PROBLEM: REDUCE IMAGE

scaling: distortion
deleting column: distortion
delete the most invisible seam

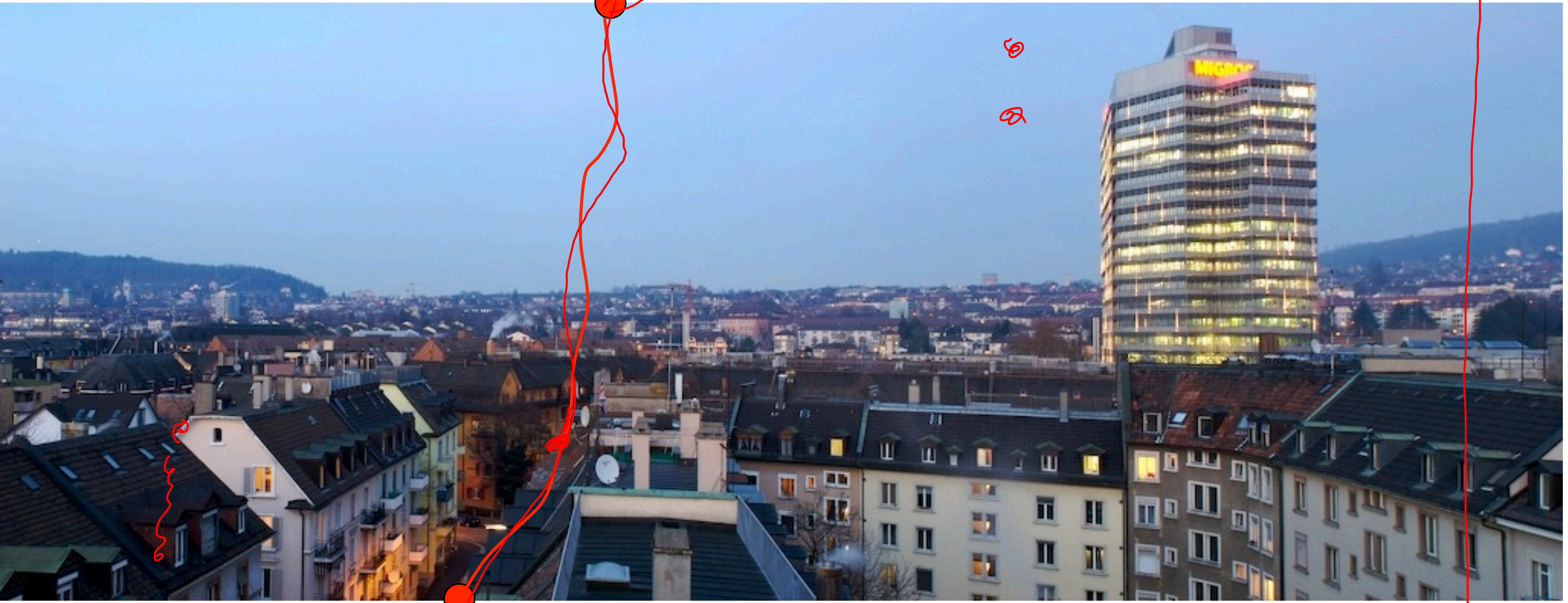
http://www.youtube.com/watch?v=qadw0BRKeMk

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

DEMO?

http://rsizr.com/

WHICH SEAM TO DELETE?



Remove a path

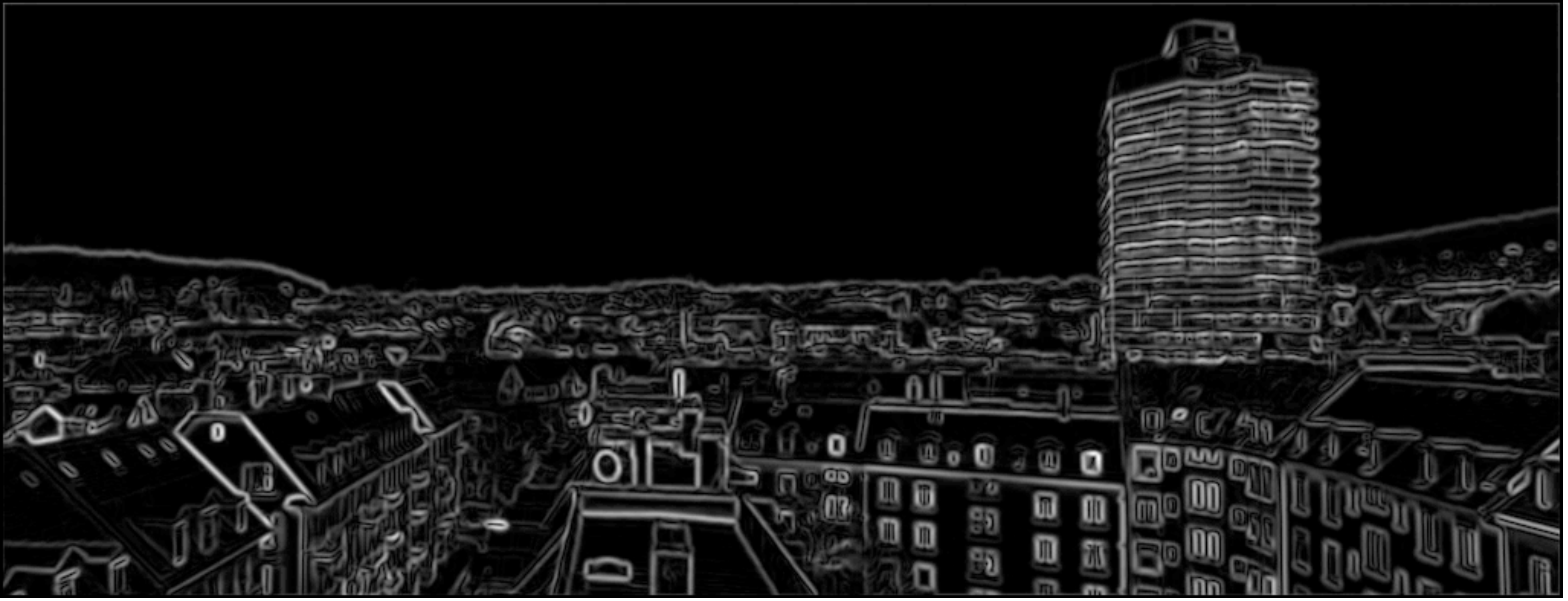
Seam: a path from the bottom row to the top row in which each pixel is connected to its UL, UM, UR neighbor

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"

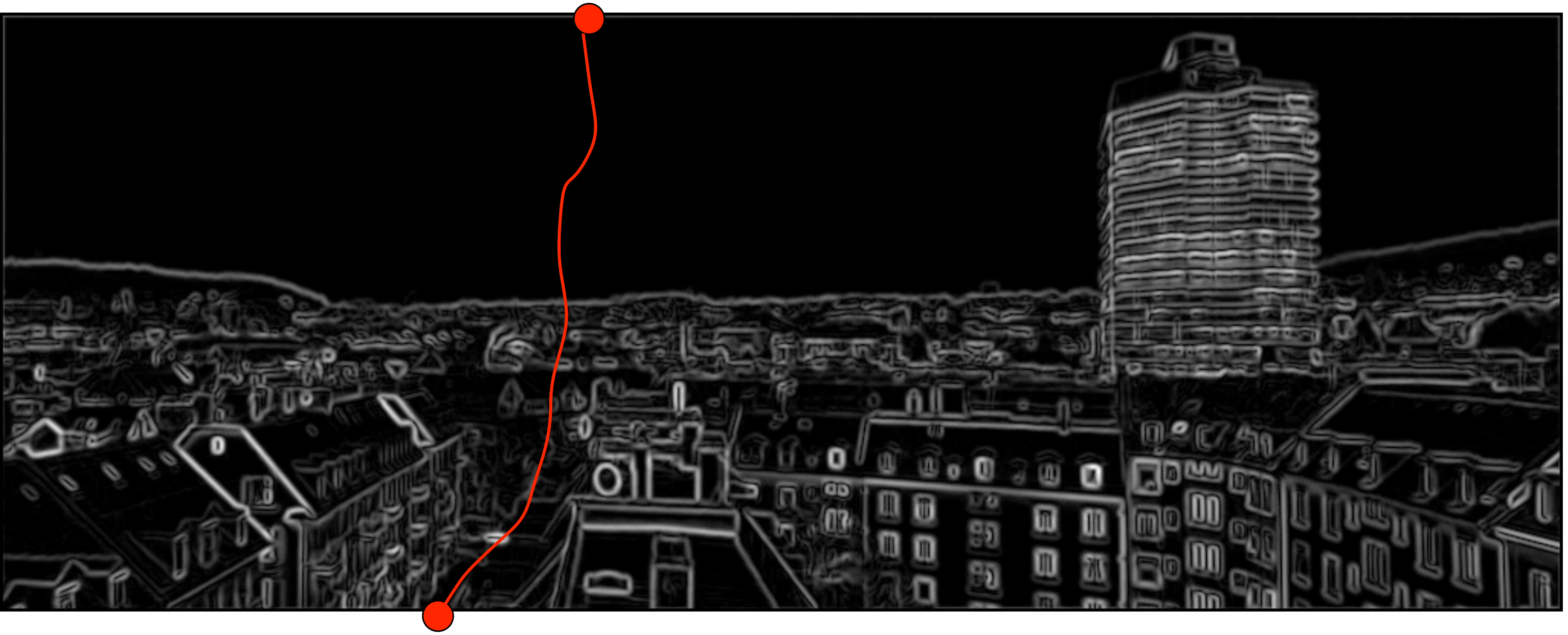
$$\frac{\partial}{\partial x} I_{x,y} = I_{x-1,y} - I_{x+1,y}$$



energy of sample image

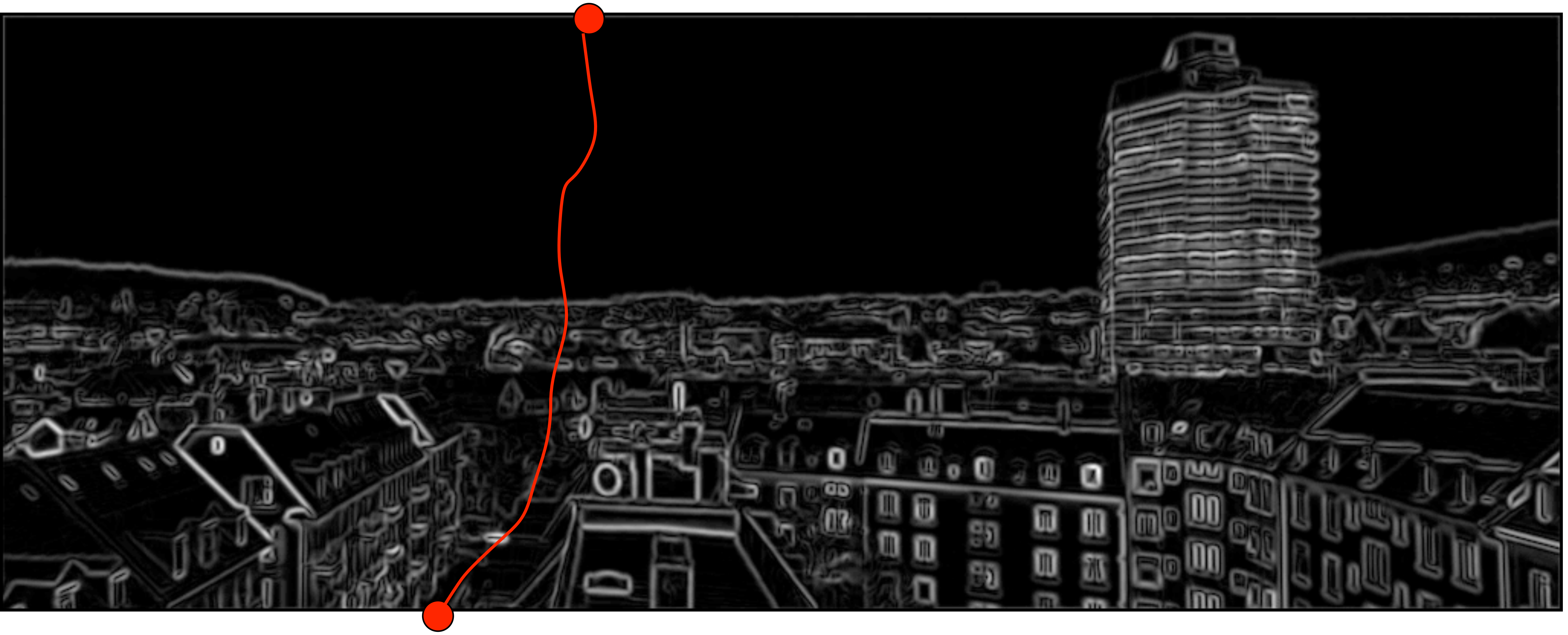
thanks to Jason Lawrence for gradient software

BEST SEAM HAS LOWEST ENERGY



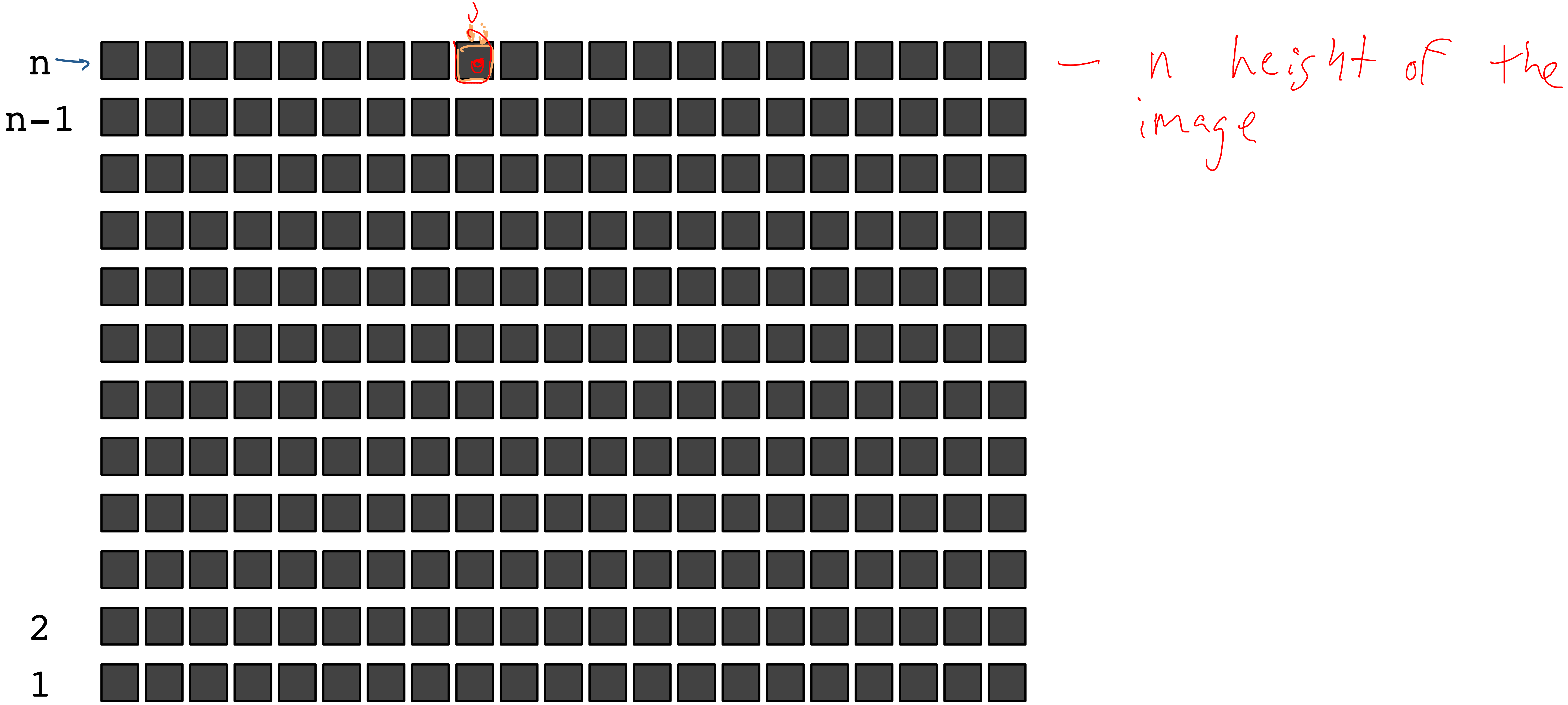
identify the lowest energy seam

Finding lowest energy seam?



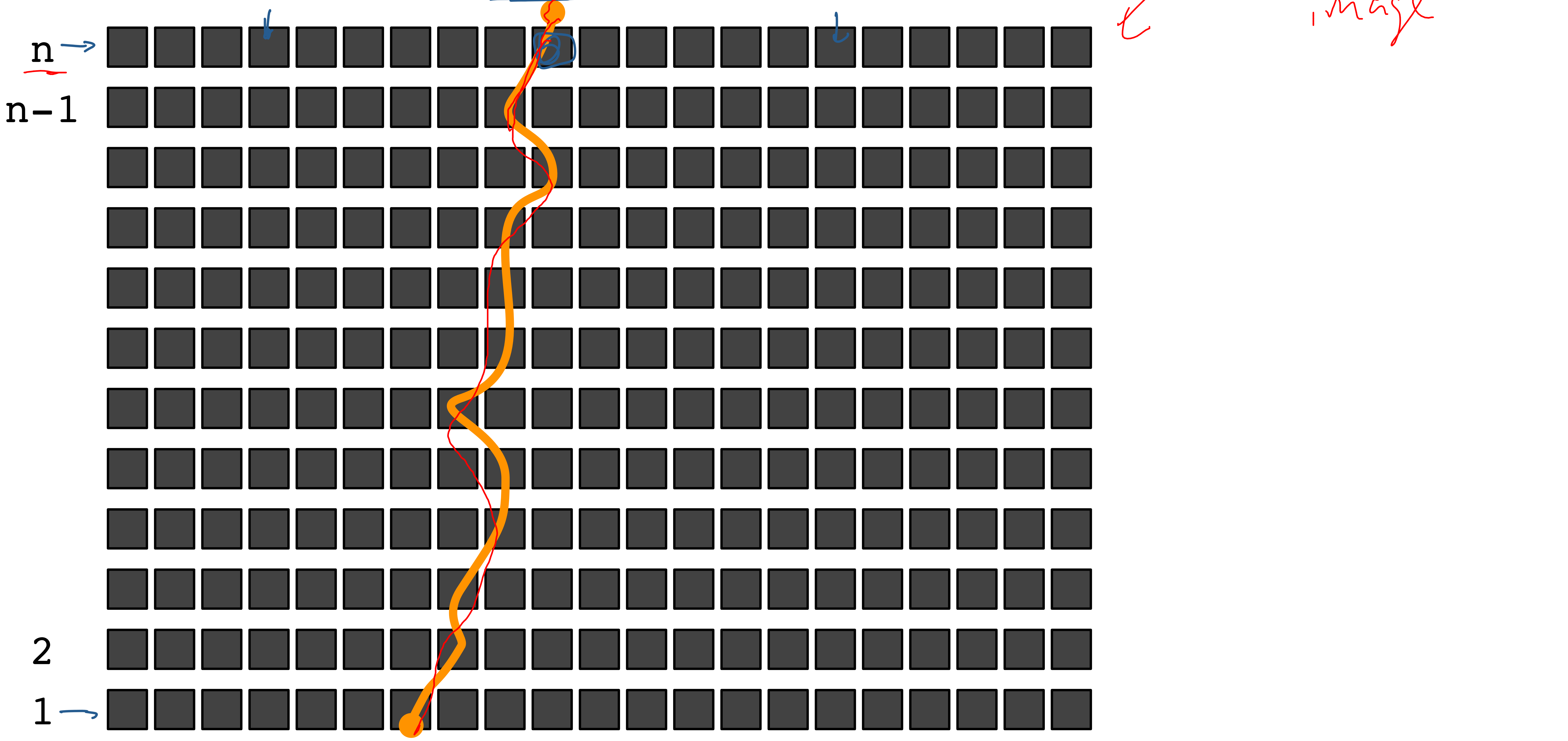
Define a variable:

$S_i(j)$: the total energy of the path that ends at pixel (i,j) and has the least energy definition: $S_n(j)$: lowest energy seam that ends @ (n, j)



definition:

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

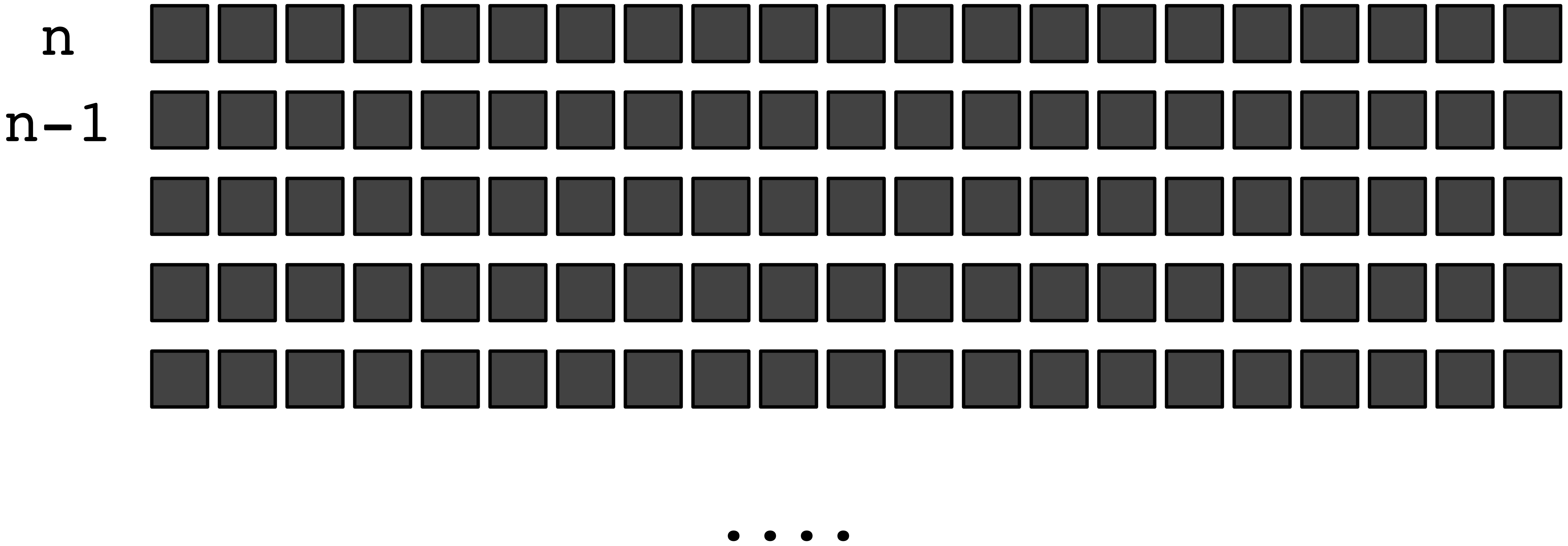


BEST SEAM TO DELETE HAS TO BE THE BEST AMONG

$$S_n(1), S_n(2), \ldots, S_n(m)$$

find min

IDEA: COMPUTE + COMPARE

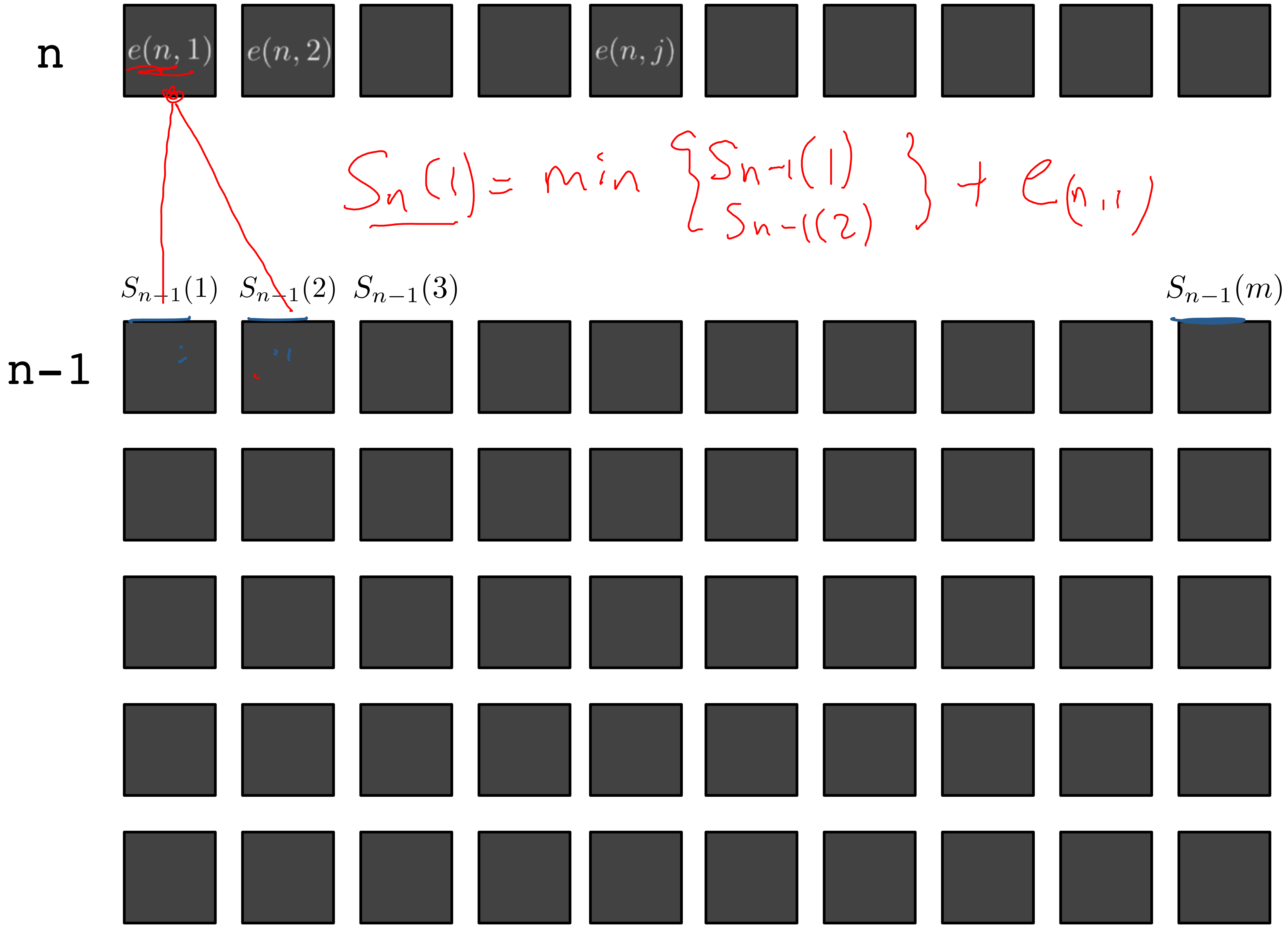


SMALLER PROBLEM APPROACH

How to compute $S_n(j)$??

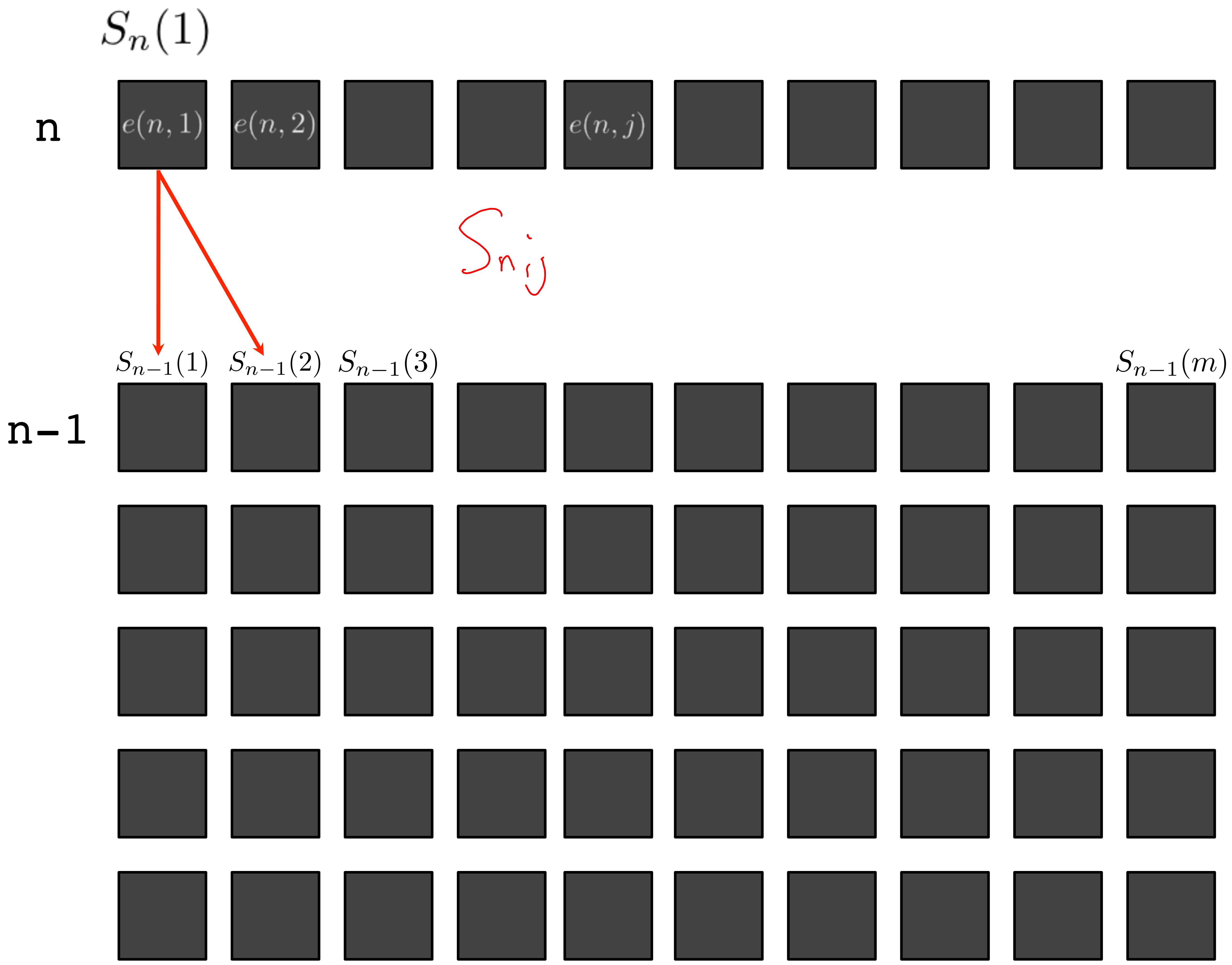
IMAGINE YOU HAVE THE SOLUTION TO THE FIRST $n-1$ ROWS

$S_n(1)$

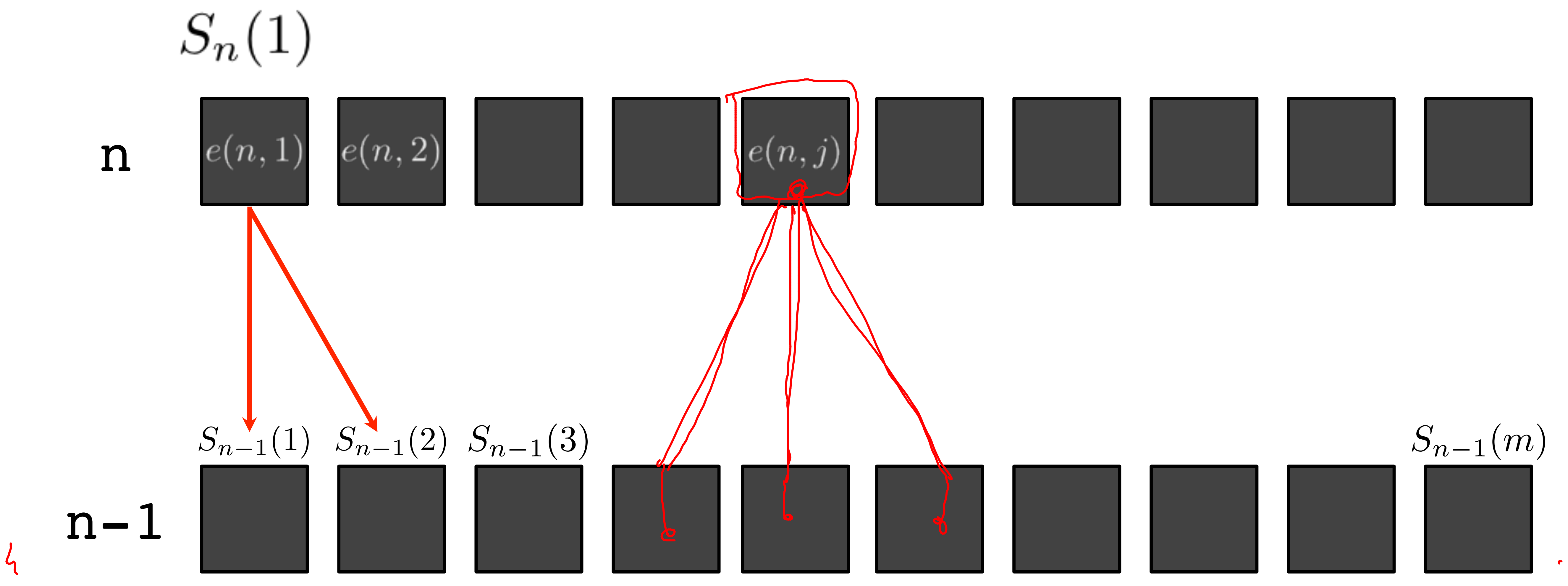


$$S_n(1) = \min \begin{Bmatrix} S_{n-1}(1) \\ S_{n-1}(2) \end{Bmatrix} + e(n,1)$$

← spse you have the solutions for n−1

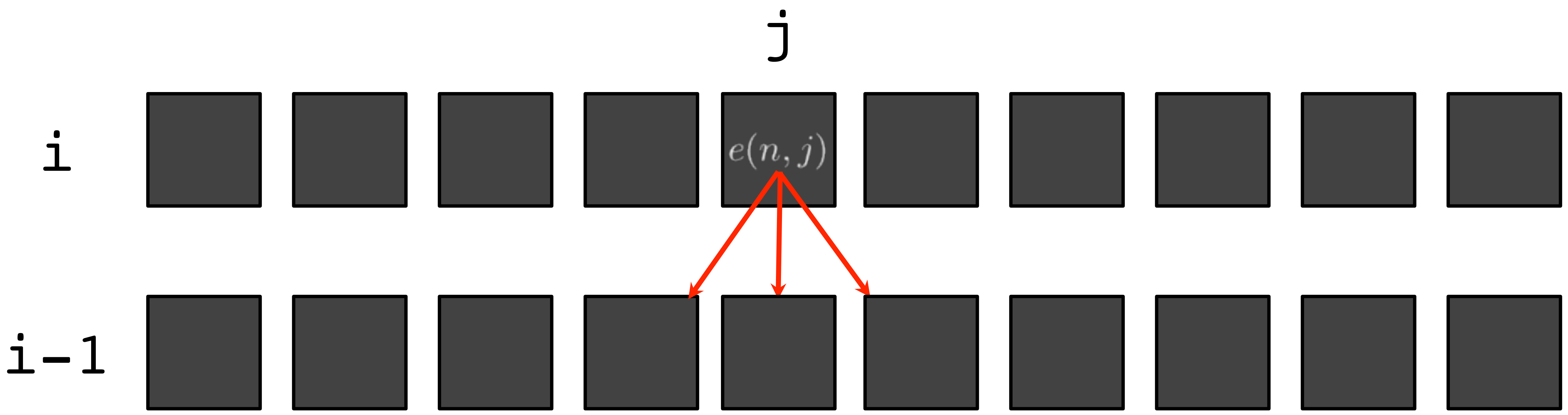


$S_n(1)$
n
$e(n,1)$
$e(n,2)$
$e(n,j)$
$S_{n,j}$
$S_{n-1}(1)$
$S_{n-1}(2)$
$S_{n-1}(3)$
$S_{n-1}(m)$
n-1

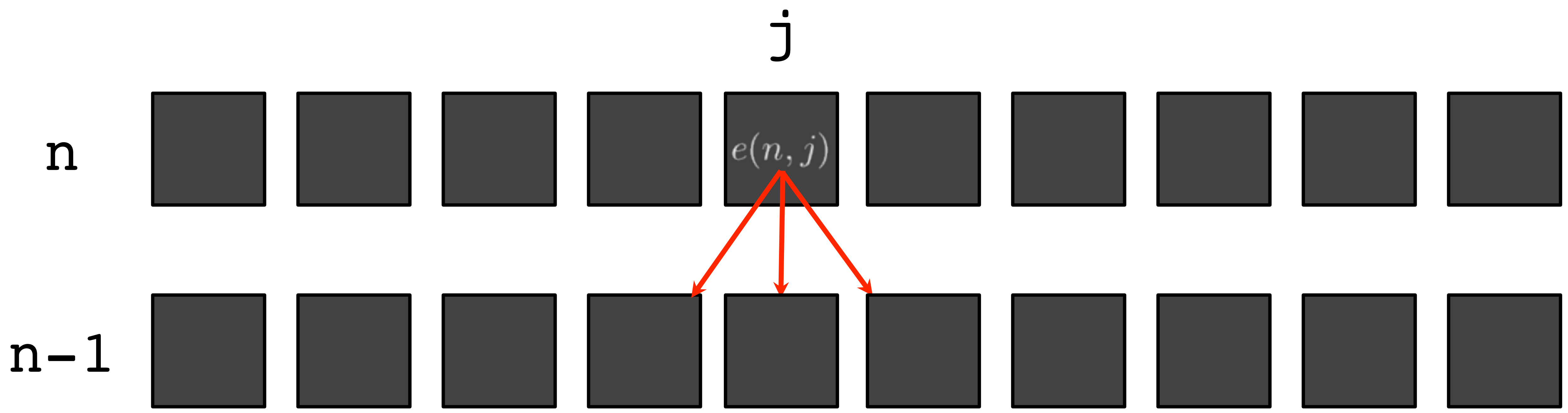


$$S_n(1) = e(n,1) + \min\{S_{n-1}(1), S_{n-1}(2)\}$$

$$S_n(j) = e(n,j) + \min \begin{cases} S_{n-1}(j-1) & \text{or } \infty \text{ if } j=1 \\ S_{n-1}(j) & \\ S_{n-1}(j+1) & \text{or } \infty \text{ if } j=m \end{cases}$$



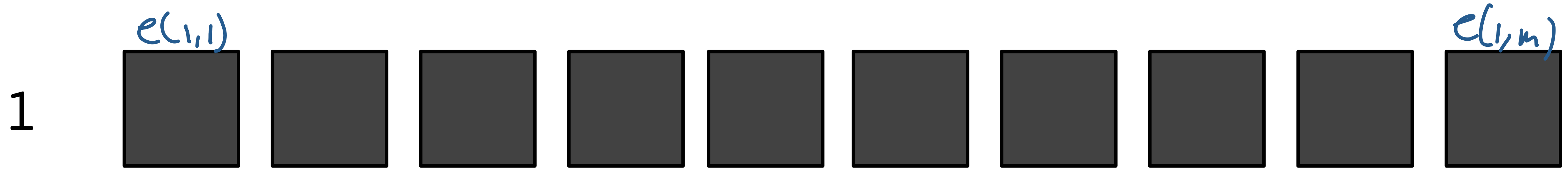
$$S_i(j) =$$



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

ALGORITHM

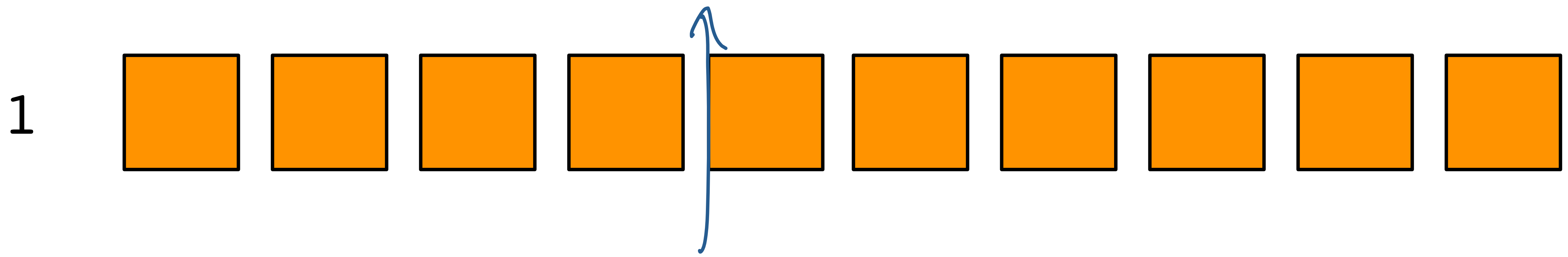
start at bottom of picture



$S_1(i) = e(1, i)$

ALGORITHM

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

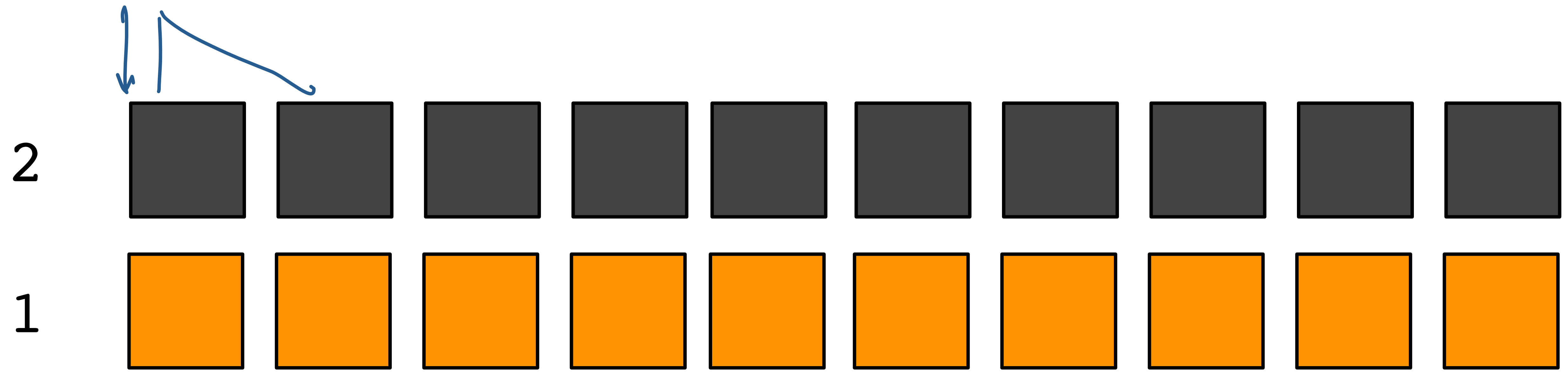


ALGORITHM

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

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

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



ALGORITHM

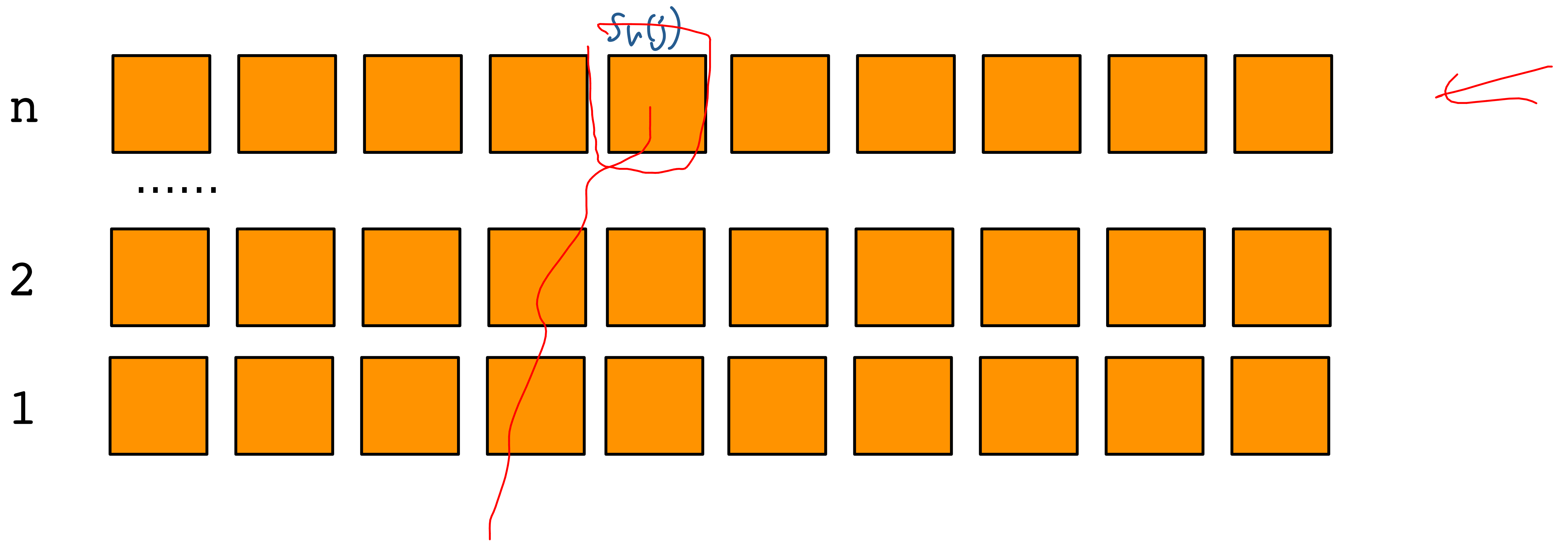
start at bottom of picture. initialize $S_1(i) = e(1, i)$

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

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

anonymous argument

identify the min among $S_n([1\ldots m])$



ALGORITHM

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

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

for j=1...m

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

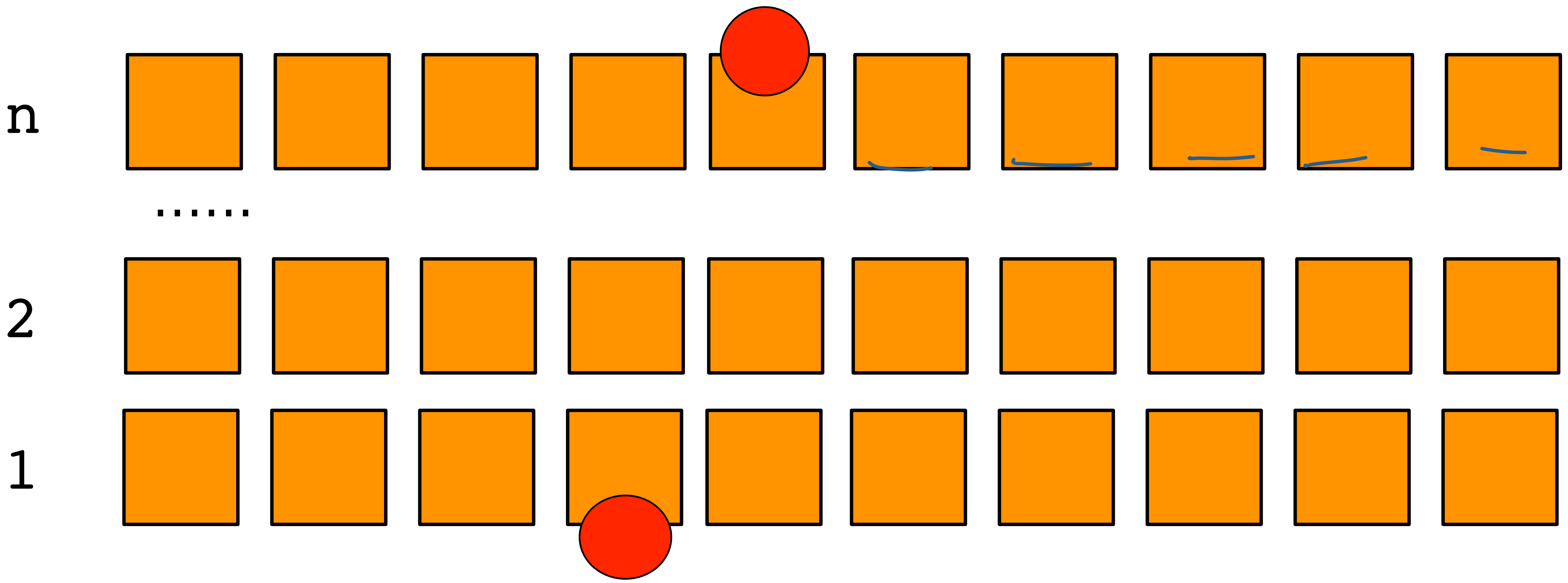
pick best among top row, backtrack.

Running time??

$\Theta(n \cdot m)$

$\Theta(m)$

$\Theta(n \cdot m)$



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 $S_1(i) = e(1, i)$

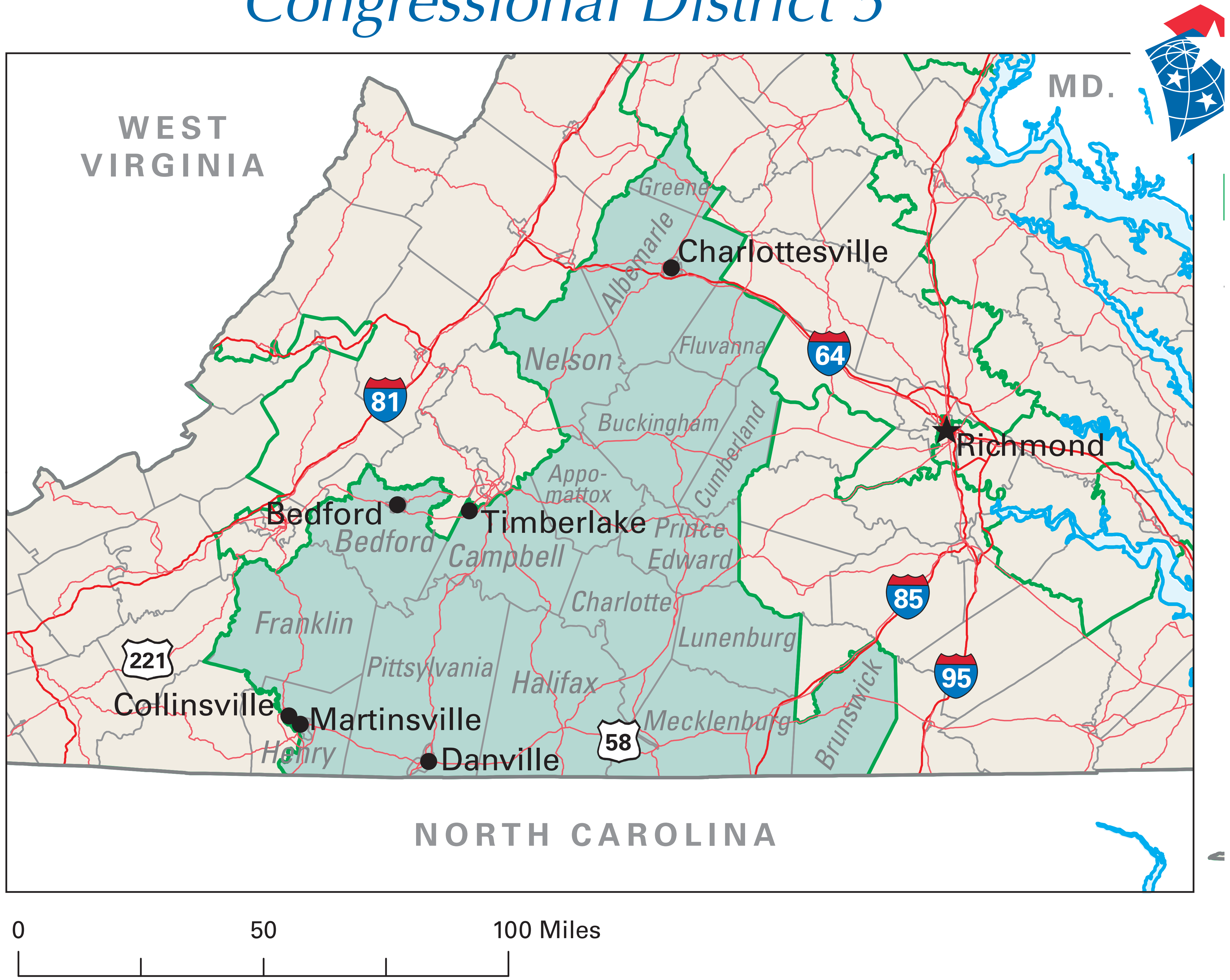
for `i=2,n` use formula to compute $S_{i+1}(\cdot)$

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

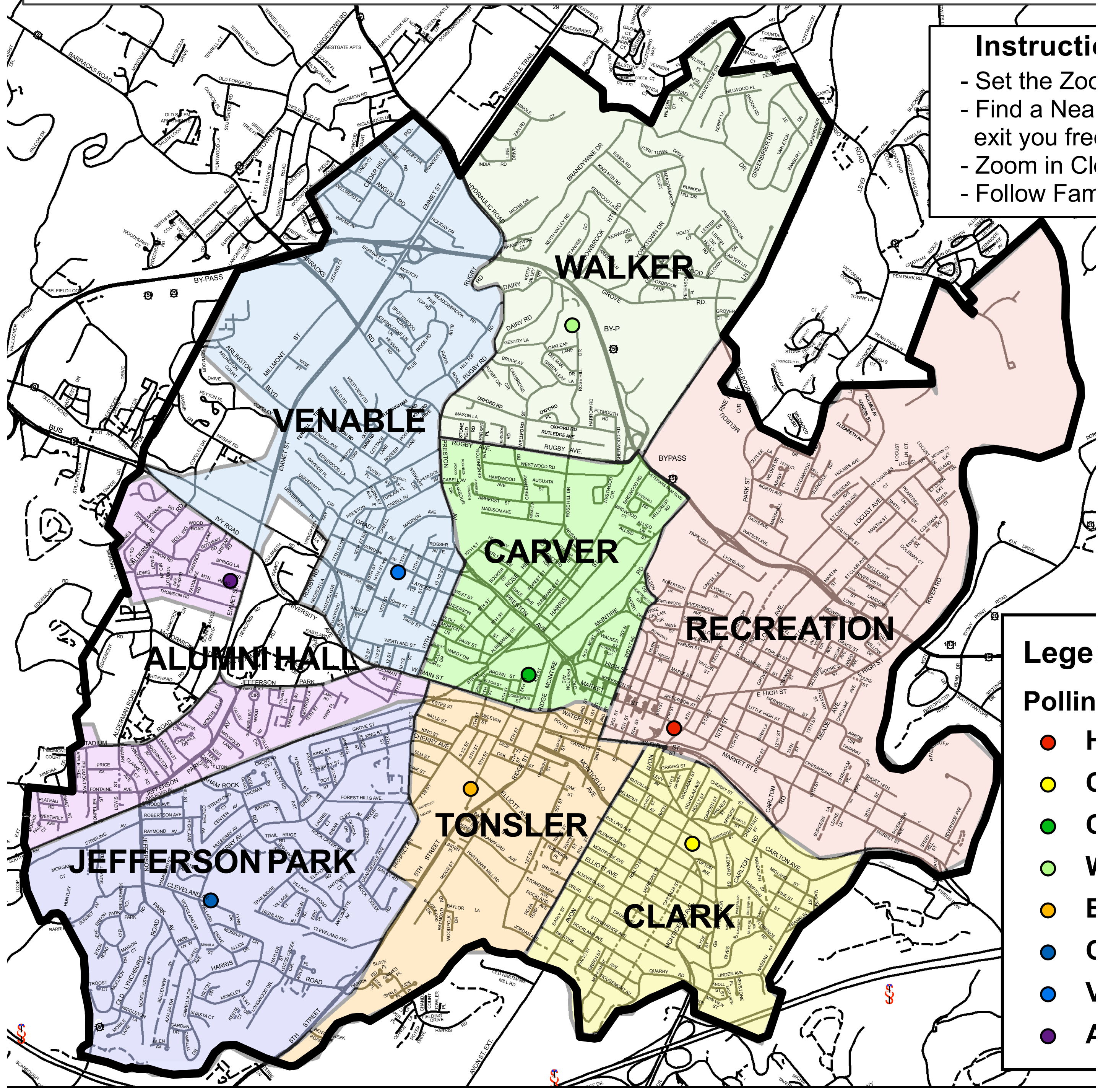
pick best among top row, backtrack.

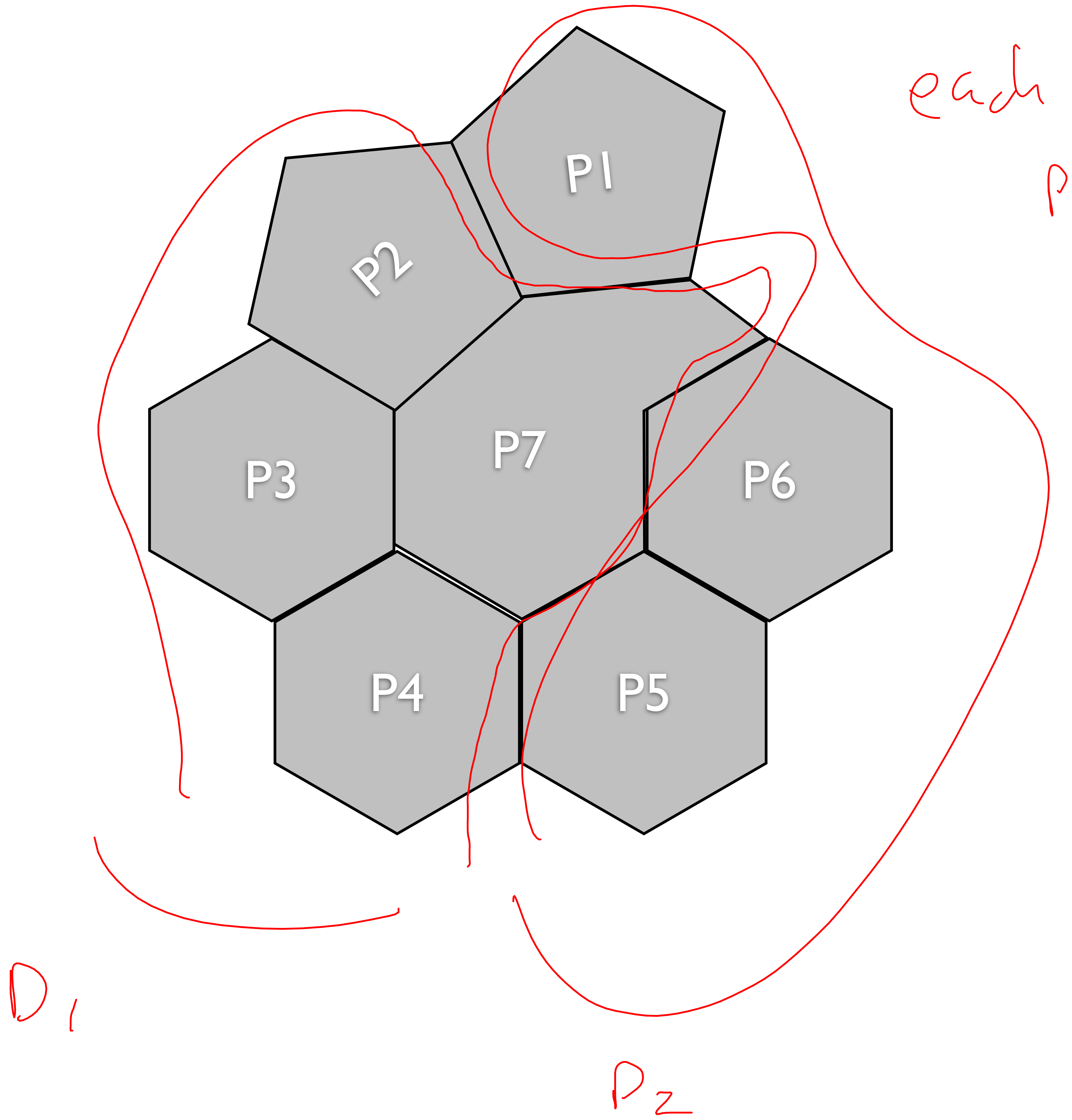
Gerrymander

Congressional District 5



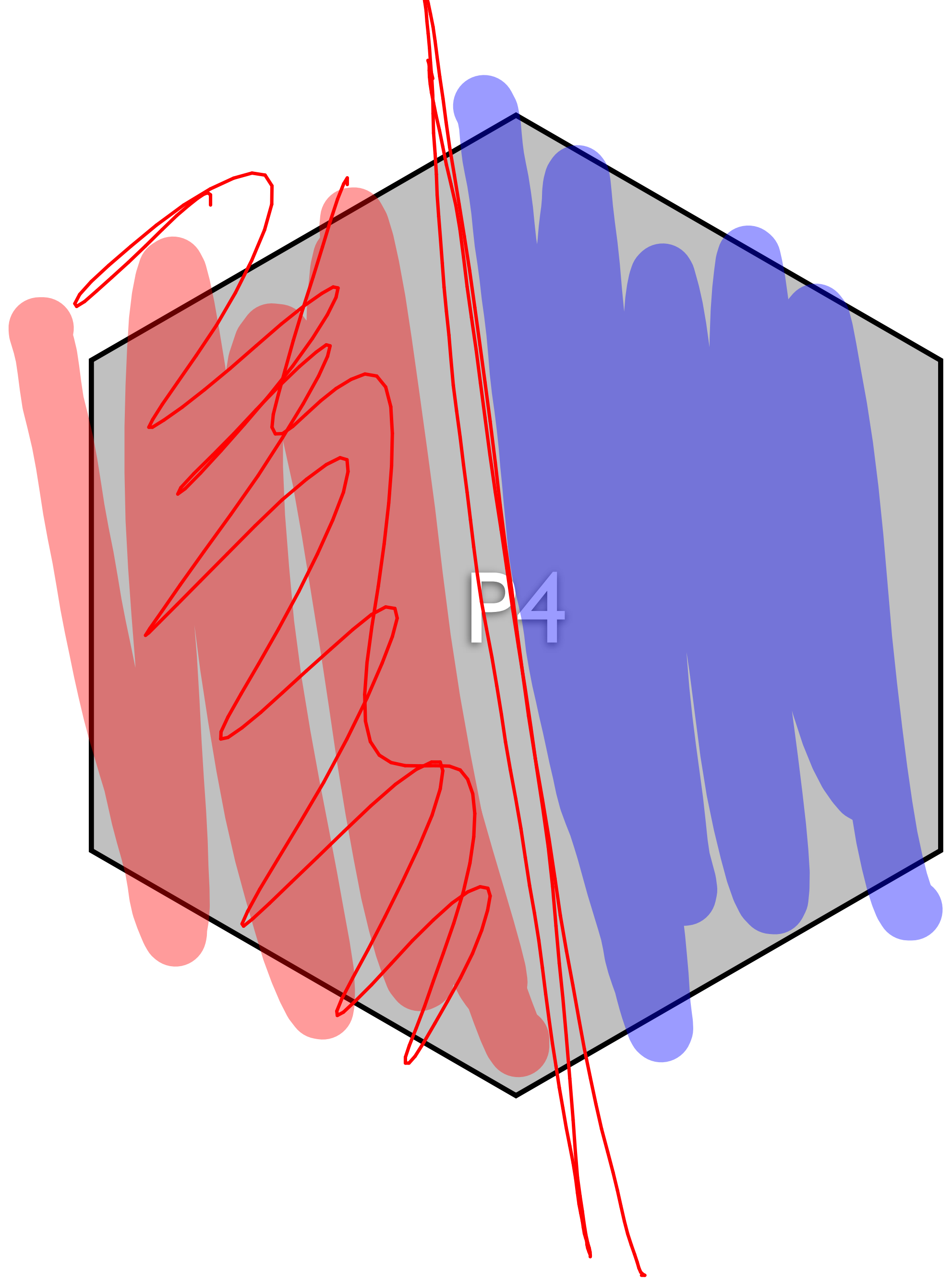
Map of Charlottesville Precincts and





each precint has M people in it.

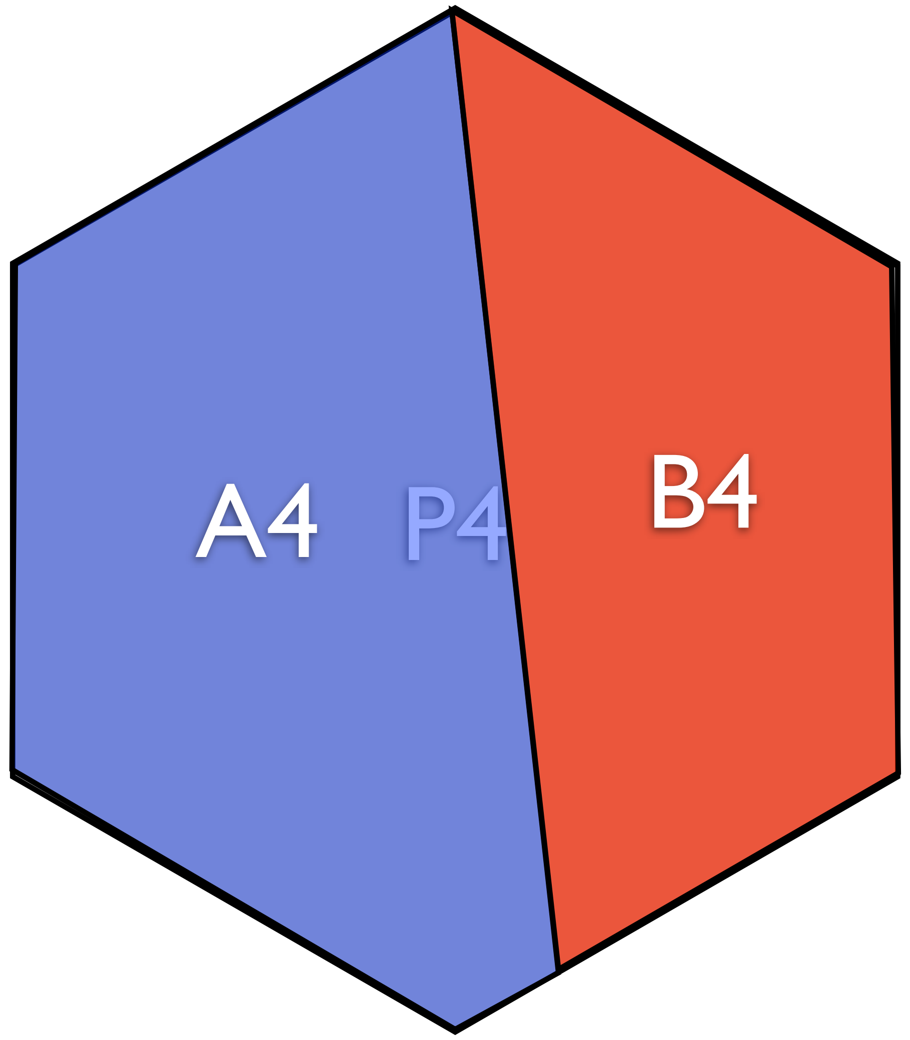
Consider a small state that has only 2 districts



Some fraction of precint 4 that votes for party A.

$A_4 \rightarrow \frac{\text{fraction for } A}{\#\text{ of votes}}$

$B_4 = \frac{M - A_4}{}$


A4
P4
B4

GERRYMANDER PROBLEM

given: M, $A_1, A_2, \ldots A_n$ ← n is even

output: D_1, D_2 2 districts

① $|D_1| = |D_2|$ i.e. same # of precincts

② $A(D_1) > \frac{m \cdot n}{4}$

$A(D_2) > \frac{m \cdot n}{4}$

i.e. A has a majority in both districts

GERRYMANDER PROBLEM

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

output: D_1, D_2

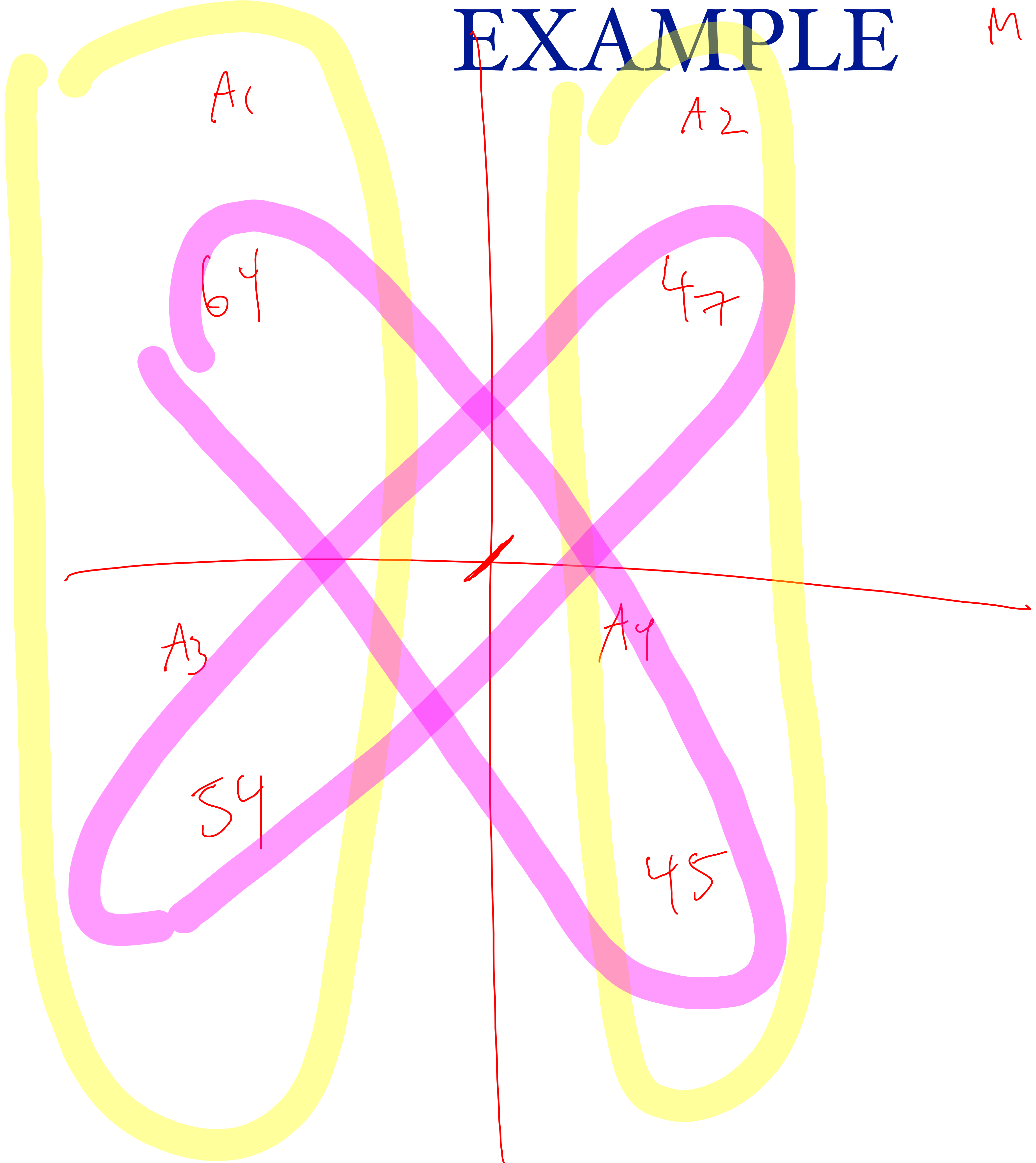
such that $|D_1| = |D_2|$

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

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

or “failure” if no such solution is possible

EXAMPLE



$M = 100$

$A_1, A_3 \to 119$ votes, majority in D,

$A_2, A_4 \to 92$ votes, minority !!

$A_1, A_4 = 109$

$A_2, A_3 = 101$

$S_{4,2,101,108} =$ FALSE

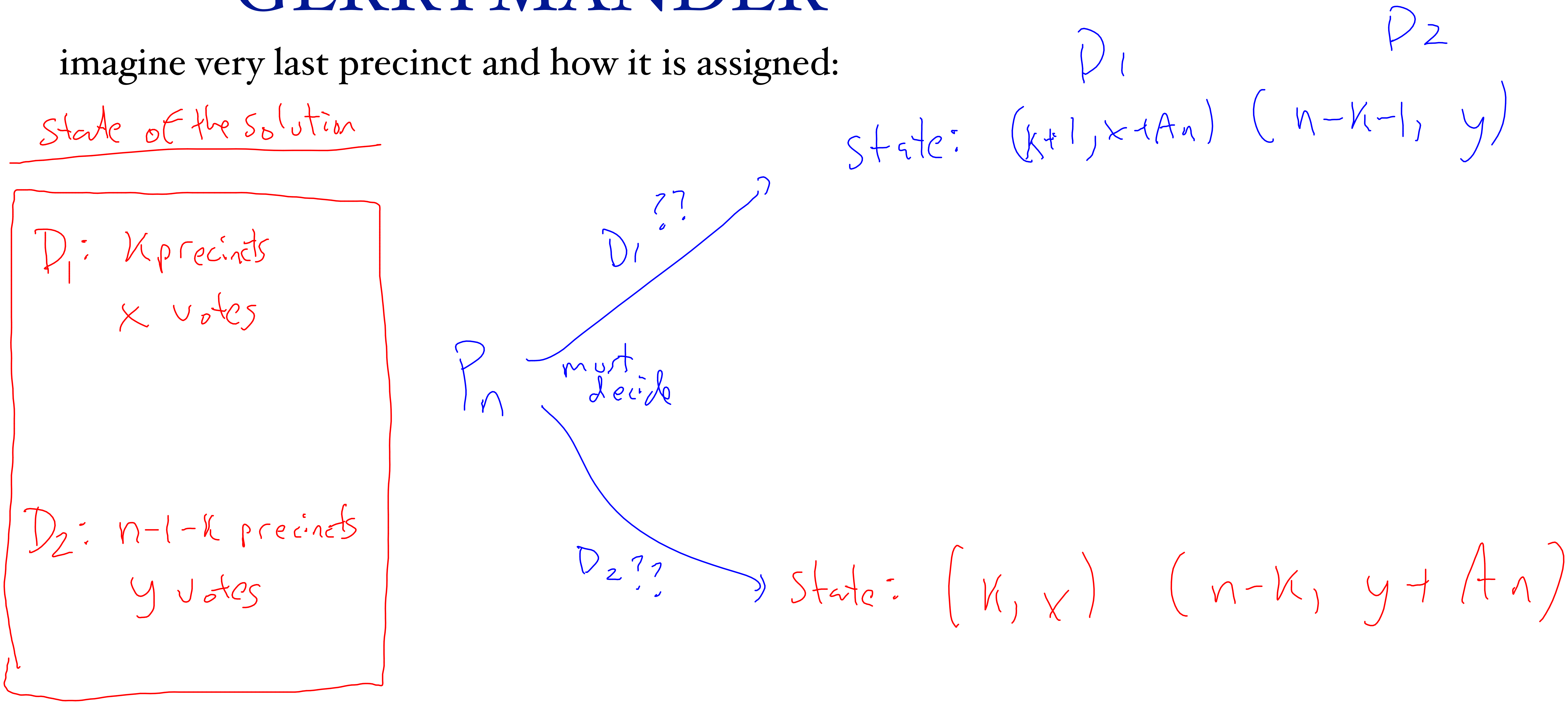
BUT $S_{4,2,101,109} =$ TRUE

THE TECHNIQUE

1d 2d 4d

GERRYMANDER

imagine very last precinct and how it is assigned:



GERRYMANDER

$S_{j,k,x,y} =$ true or false:

true if $\exists$ an assignment of the first j precincts such that

$|D_1| = k$ and

$A(D_1) = x$ and

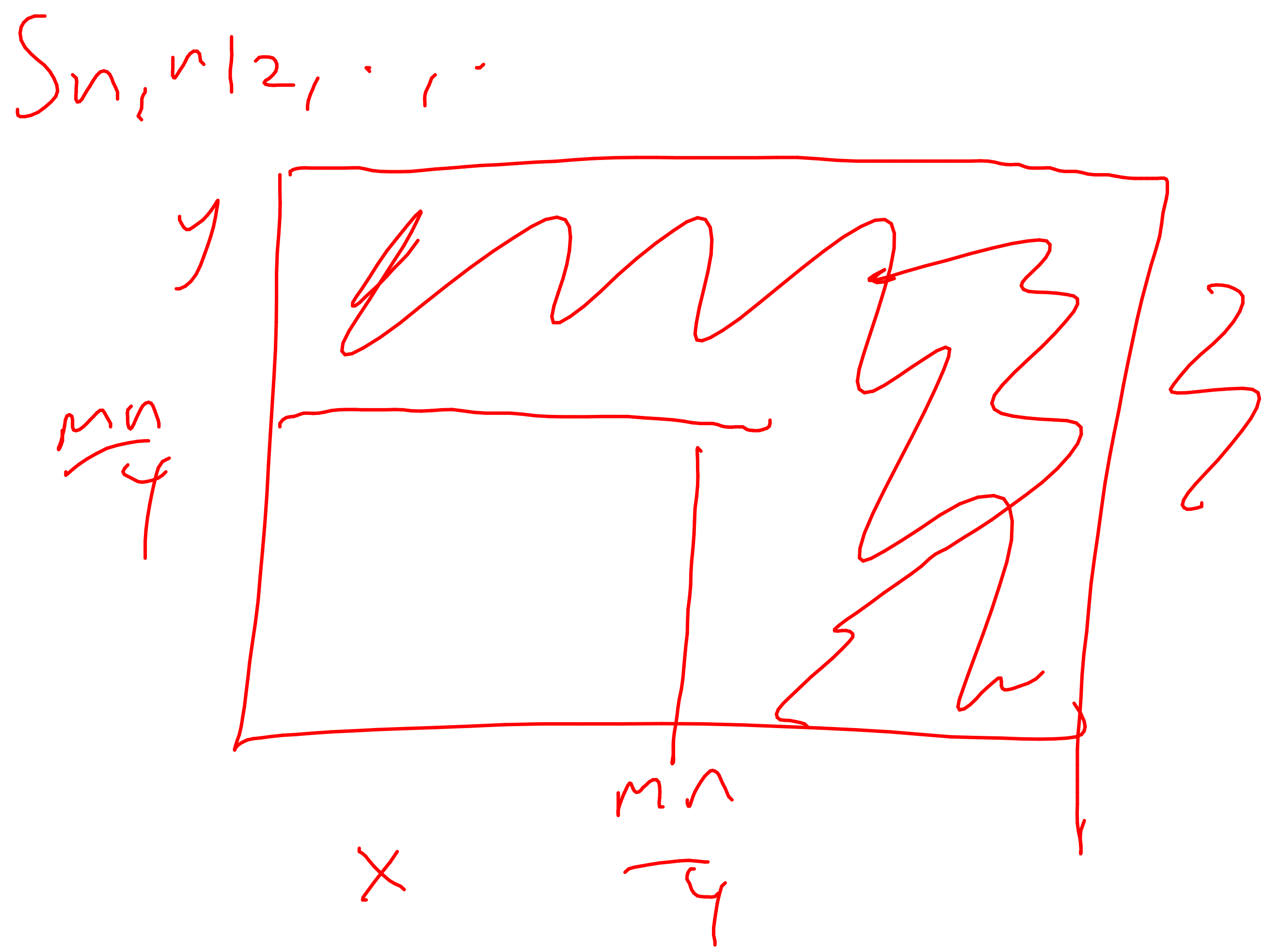
$A(D_2) = y$

what if

$S_{n,\,n/2,\,\frac{mn}{4}+1,\,\frac{mn+1}{4}}$ = ?? true ??

if there any $S_{n,n/2,x,y}$ that is true when

$x > \frac{mn}{4}$ $y > \frac{mn}{4}$



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

$S_{j,k,x,y} = S_{j-1,k-1,x-A_n,y}$ OR $S_{j-1,k,x,y-A_n}$

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

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