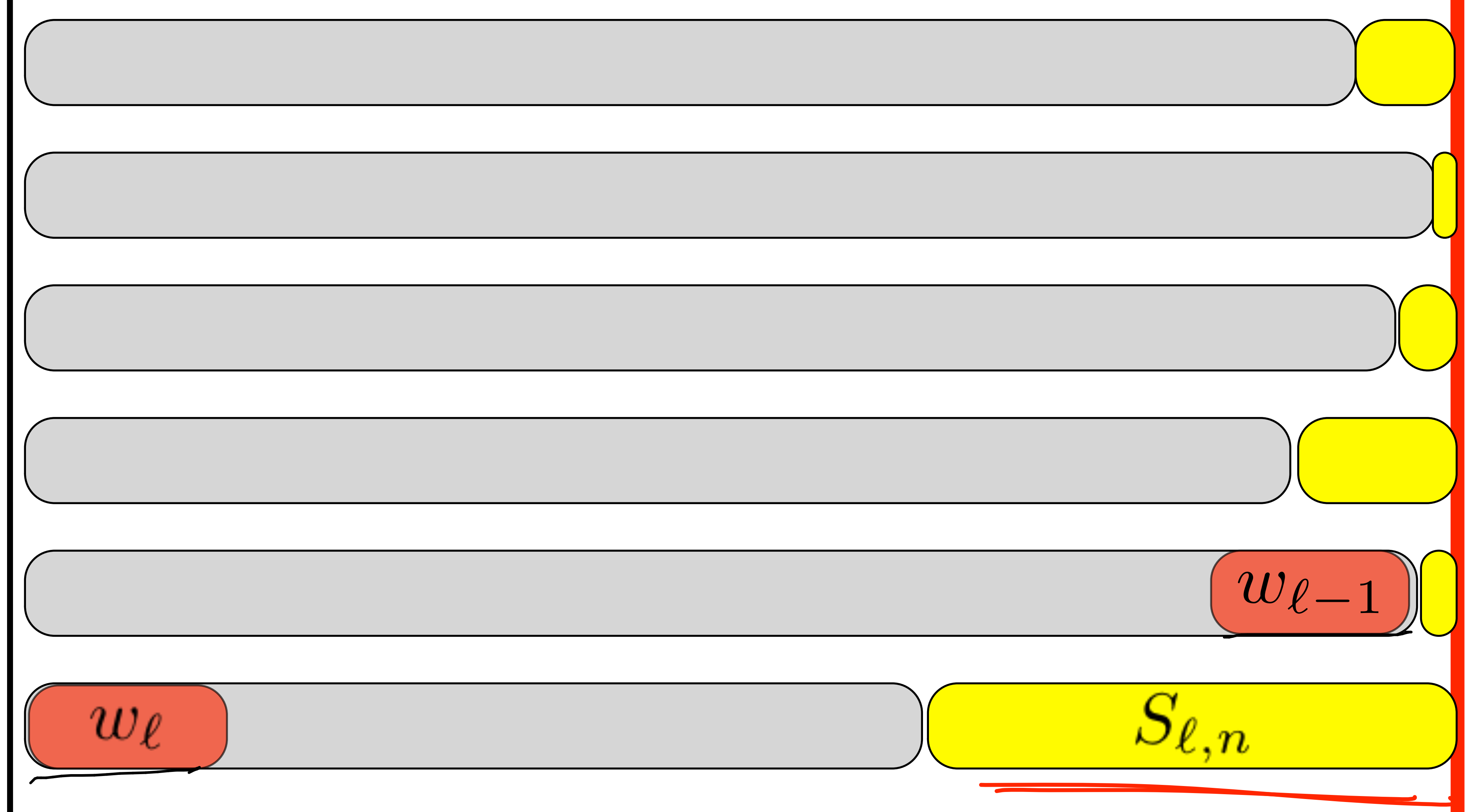


last line

fwoll is w_ℓ

slack when line starts with w_ℓ, and goes to word n.

imagine optimal solution



last line

fwoll is w_ℓ

slack when line starts with w_ℓ

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

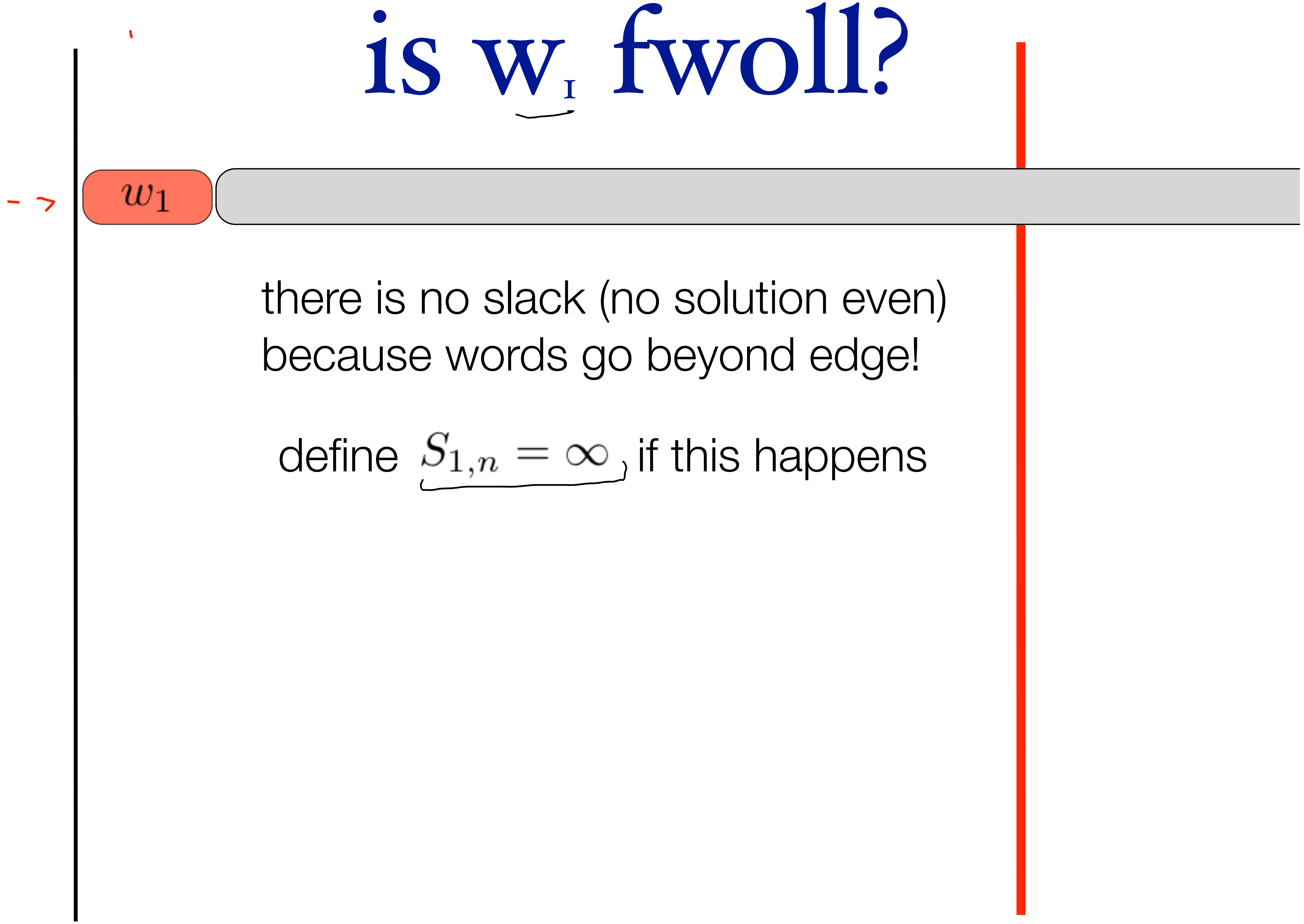
how many candidates are there for the fwoll?

n candidates.

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

slack when typesetting the line that begins with word k+1 and ends w/word n.

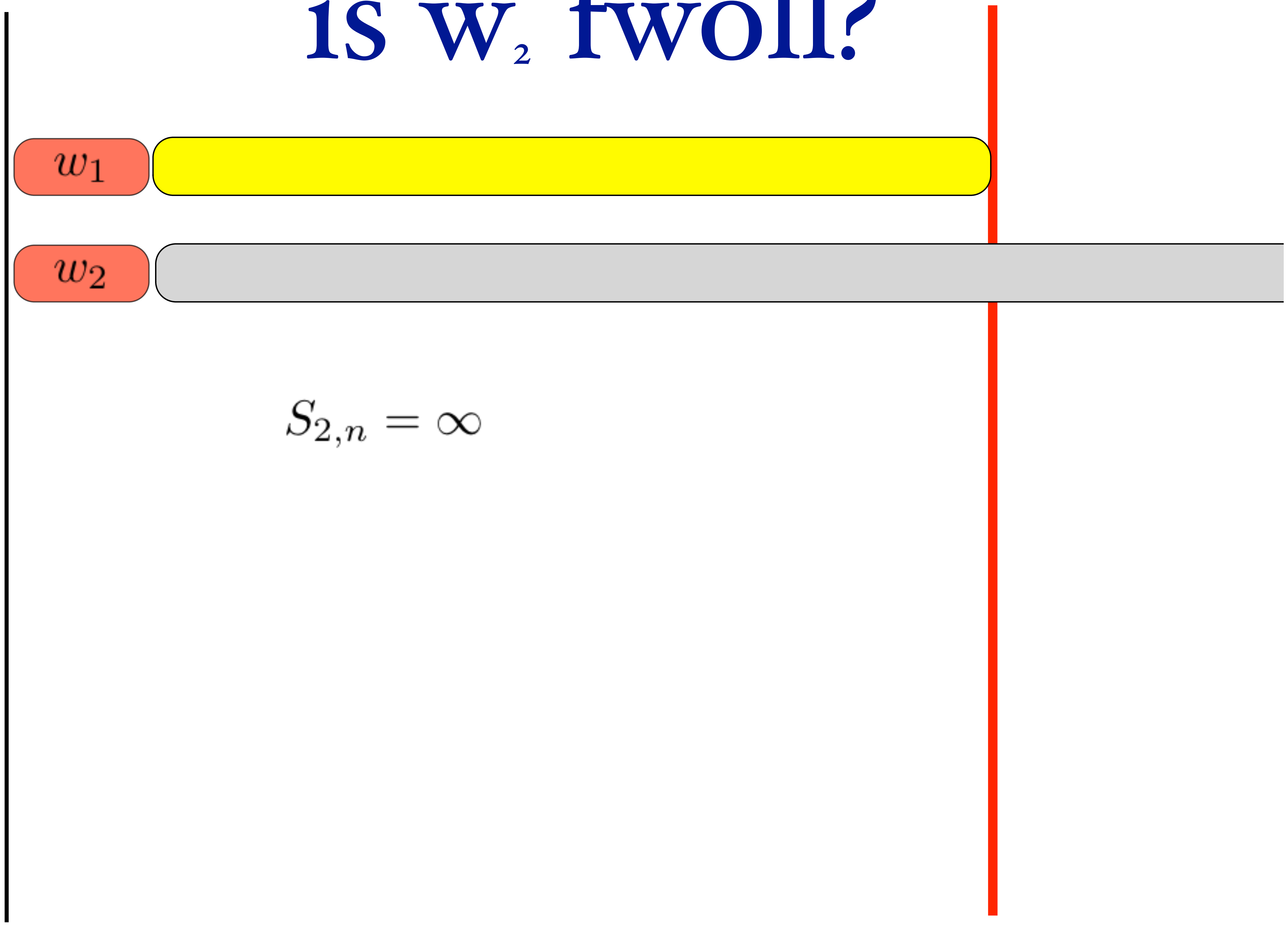
is $\underline{w_1}$ fwoll?



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

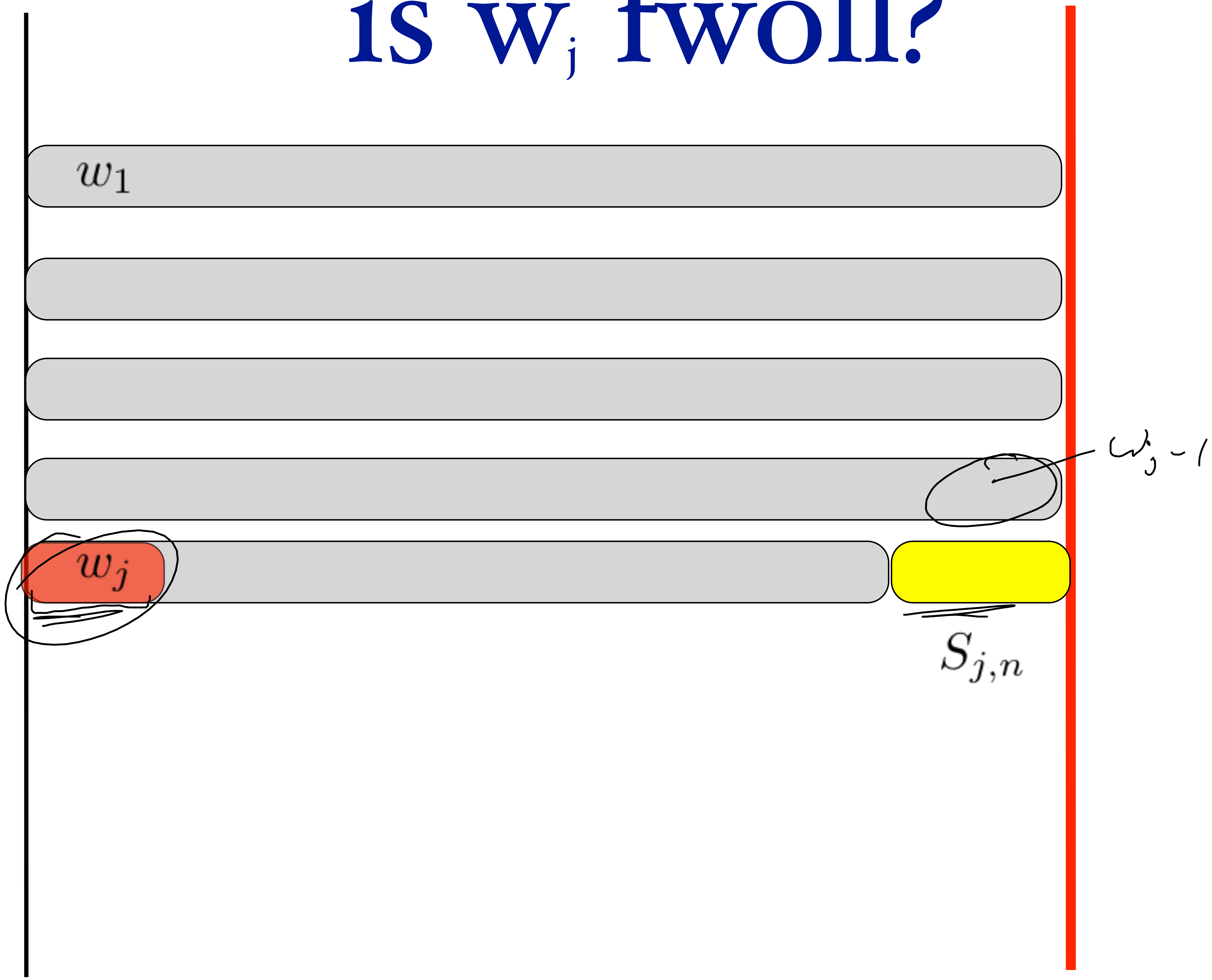
define $\underline{S_{1,n} = \infty}$, if this happens

is w_2 fwoll?



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

is w_j fwoll?



which word is fwoll?

$$\mathrm{BEST}_n = \min \left\{ \vphantom{\begin{array}{c} \\ \\ \\ \\ \end{array}} \right.$$

which word is fwoll?

$$\underline{\text{BEST}_n} = \underline{\min} \begin{cases} \overset{0}{\underline{\text{BEST}_0}} + \underline{S^2_{1,n}} \\ \underline{\text{BEST}_1} + \underline{S^2_{2,n}} \\ \text{BEST}_2 + S^2_{3,n} \\ \dots \\ \text{BEST}_{\ell-1} + S^2_{\ell,n} \\ \xrightarrow{\qquad\qquad\qquad} \text{Best}_j + \left(S_{(j+1),n}\right)^2 \\ \dots \\ \text{BEST}_{n-1} + \underline{S^2_{n,n}} \end{cases}$$

typesetting algorithm

typesetting algorithm

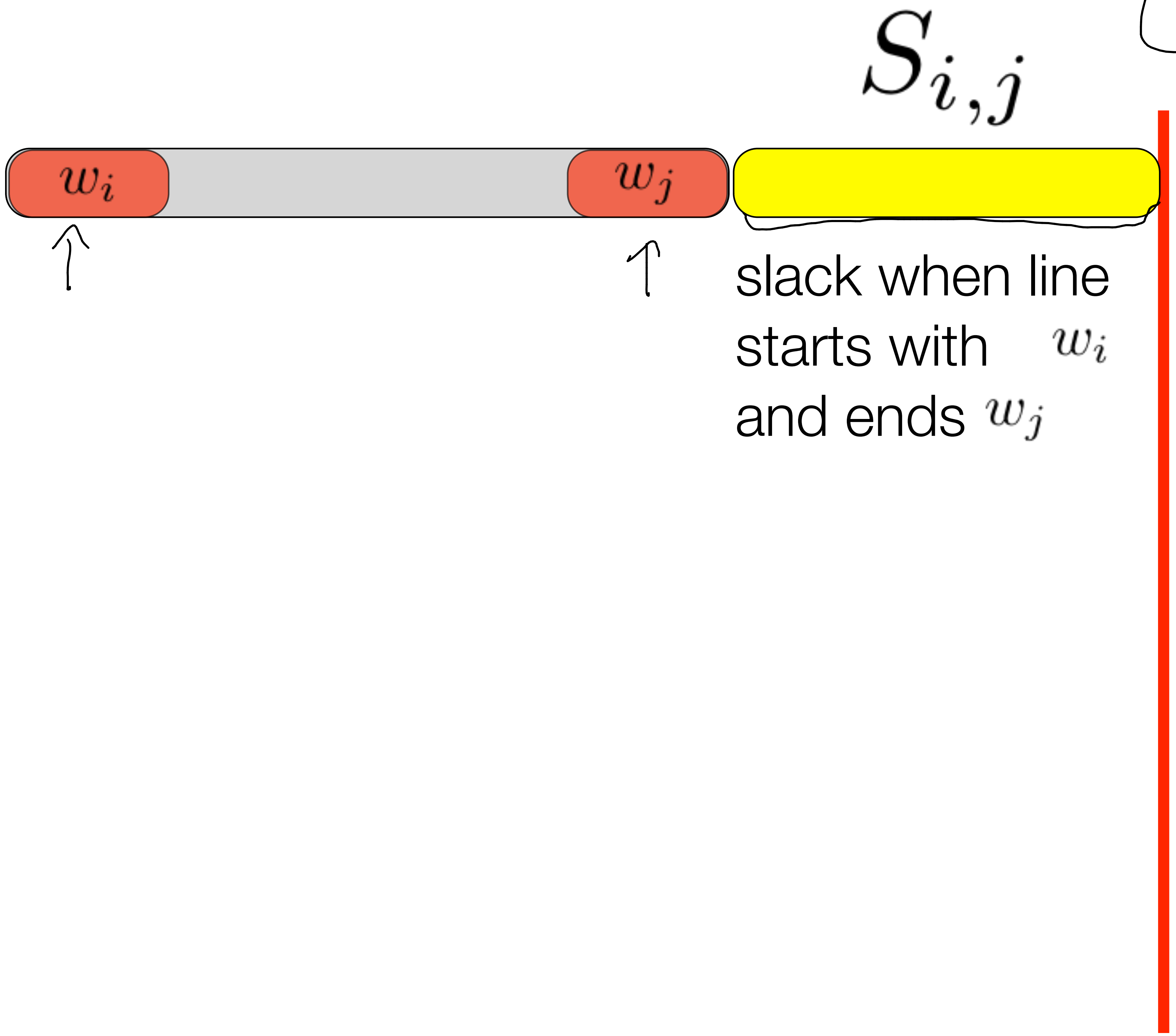
make table for $S_{i,j}$

for **i=1** to **n**

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

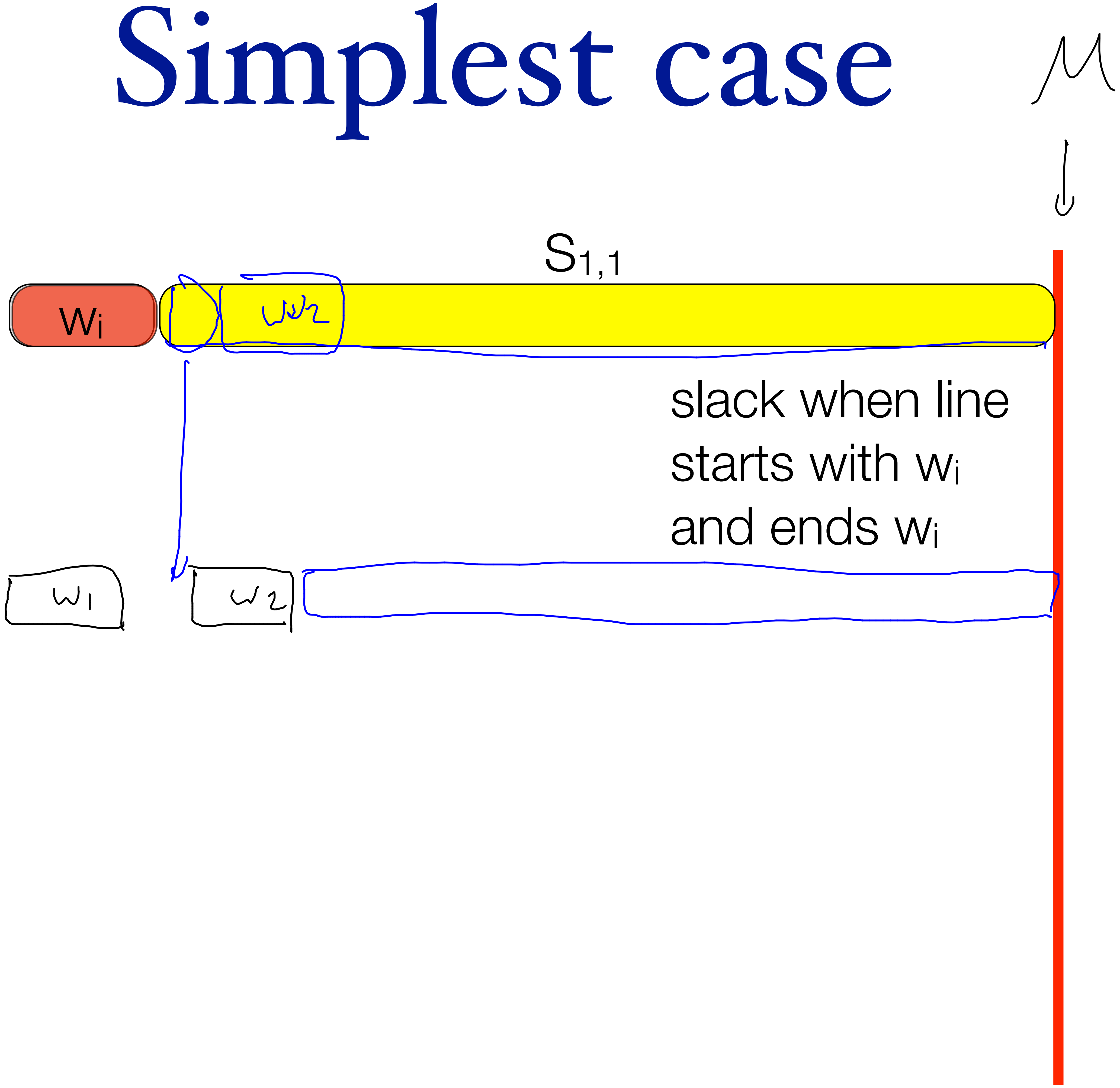
```
    // compute best_0,...,best_n
     int best[] = new int[n+1];
     int choice[] = new int[n+1];
     best[0] = 0;
     for(int i=1;i<=n;i++) {
         int min = infty;
         int ch  = 0;
         for(int j=0;j<i;j++) {
             int t = best[j] + S[j+1][i]*S[j+1][i];
             if (t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

how to compute $S_{i,j}$



slack when line starts with w_i and ends w_j

Simplest case

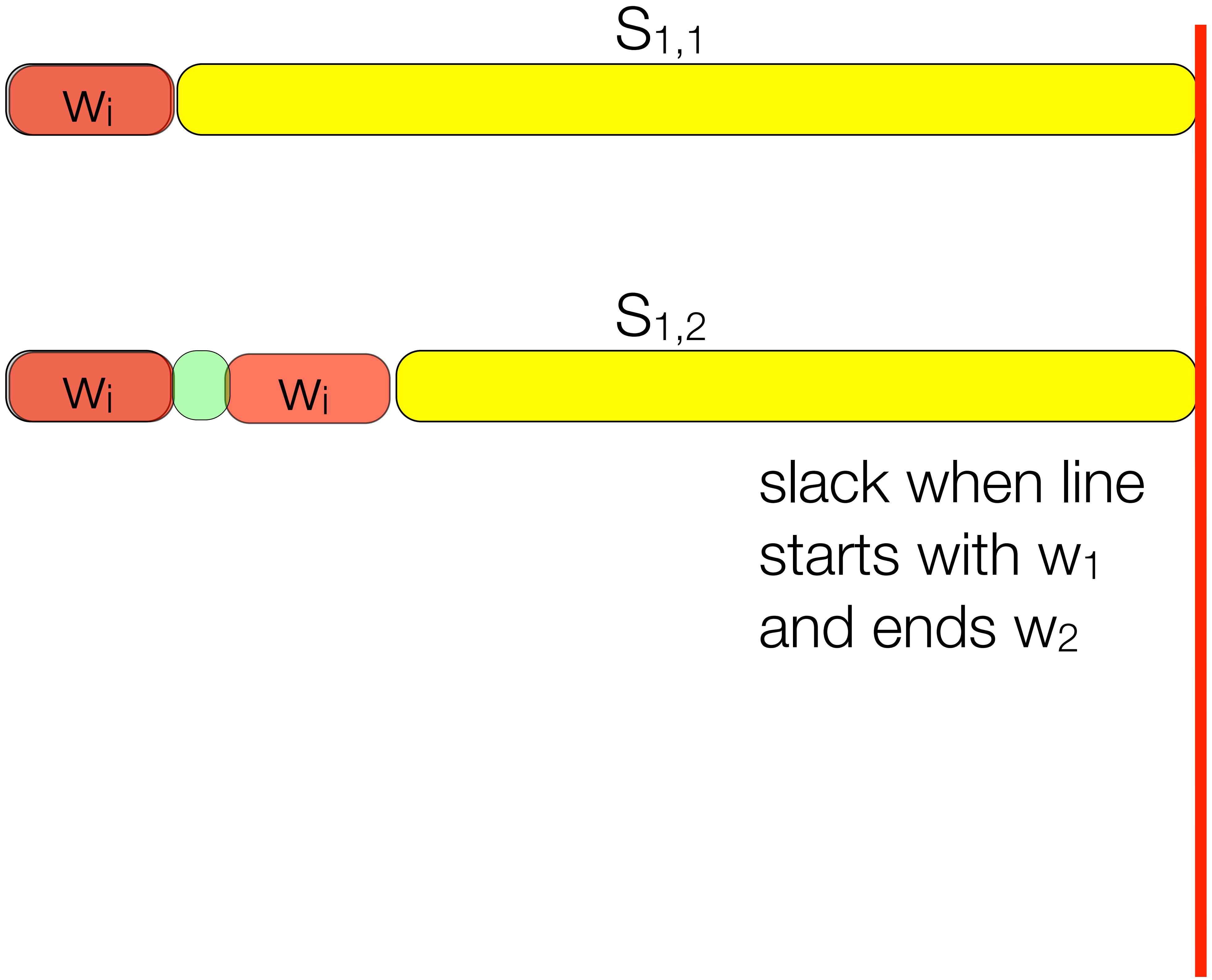


slack when line starts with w_i and ends w_i

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

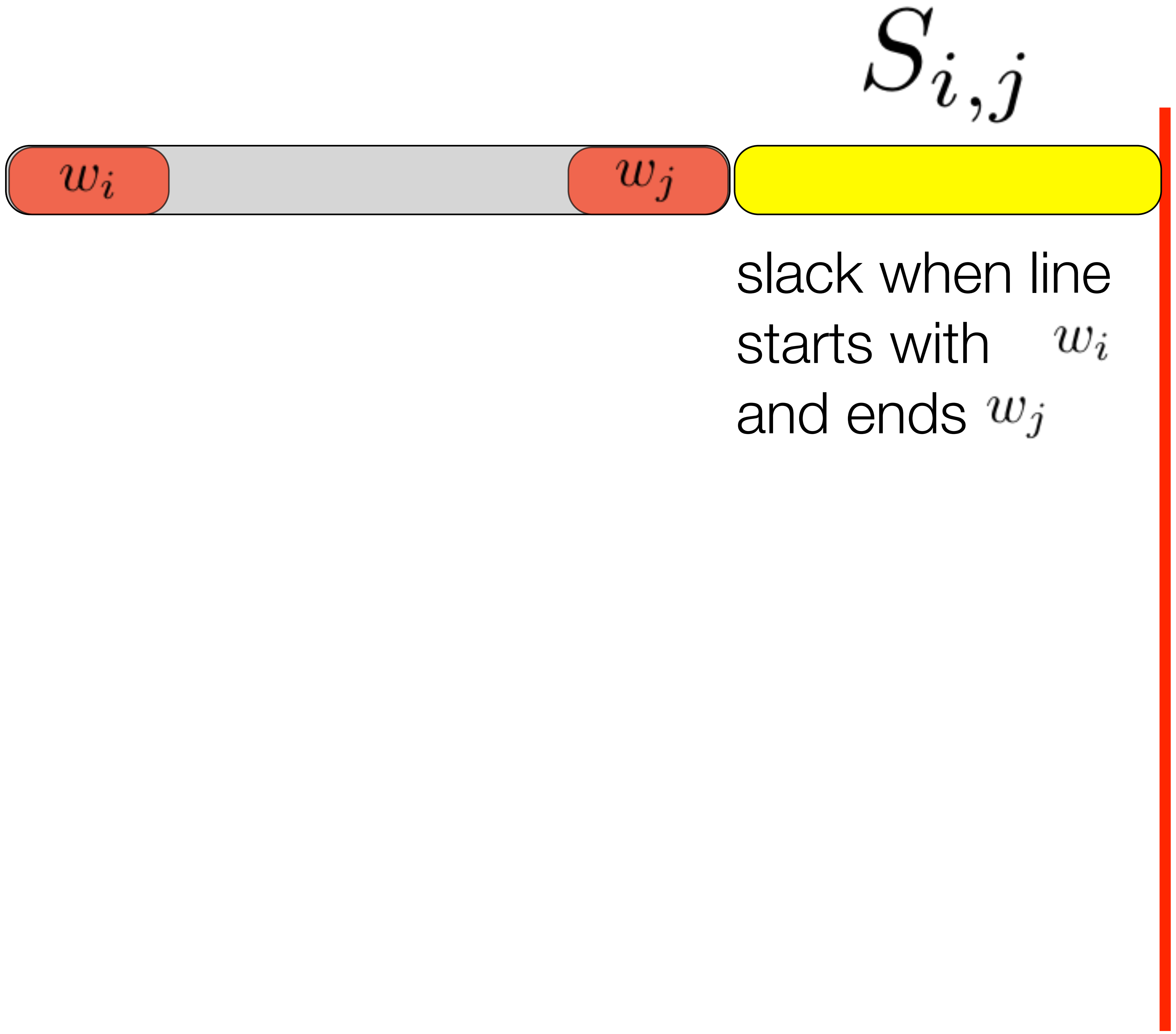
$S_{1,2} = S_{1,1} - 1 - |w_2|$

Simplest case



slack when line starts with w_1 and ends w_2

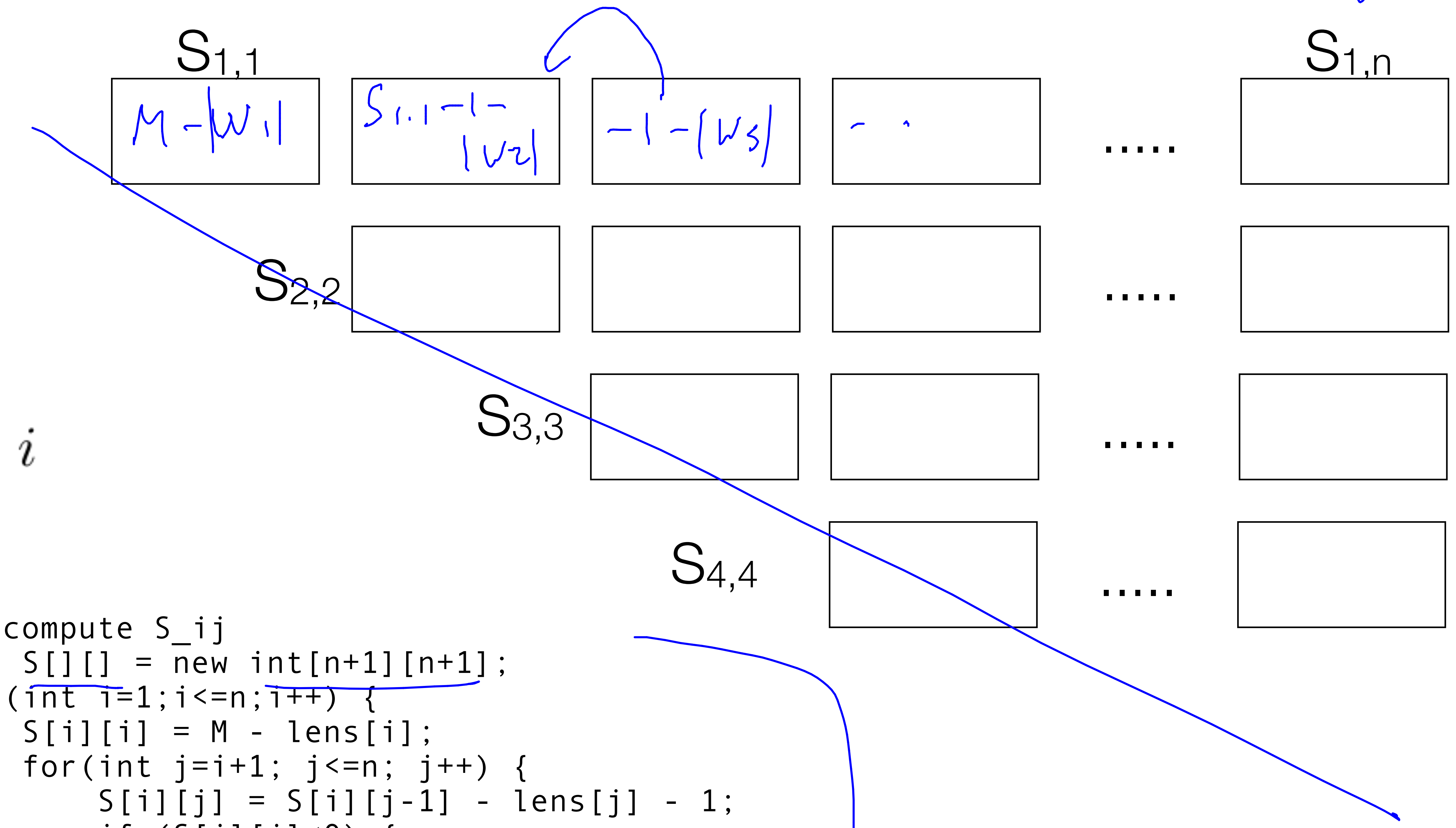
how to compute $S_{i,j}$



slack when line starts with w_i and ends w_j

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

How to compute $S_{i,j}$



```
// 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
```

```
  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
```

$S_{i,j}$ ↓

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

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	40	36											
2		39											

↑

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

$M = 42$

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

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

S_{22} → slack when typesetting word 2 ... word 2 on a line.

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											

2 $\underline{3}$ 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

$M = 42$

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

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

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

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

slack when typesetting word 2 ... 2

$S_{3,3} = M - 3 = 39$

$S_{i,j}$

second step: compute

	0	1	2	3	4	5	6	7	8	9	10	...
best	0	1600	1296	1024								

$$\mathrm{Best}_1 = \mathrm{Best}_0 + (S_{1,1})^2 = 0 + 40^2$$

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

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

	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
3			39										

$$\mathrm{Best}_3 = \min \begin{cases} 0 + 32^2 \\ 1600 + 35^2 \\ 1296 + 39^2 \end{cases} = 1024$$

it was the

it
was the

it was
the

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

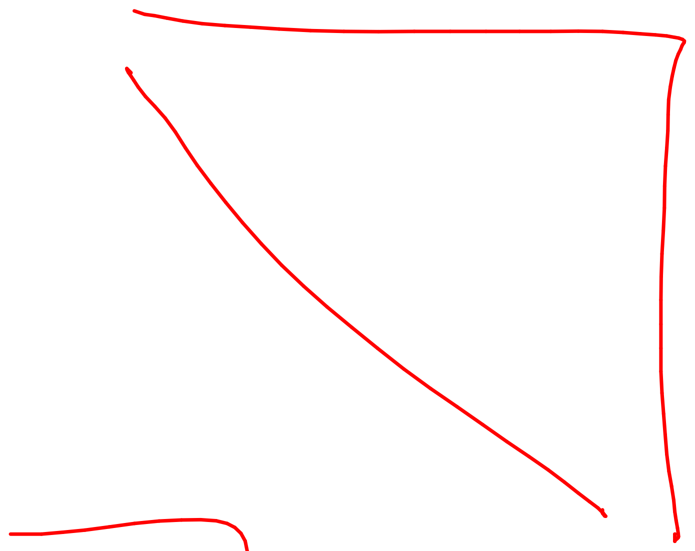
	0	1	2	3	4	5	6	7	8	9	10	...
best	0	1600	1296									

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

	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          ← n iterations
    best[i] = min{ best[j] + s[j+1][i]$^2$ }   → $\Theta(n^2)$
```

↑ n iterations

$\Theta(n^2)$

Problem: reduce image

scaling: distortion

deleting column: distortion

delete the most invisible seam

http://www.youtube.com/watch?v=qadw0BRKeMk

Shai Avidan
Mitsubishi Electric Research Lab
Ariel Shamir
The interdisciplinary Center & MERL

http://www.youtube.com/watch?v=qadw0BRKeMk

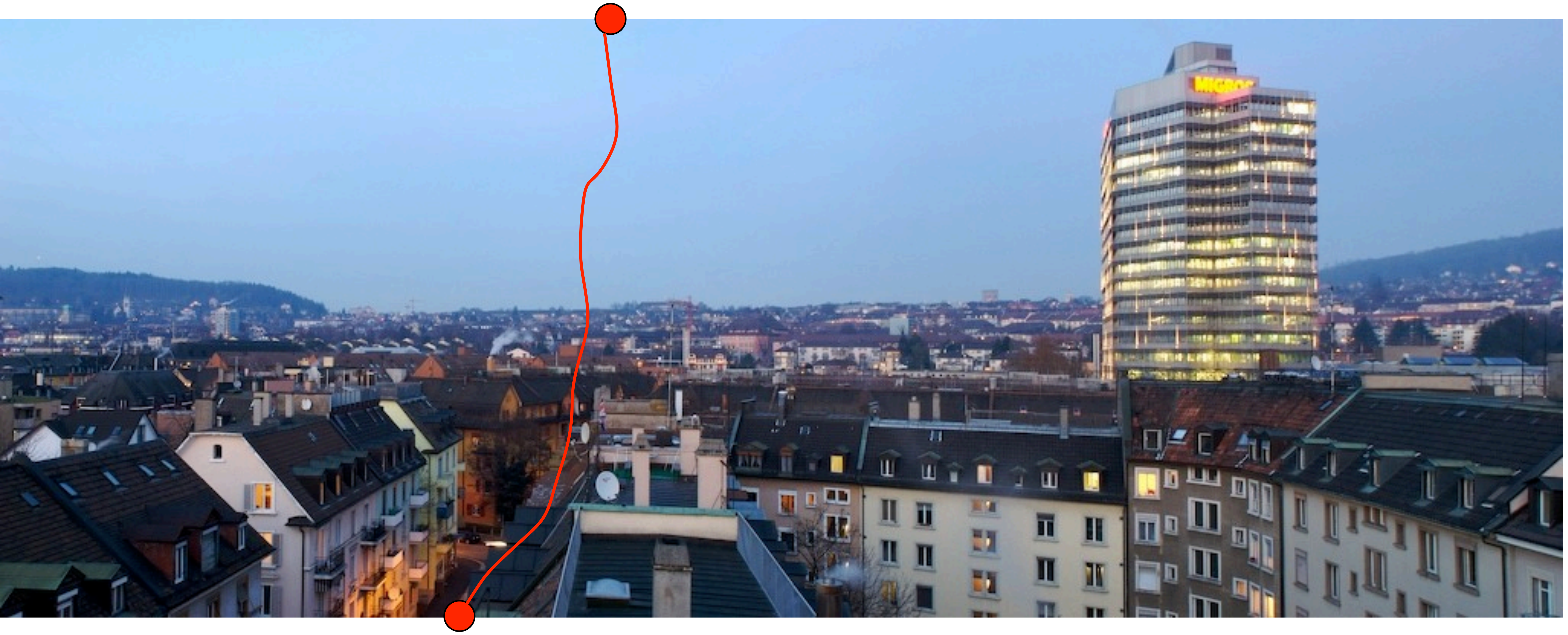
DEMO?

http://rsizr.com/

MIGROS

MIGROS

WHICH SEAM TO DELETE?

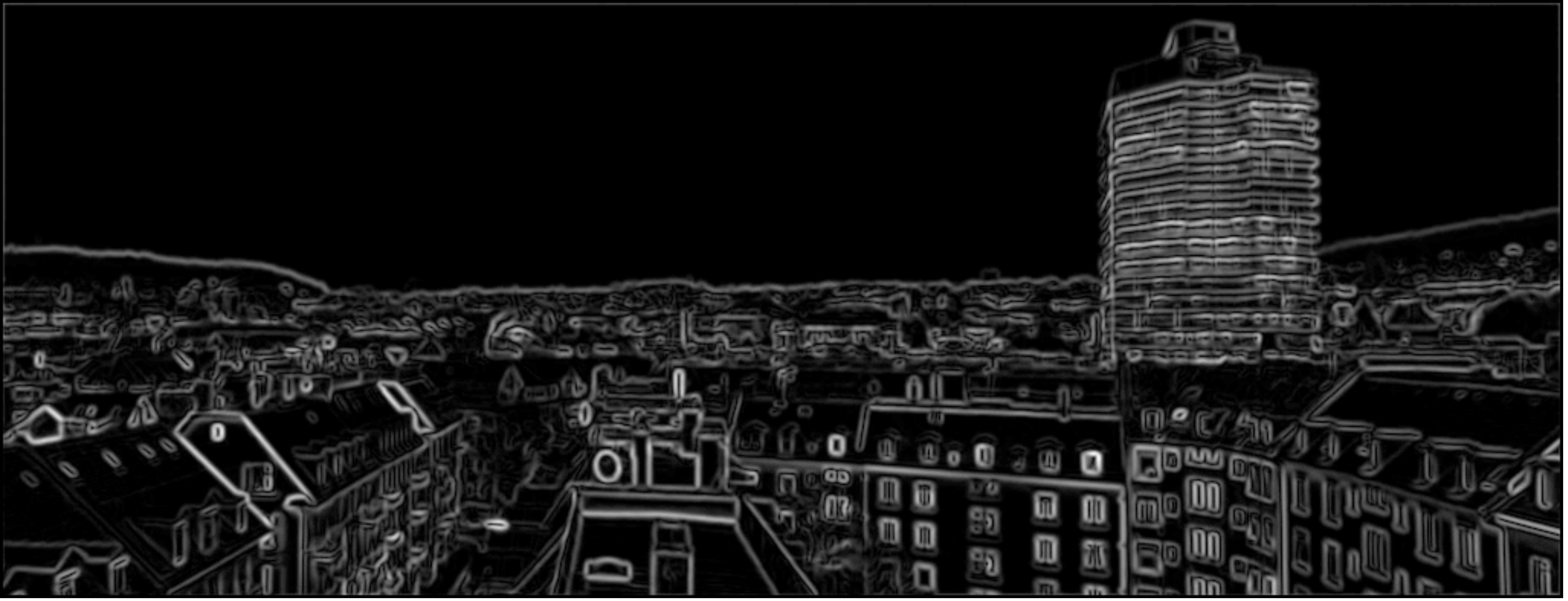


ENERGY OF AN IMAGE

$$e(\mathbf{I}) = \left|\frac{\partial}{\partial x}\mathbf{I}\right| + \left|\frac{\partial}{\partial y}\mathbf{I}\right|$$

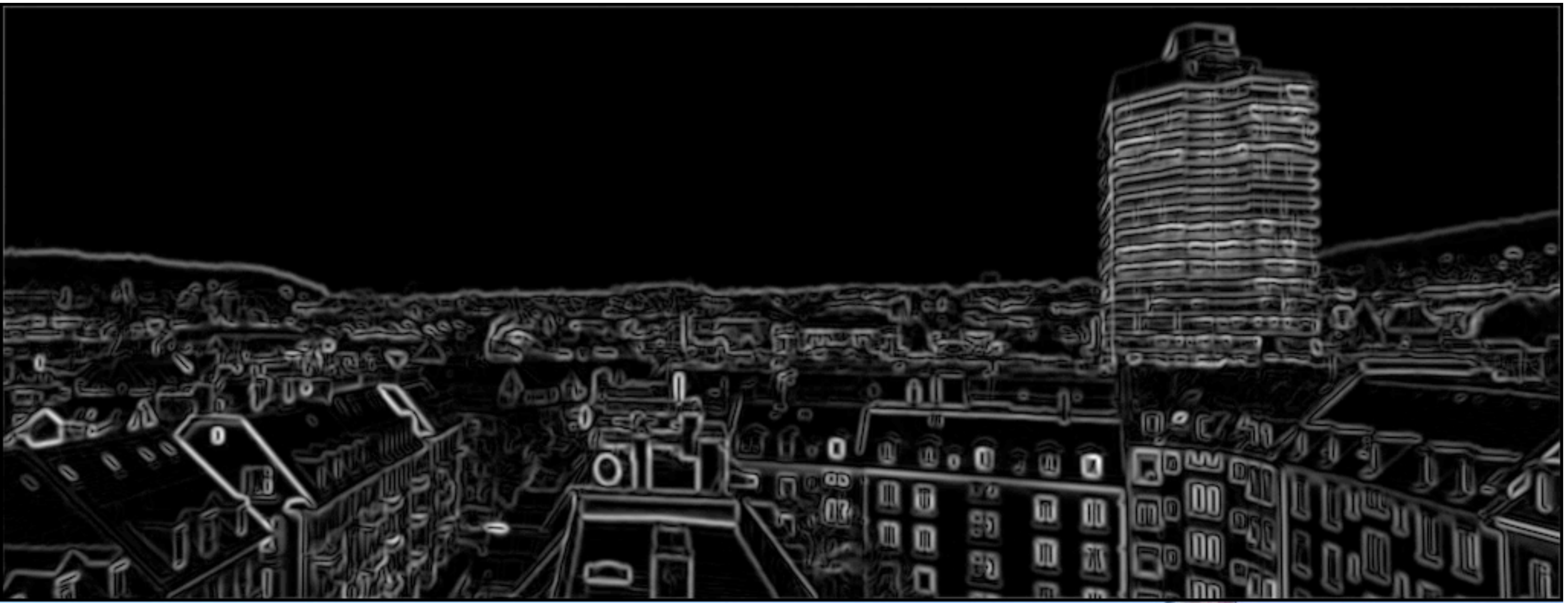
“magnitude of gradient at a pixel”

$$\frac{\partial}{\partial x} I_{x,y} = I_{x-1,y} - I_{x+1,y}$$



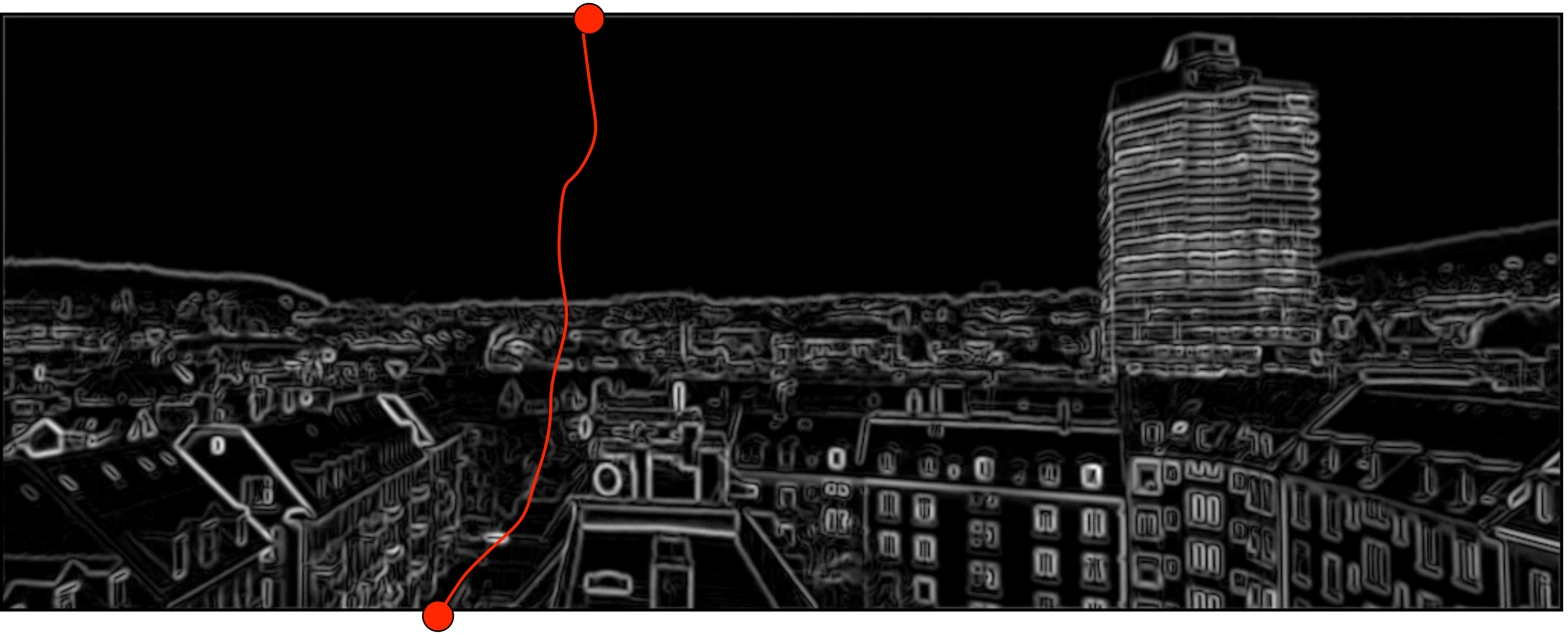
energy of sample image

thanks to Jason Lawrence for gradient software

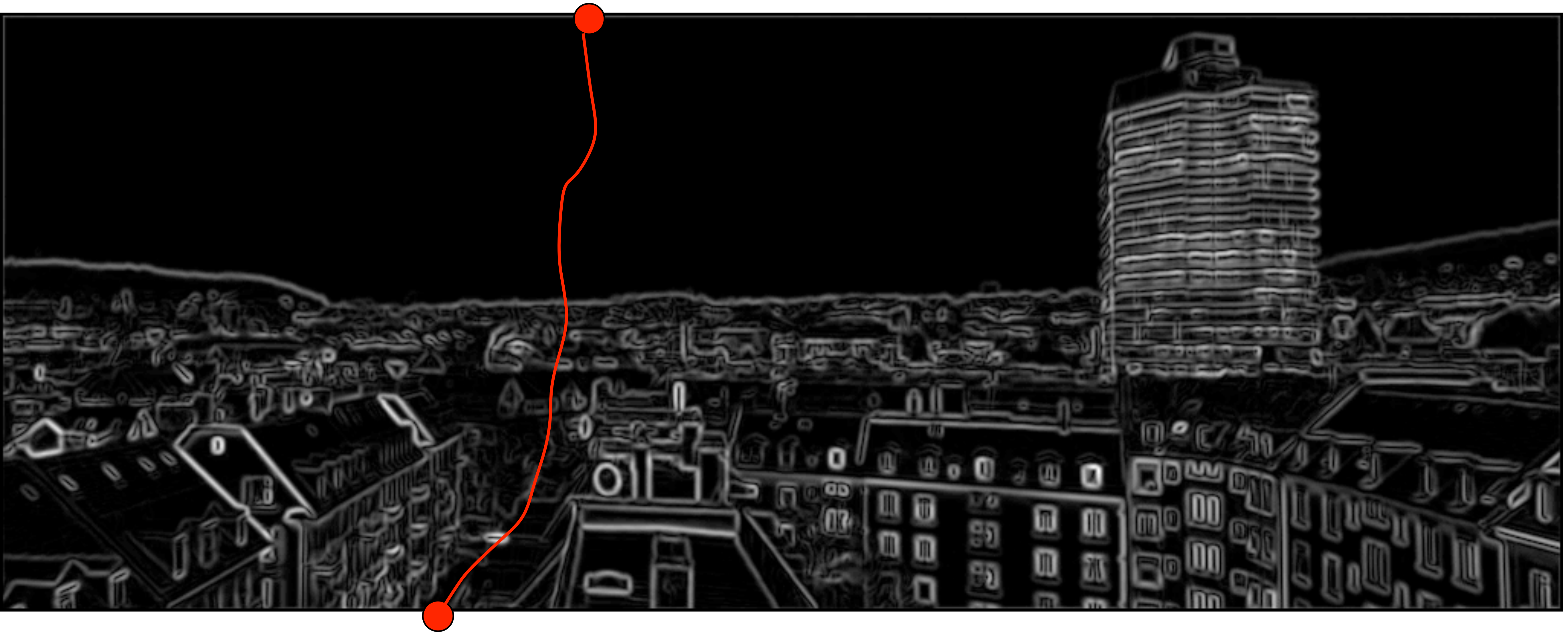


MIGROS

BEST SEAM HAS LOWEST ENERGY



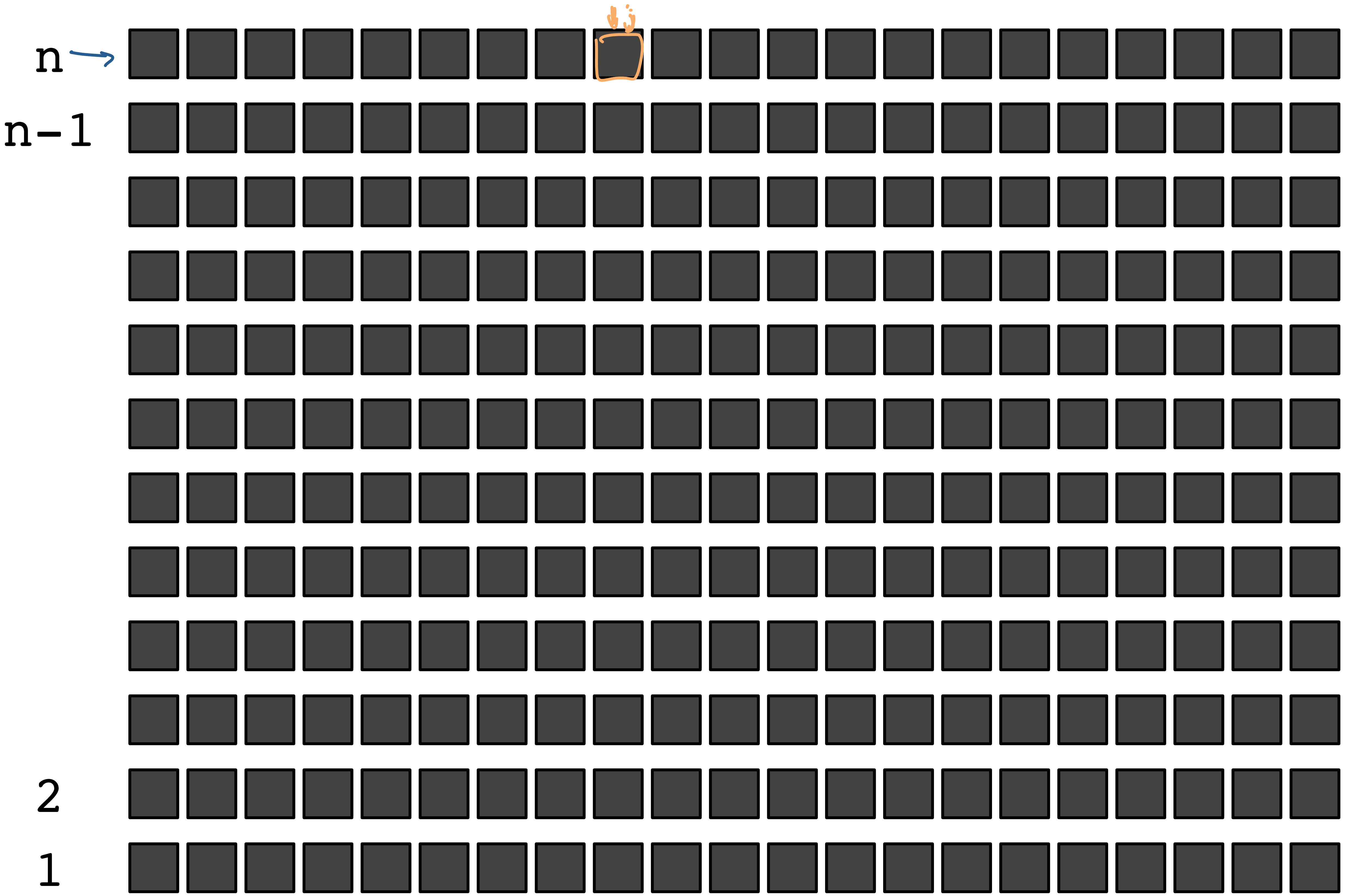
Finding lowest energy seam?



Define a variable:

$$S_i(j)$$

definition: $S_n(j)$



n

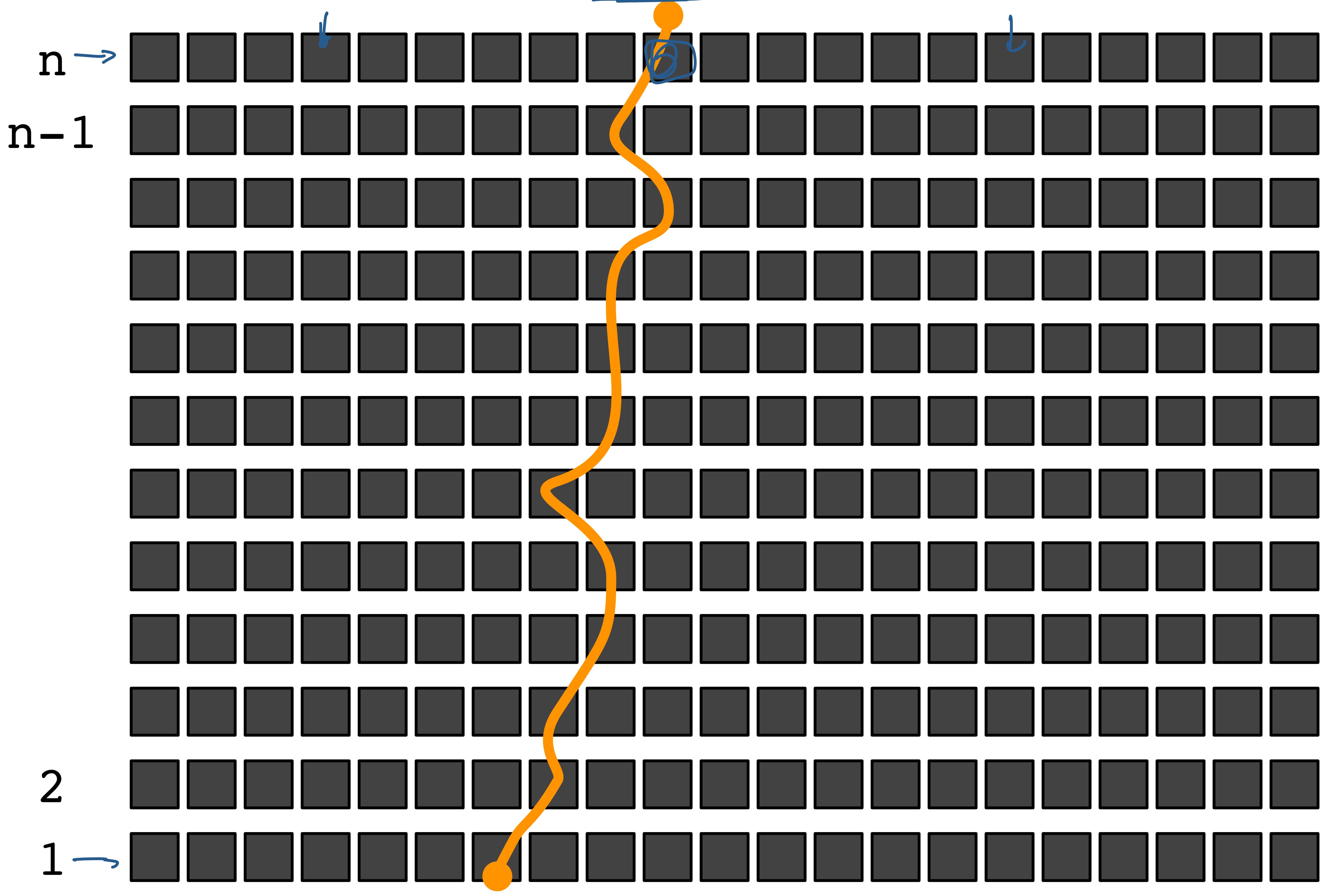
n-1

2

1

definition:

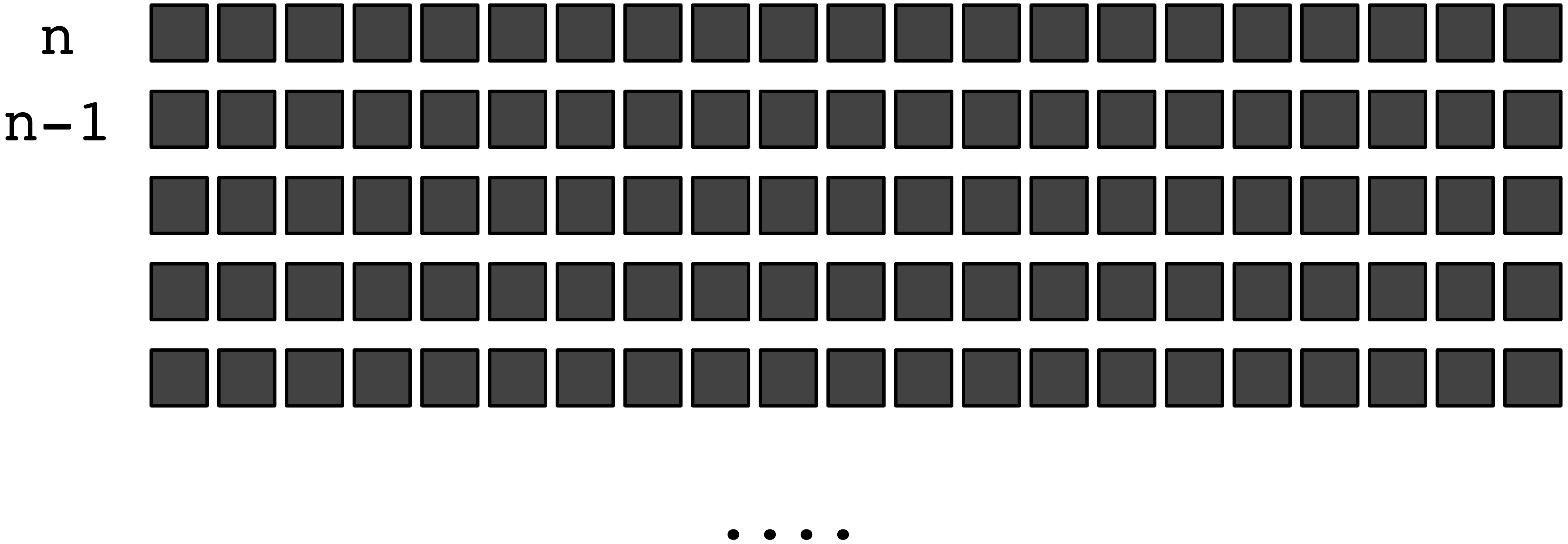
$S_n(j)$ best seam ending at (n,j)



BEST SEAM TO DELETE HAS TO BE THE BEST AMONG

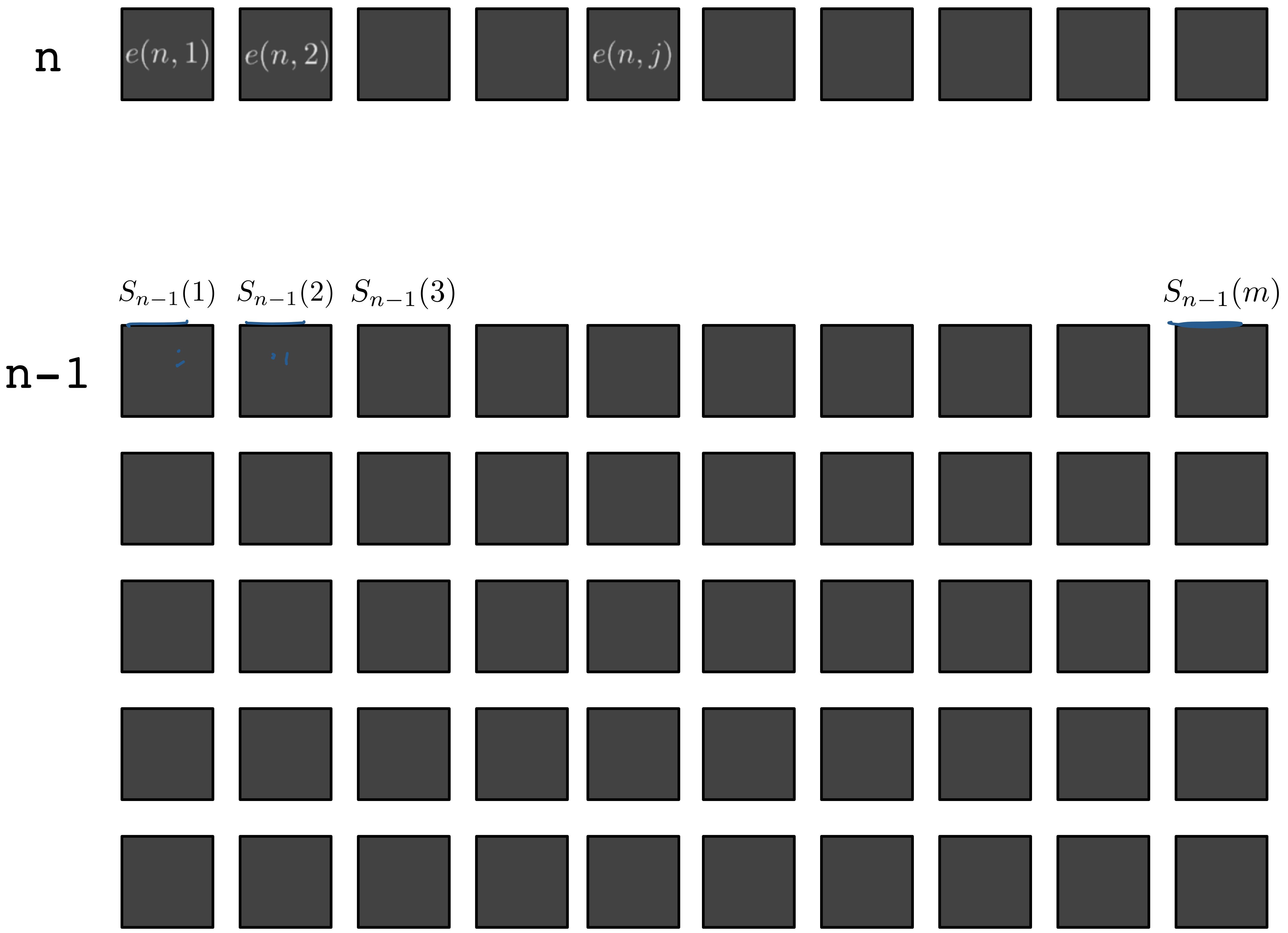
$$S_n(1), \underline{S_n}(2), \ldots, S_n(m)$$

IDEA: COMPUTE + COMPARE



SMALLER PROBLEM APPROACH

IMAGINE YOU HAVE THE SOLUTION TO THE FIRST $n-1$ ROWS

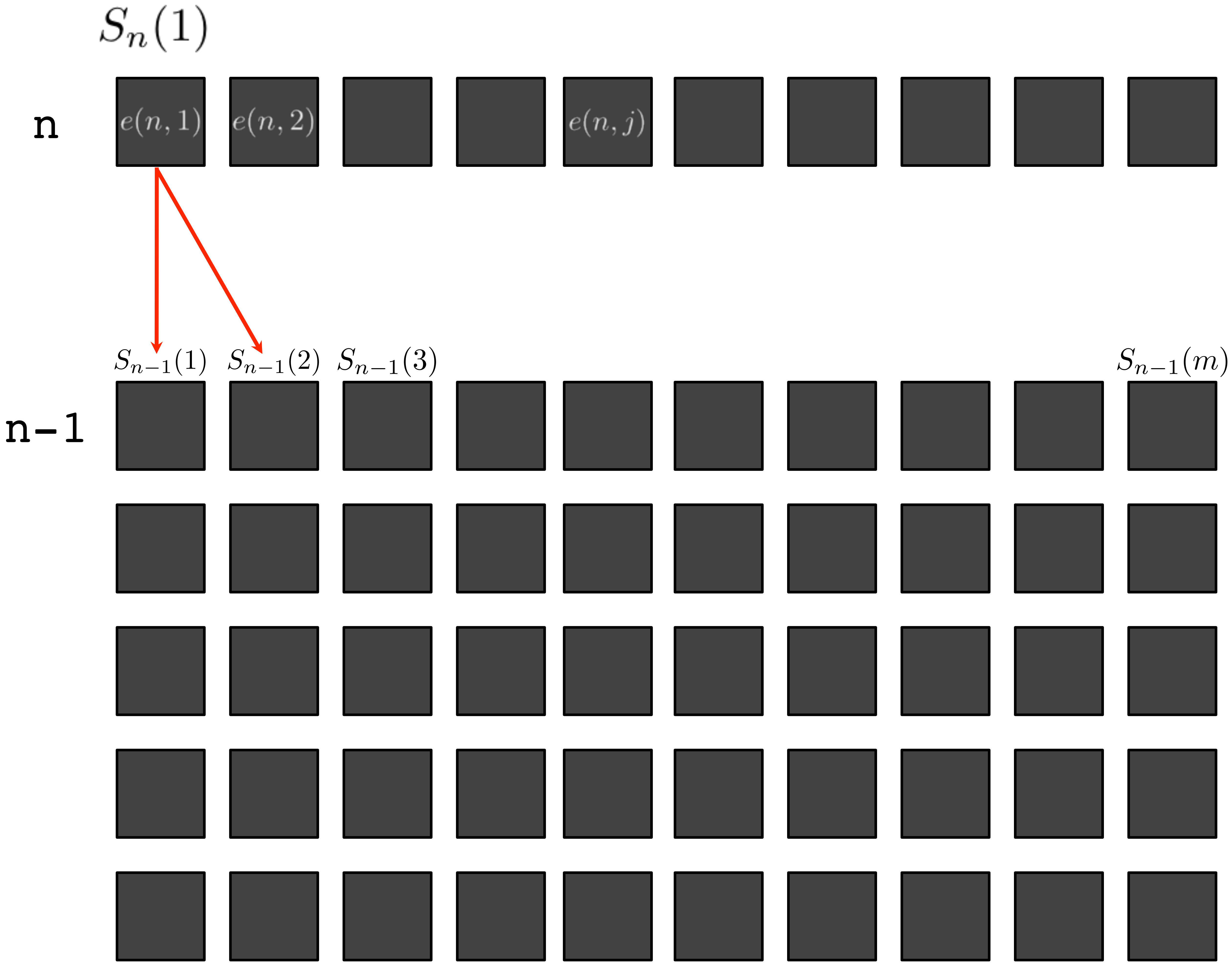


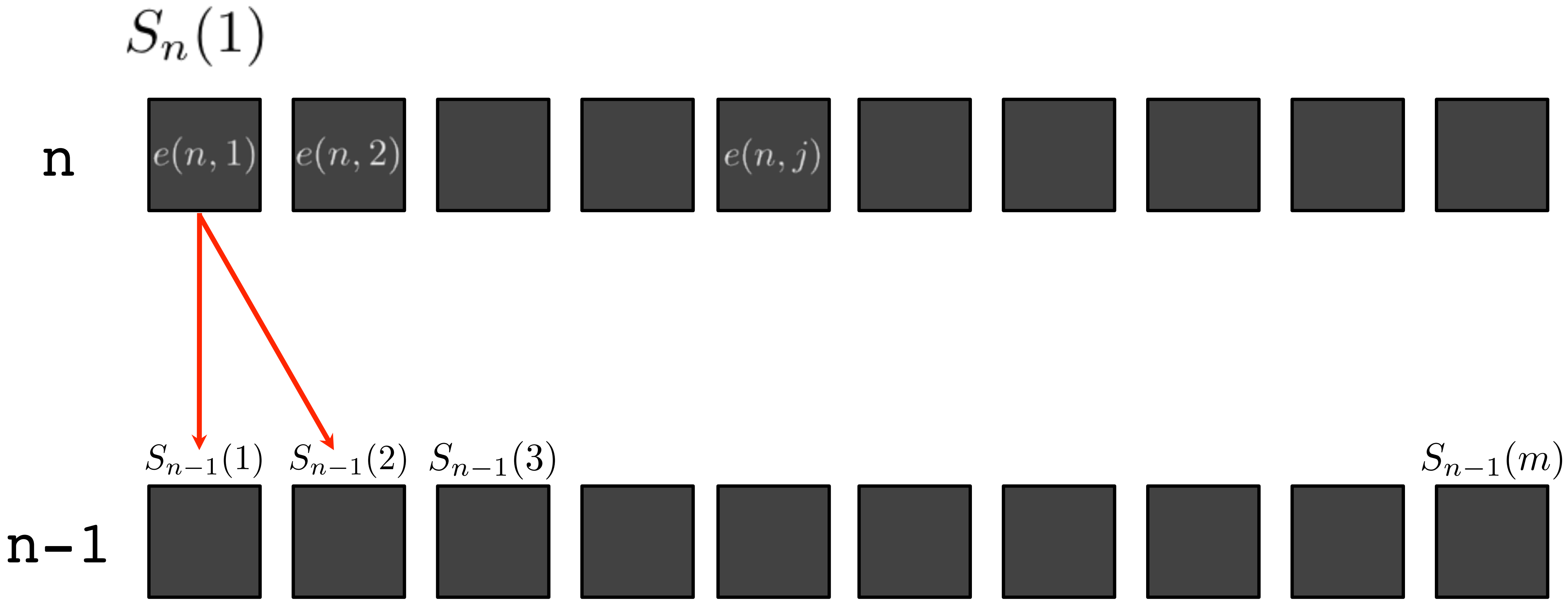
n

$e(n,1)$ $e(n,2)$ $e(n,j)$

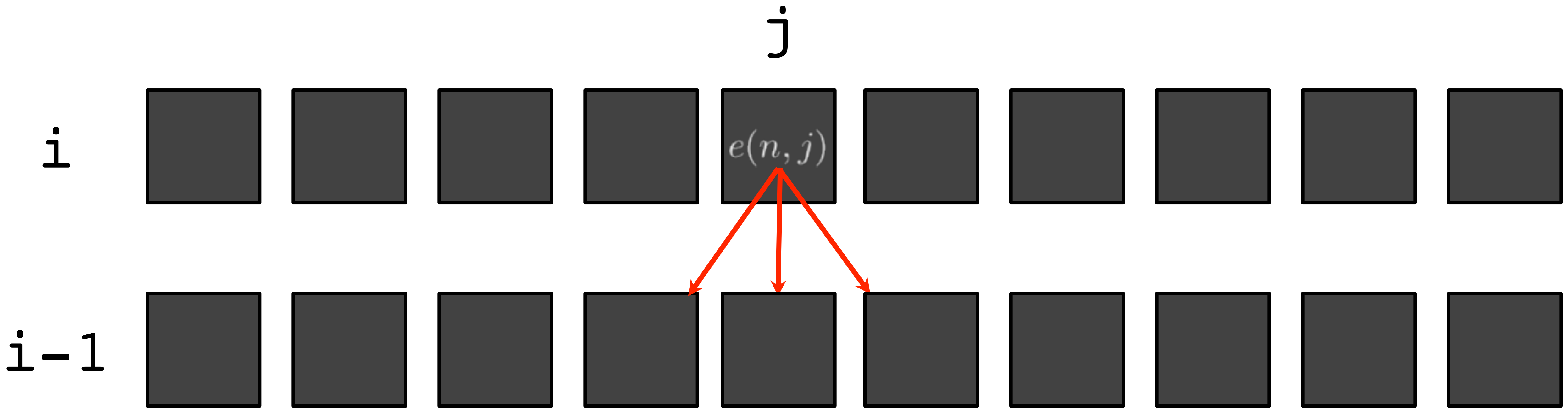
$S_{n-1}(1)$ $S_{n-1}(2)$ $S_{n-1}(3)$ $S_{n-1}(m)$

n-1

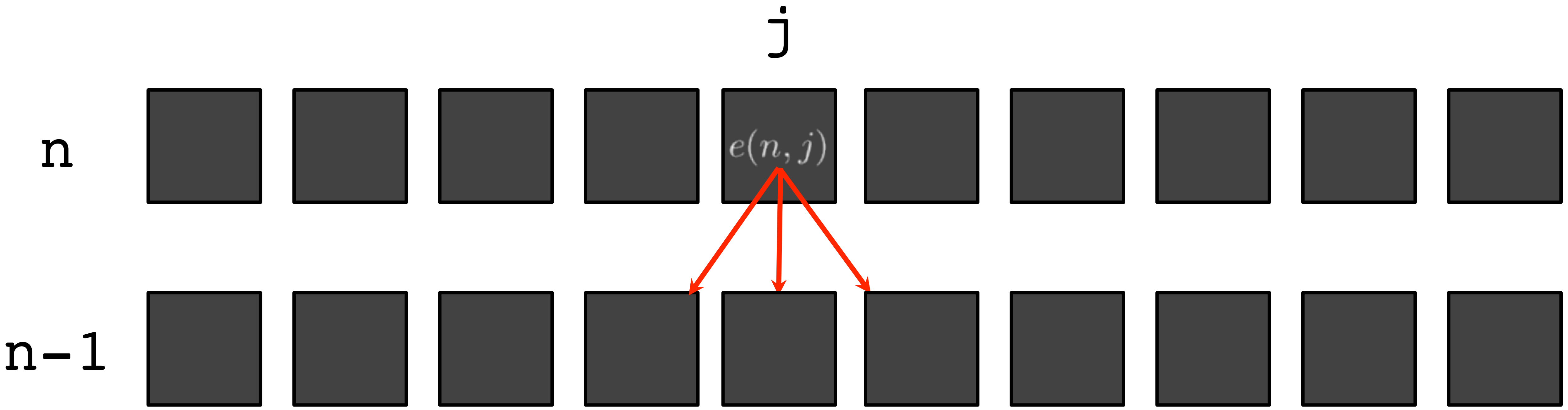

$S_n(1)$
n
$e(n,1)$
$e(n,2)$
$e(n,j)$
$S_{n-1}(1)$
$S_{n-1}(2)$
$S_{n-1}(3)$
$S_{n-1}(m)$
n-1



$$S_n(1) = e(n,1) + \min\{S_{n-1}(1), S_{n-1}(2)\}$$



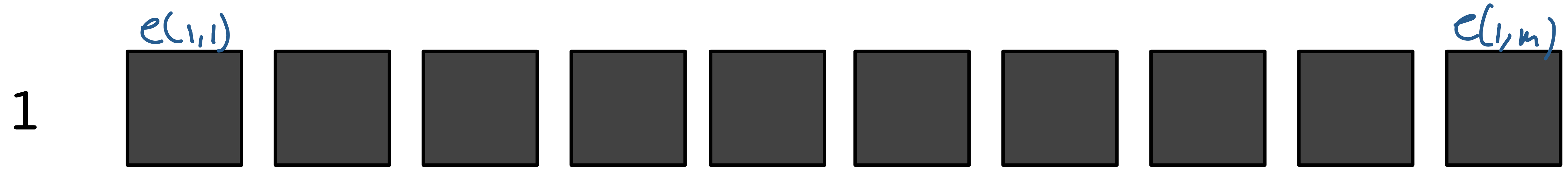
$$S_i(j) =$$



$$S_i(j) = e(i,j) + \min \begin{cases} S_{i-1}(j-1) \\ S_{i-1}(j) \\ S_{i-1}(j+1) \end{cases}$$

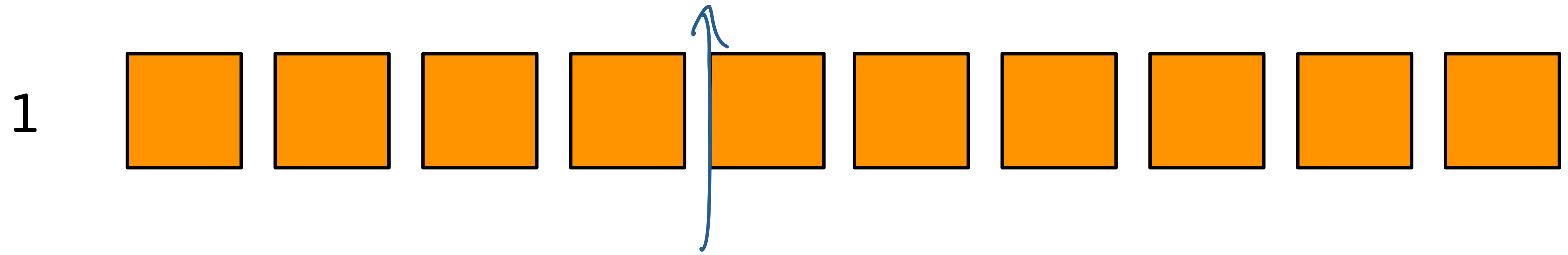
ALGORITHM

start at bottom of picture



ALGORITHM

start at bottom of picture.   initialize $S_1(i) = e(1,i)$

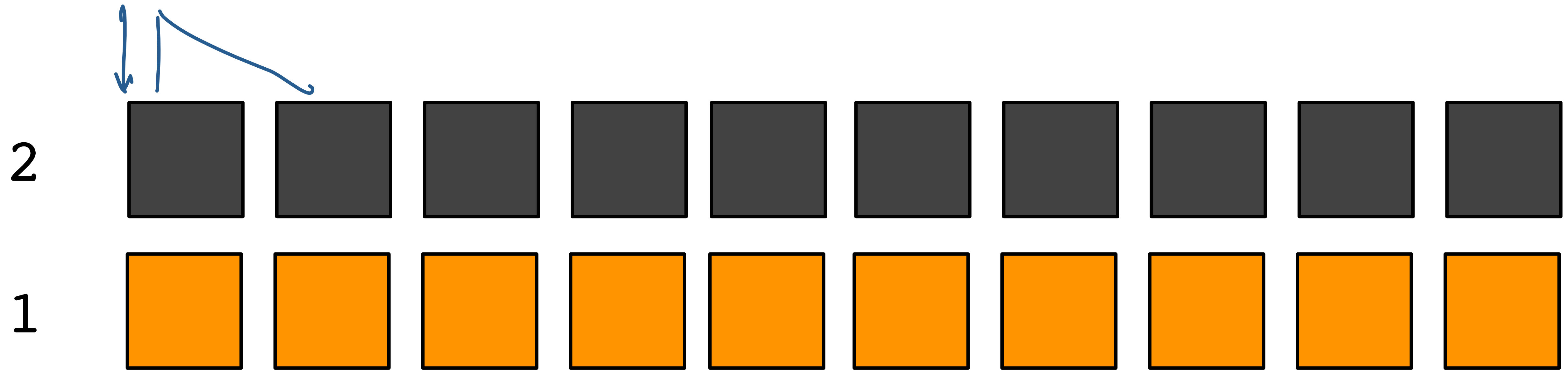


ALGORITHM

start at bottom of picture. initialize $S_1(i) = e(1, i)$

for `i=2,n` use formula to compute $\underline{S_{i+1}}(\cdot)$

$$\underline{S_i(j) = e(i,j) + \min \begin{cases} S_{i-1}(j-1) \\ S_{i-1}(j) \\ S_{i-1}(j+1) \end{cases}}$$

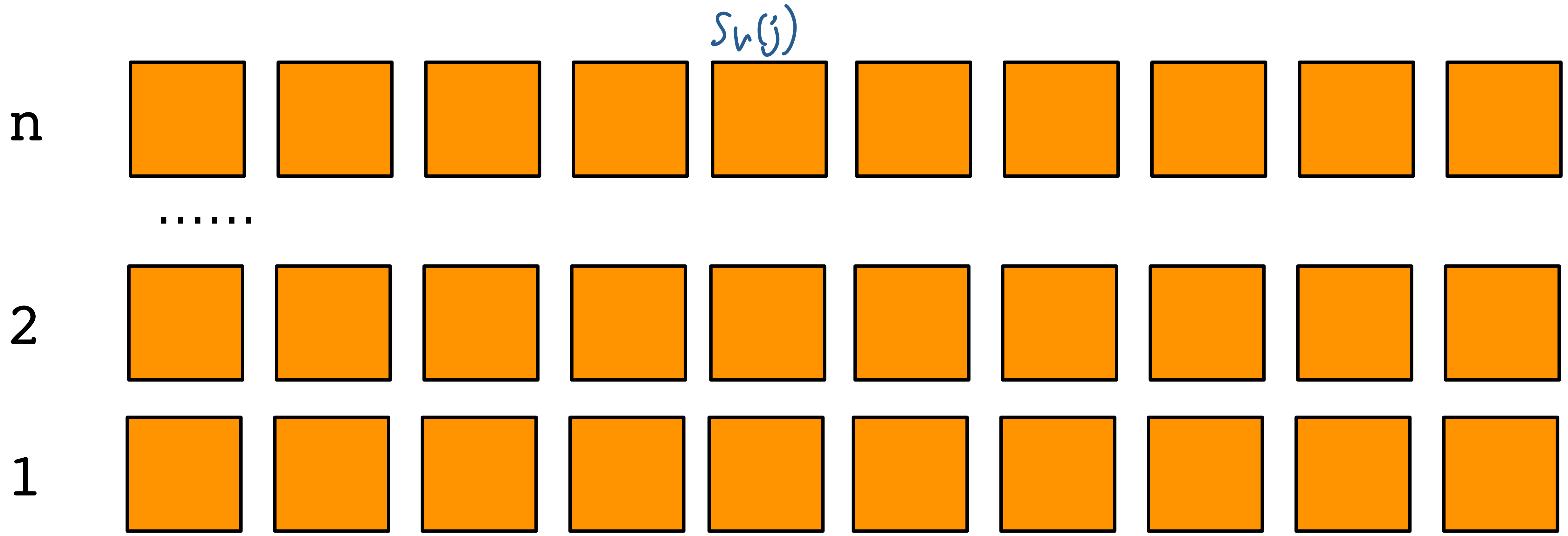


ALGORITHM

start at bottom of picture. initialize $S_1(i) = e(1, i)$

for `i=2,n` use formula to compute $S_{i+1}(\cdot)$

$$S_i(j) = e(i,j) + \min \begin{cases} S_{i-1}(j-1) \\ S_{i-1}(j) \\ S_{i-1}(j+1) \end{cases}$$



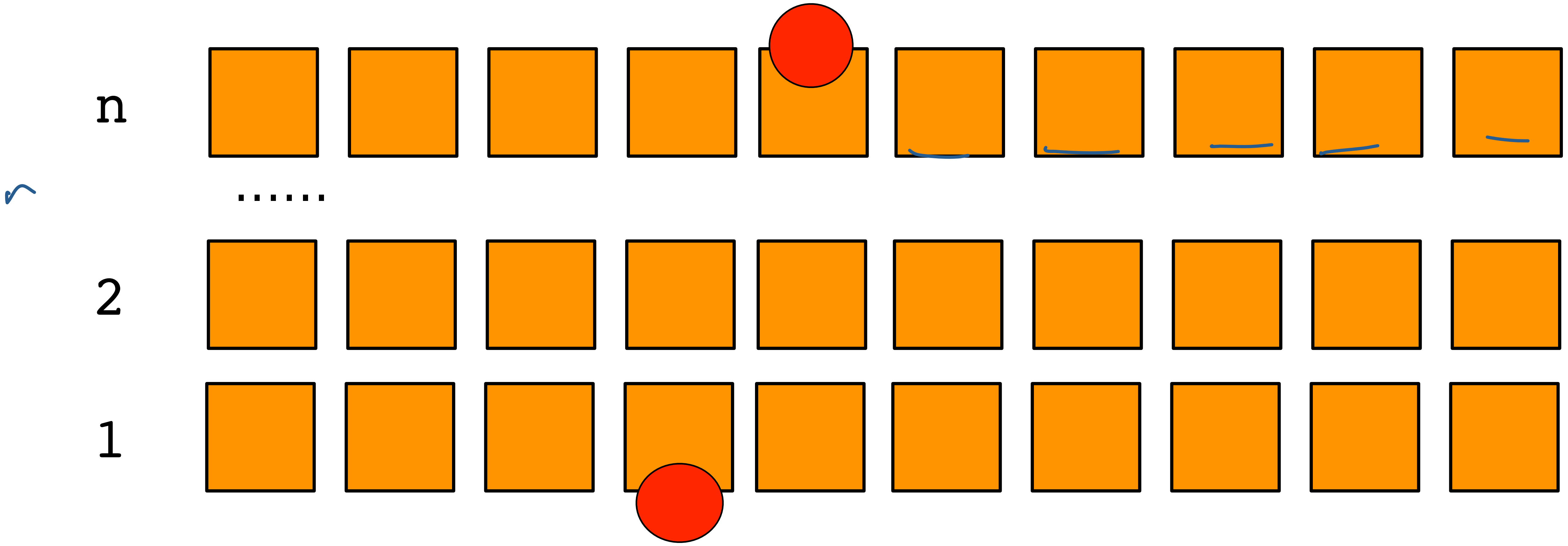
ALGORITHM

start at bottom of picture. initialize $S_1(i) = e(1,i)$

for i=2,n use formula to compute $S_{i+1}(\cdot)$

$$S_i(j) = e(i,j) + \min \begin{cases} S_{i-1}(j-1) \\ S_{i-1}(j) \\ S_{i-1}(j+1) \end{cases}$$

pick best among top row, backtrack.



RUNNING TIME

start at bottom of picture. initialize $S_1(i) = e(1,i)$

for `i=2,n` use formula to compute $S_{i+1}(\cdot)$

$$S_i(j) = e(i,j) + \min \begin{cases} S_{i-1}(j-1) \\ S_{i-1}(j) \\ S_{i-1}(j+1) \end{cases}$$

pick best among top row, backtrack.

RUNNING TIME

start at bottom of picture. initialize $S_1(i) = e(1, i)$

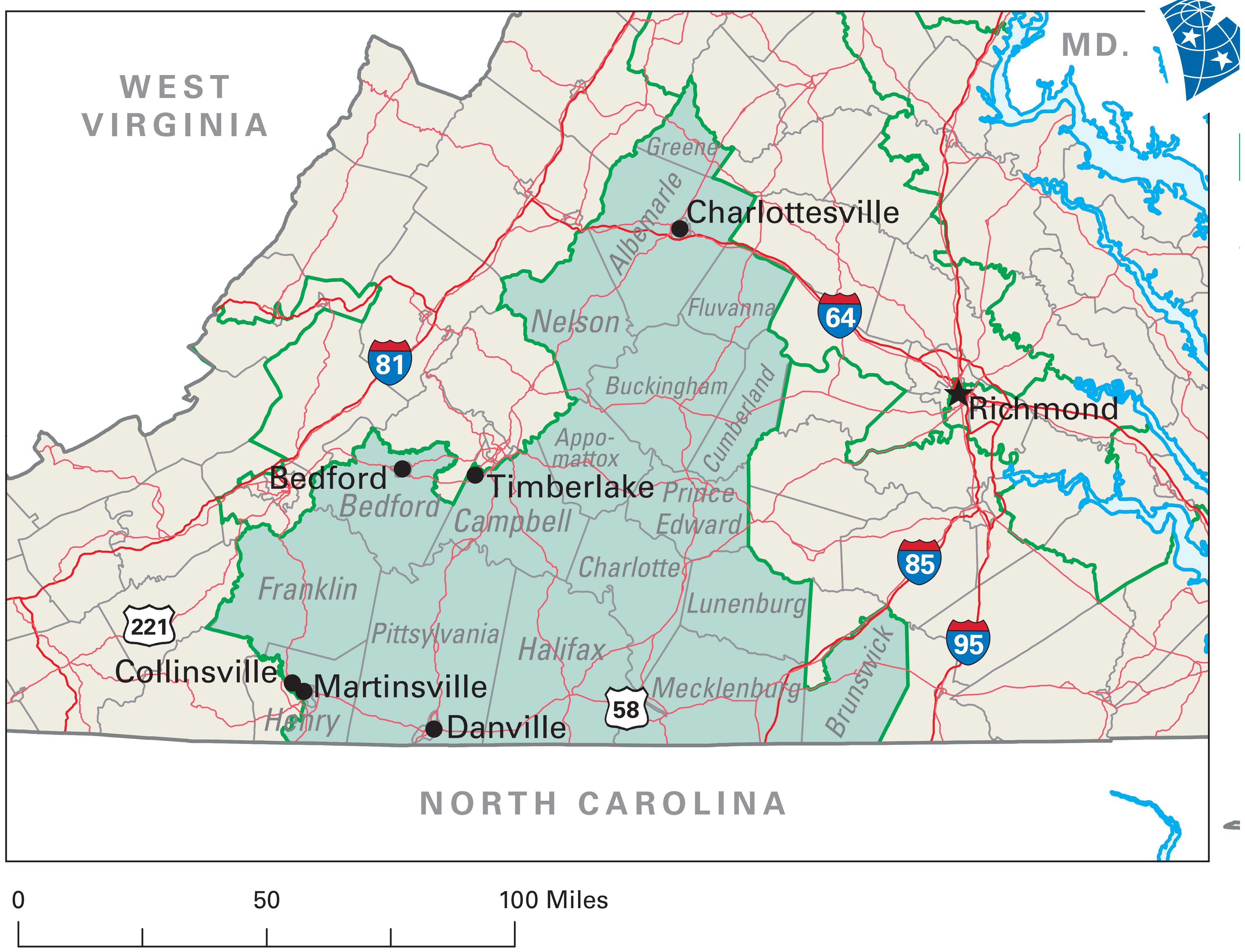
for `i=2,n` use formula to compute $S_{i+1}(\cdot)$

$$S_i(j) = e(i,j) + \min \begin{cases} S_{i-1}(j-1) \\ S_{i-1}(j) \\ S_{i-1}(j+1) \end{cases}$$

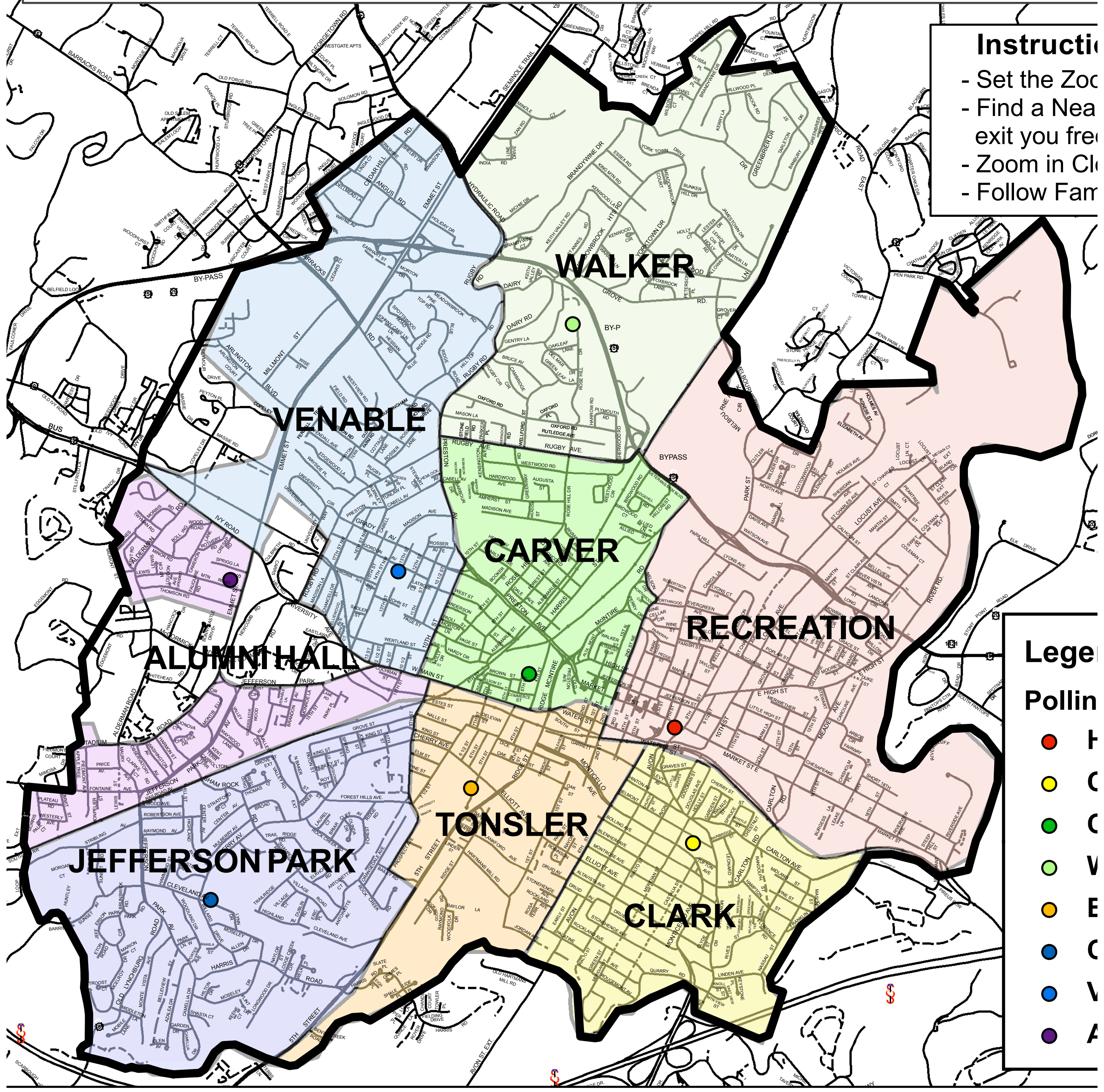
pick best among top row, backtrack.

Gerrymander

Congressional District 5



Map of Charlottesville Precincts and



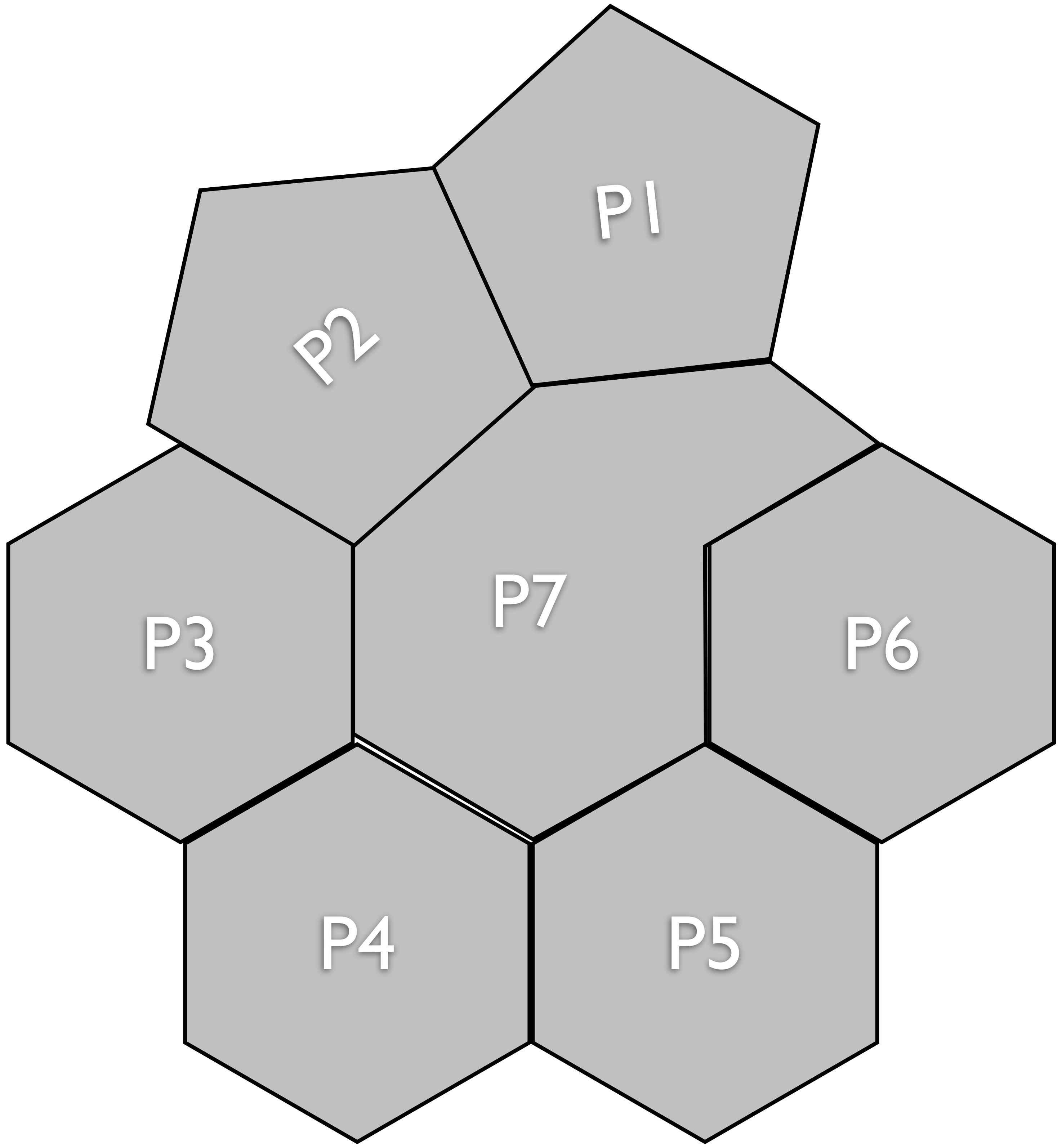
Instructi

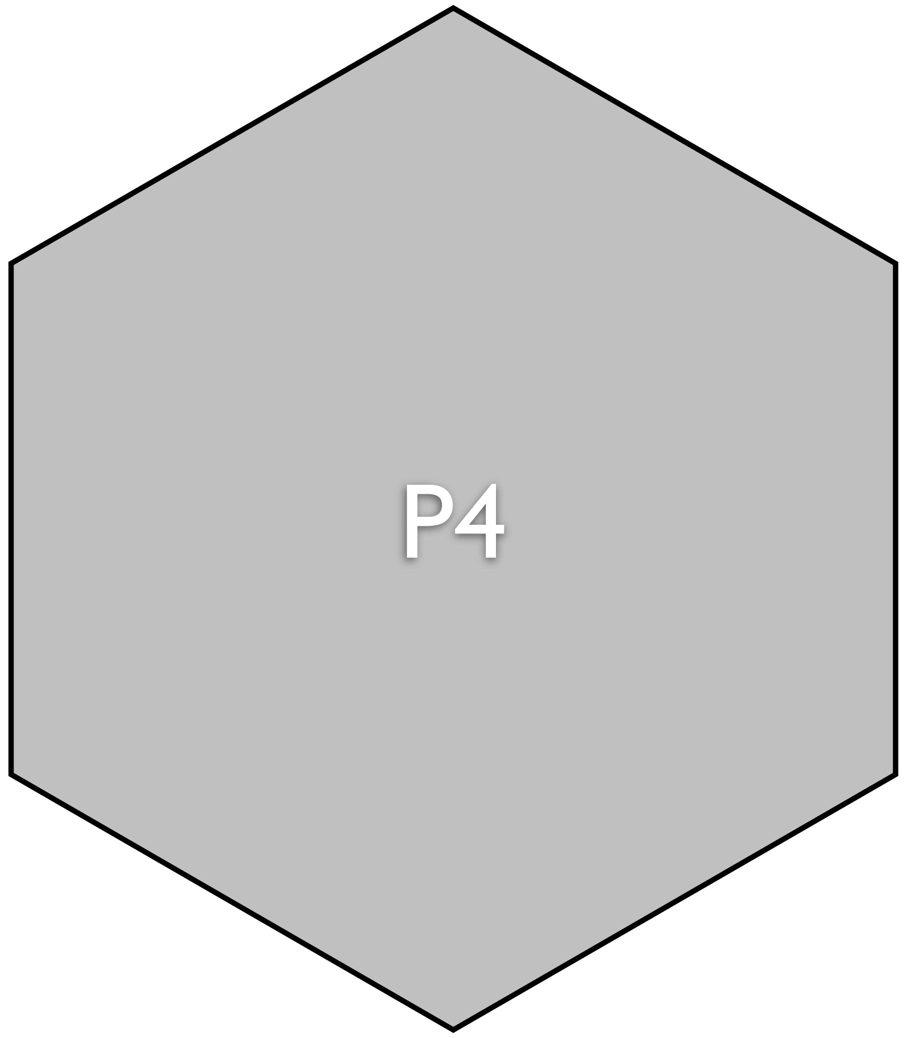
- Set the Zoo
- Find a Nea
  exit you fre
- Zoom in Cl
- Follow Fam

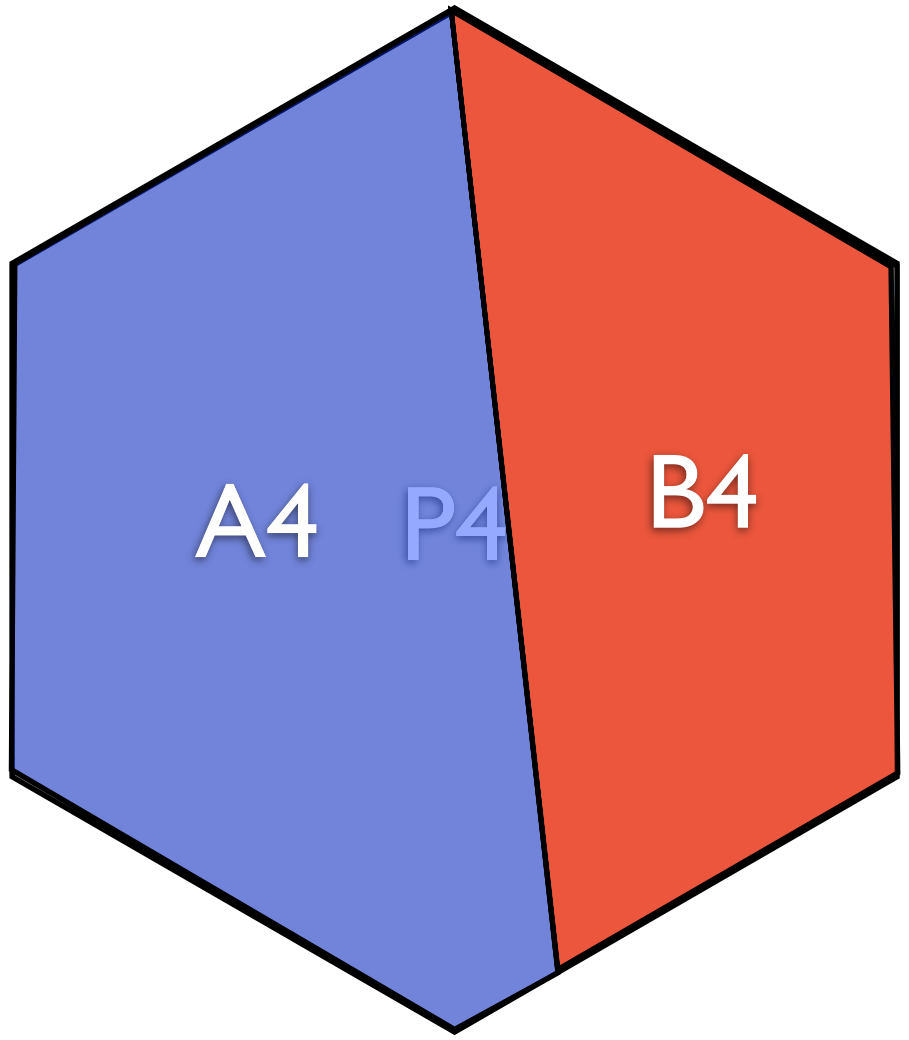
Lege

Pollin

- 🔴 H
- 🟡 C
- 🟢 C
- 🟢 W
- 🟠 E
- 🔵 C
- 🔵 V
- 🟣 A


P1
P2
P3
P7
P6
P4
P5


P4


A4
P4
B4

GERRYMANDER PROBLEM

given:

output:

GERRYMANDER PROBLEM

given: $m \quad A_1, A_2, \ldots, A_n$ n is even

output: D_1, D_2

such that $|D_1| = |D_2|$

$$A(D_1) > \frac{mn}{4}$$

$$A(D_2) > \frac{mn}{4}$$

or “failure” if no such solution is possible

EXAMPLE

THE TECHNIQUE

GERRYMANDER

imagine very last precinct and how it is assigned:

GERRYMANDER

$$S_{j,k,x,y} =$$

GERRYMANDER

$S_{j,k,x,y}$ = there is a split of first **j** precincts
in which |D1|=**k** and
x people in D1 vote A
y people in D2 vote A

$$S_{j,k,x,y} = S_{j-1,k-1,x-A_j,y} \ \vee \ S_{j-1,k,x,y-A_j}$$

```
GERRYMANDER(P,A,m)
    initialize array S[0,0,0,0]
```

$$S_{j,k,x,y} = S_{j-1,k-1,x-A_j,y} \; \vee \; S_{j-1,k,x,y-A_j}$$

GERRYMANDER(P,A,m)

```
initialize array S[0,0,0,0]
for j=1,...,n
      for k=1,...,n/2
         for x=0,...,jm
            for y=0,...,jm
               fill table according to equation
 search for true entry at S[n,n/2, >mn/4, >mn/4]
```

Scheduling

	start	end
sy333	2	3.25
en162	1	4
ma123	3	4
cs4102	3.5	4.75
cs4402	4	5.25
cs6051	4.5	6
sy333	5	6.5
cs1011	7	8

problem statement

$(a_1, \ldots, a_n)$

$(s_1, s_2, \ldots, s_n)$

$(f_1, f_2, \ldots, f_n)$ (sorted) $s_i < f_i$

(compatible)

find largest subset of activities C={a_i} such that

problem statement

$(a_1, \ldots, a_n)$

$(s_1, s_2, \ldots, s_n)$

$(f_1, f_2, \ldots, f_n)$ (s0rted) $s_i < f_i$

(compatible)

find largest subset of activities C={a$_i$} such that

$$a_i, a_j \in C, i < j$$

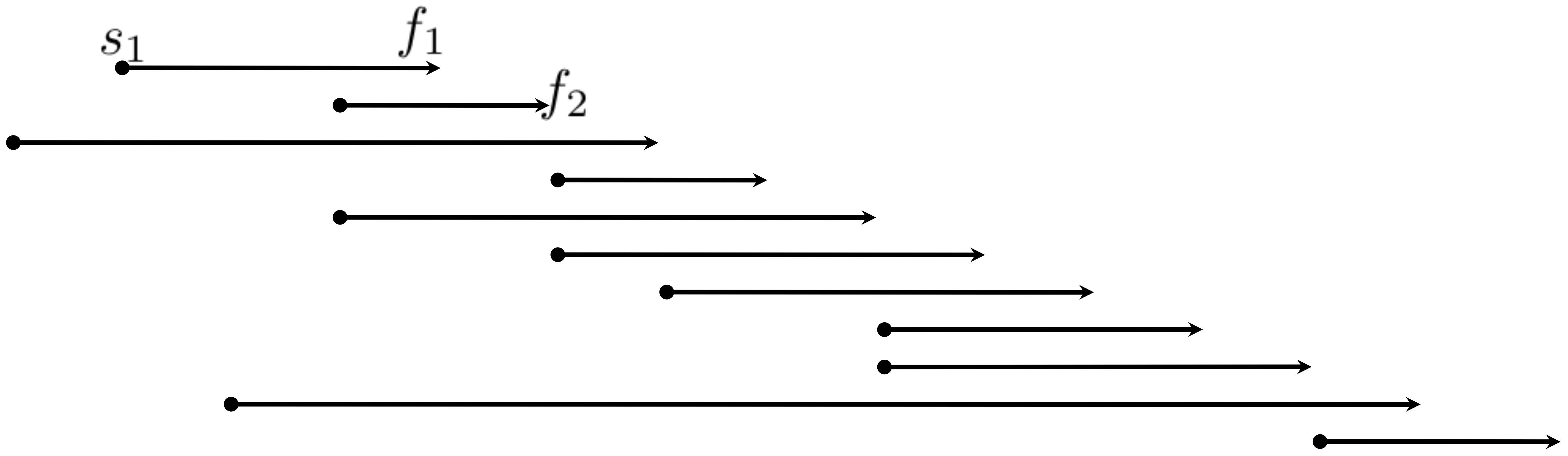
$$f_i \leq s_j$$

problem statement

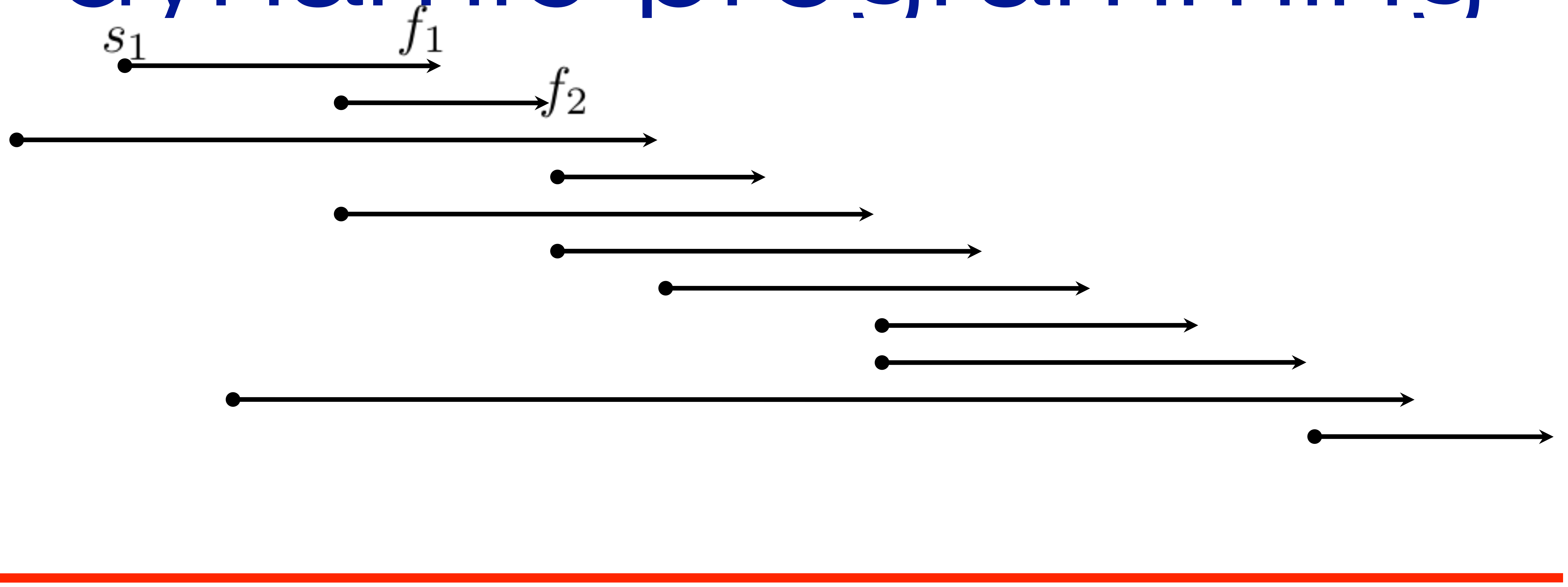
$(a_1, \ldots, a_n)$

$(s_1, s_2, \ldots, s_n)$

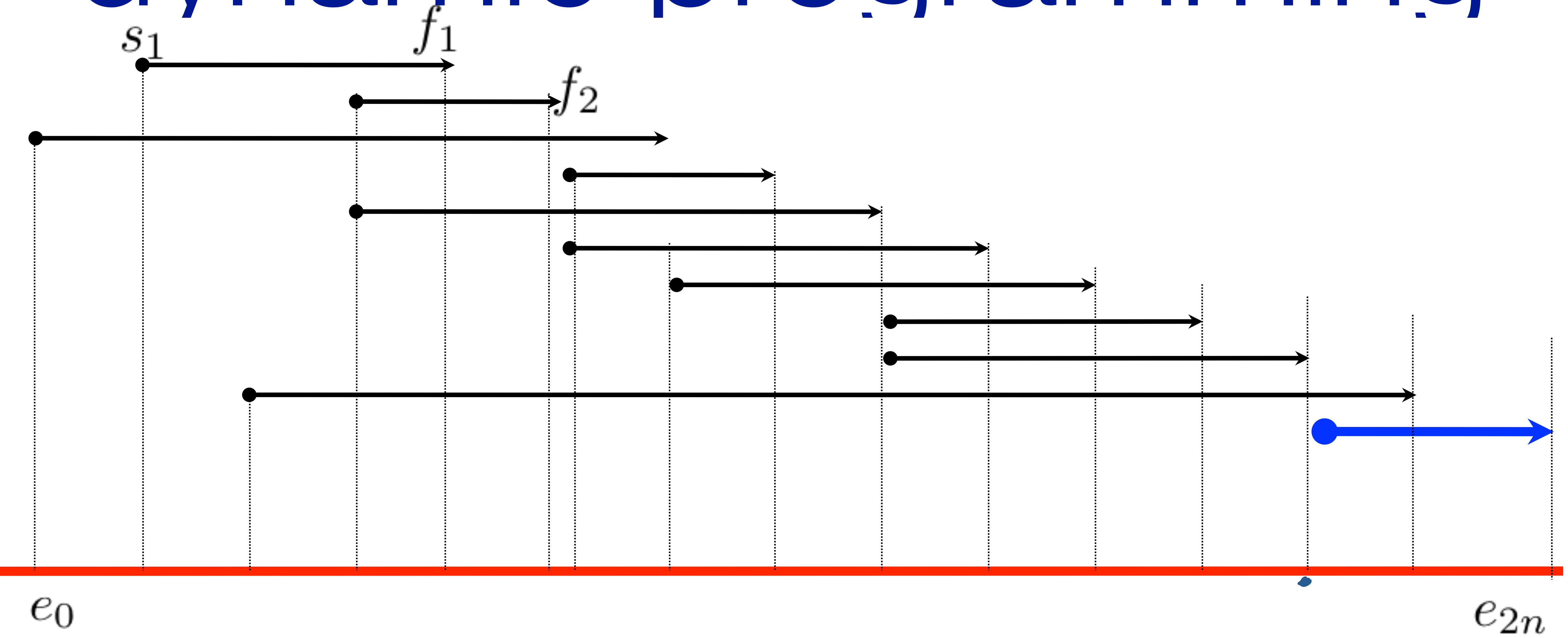
$(f_1, f_2, \ldots, f_n)$ LiRpo. $s_i < f_i$



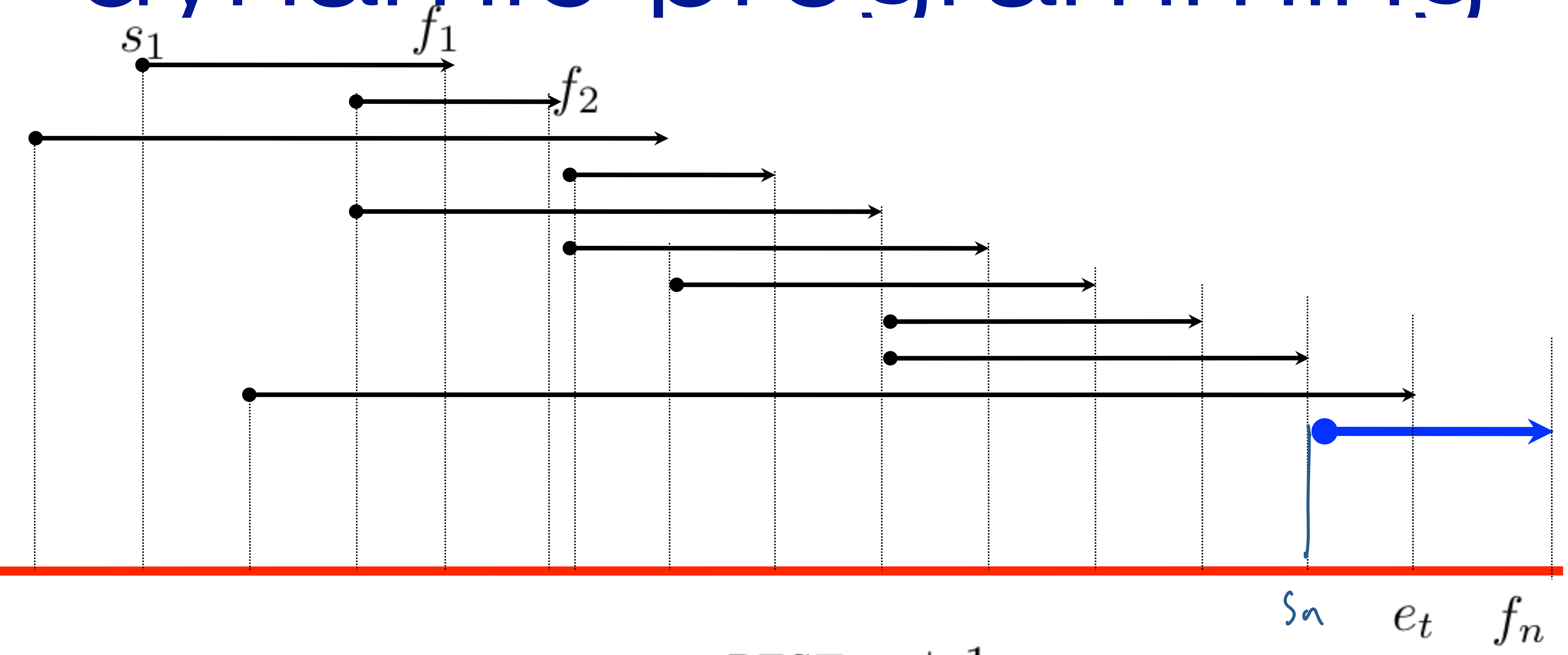
dynamic programming



dynamic programming

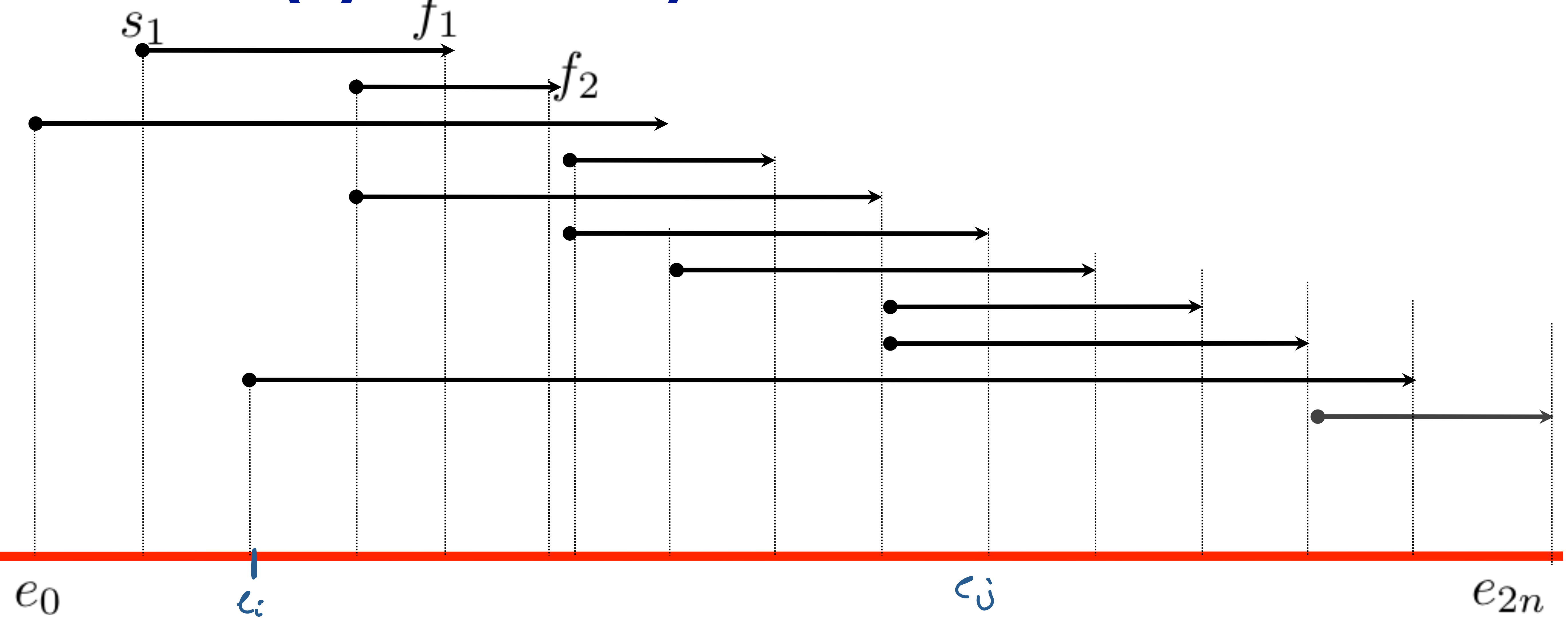


dynamic programming



$$\mathrm{BEST}_{f_n} = \max \begin{cases} \mathrm{BEST}_{s_n} + 1 & a_n \text{ in:} \\ \mathrm{BEST}_{e_t} & a_n \text{ out:} \end{cases}$$

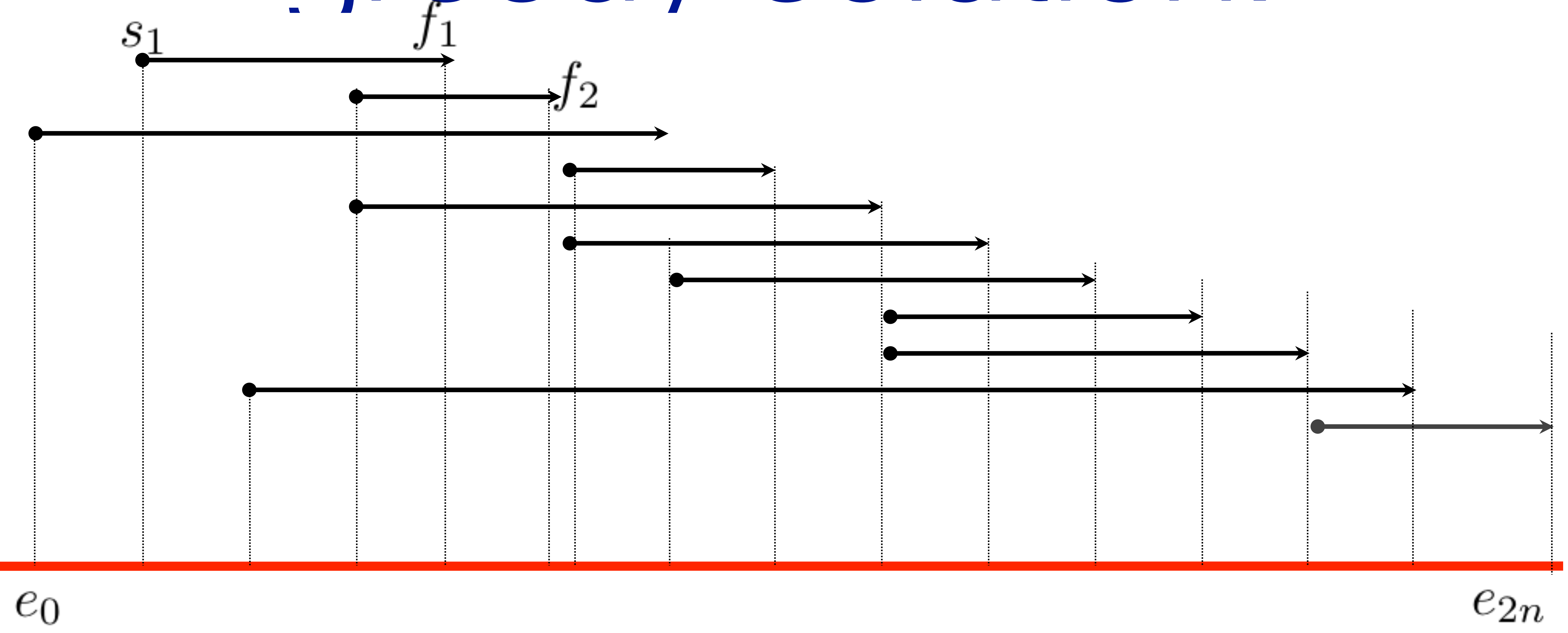
greedy solution:



[illegible]

$\mathrm{SOLTN}_{i,j}$

greedy solution:



$\mathrm{SOLTN}_{i,j}$

RZLyW $\mathrm{SOLTN}_{0,2n}$