L10

4102

Sep 24 2009

abhi shelat

log cutting -

matrix chain –

typesetting – auxiliary variable $S_{i,j}$

seam carving -

gerrymander - relevant to a problem on the homework.

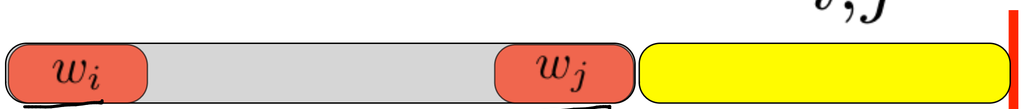
Typesetting

From the algorithm
given in lecture
to java code.

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.

WHAT ABOUT $S_{i,j}$

$S_{i,j}$



slack when line starts with w_i and ends w_j

WHAT ABOUT $S_{i,j}$

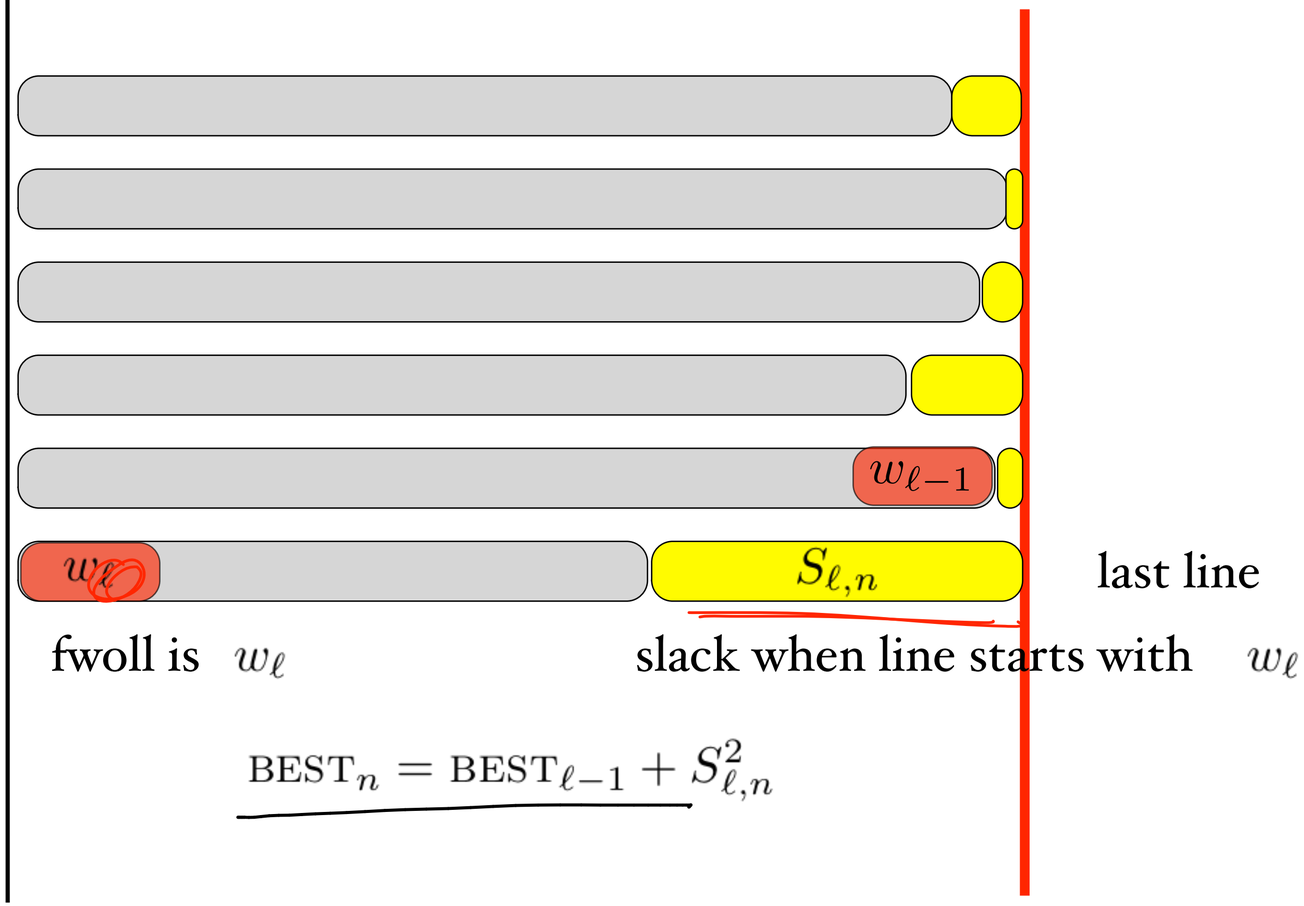
$S_{i,j}$

slack when line starts with w_i and ends w_j

$$S_{i,j} = \begin{cases} M - c_i & j = i \\ S_{i,j-1} - c_j - 1 & i < j \end{cases}$$

∞ if $S_{i,j} < 0$

IMAGINE OPTIMAL SOLUTION



fwoll is w_ℓ slack when line starts with w_ℓ

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

WHICH WORD IS FWOLL?

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

TYPESETTING ALGORITHM

1 make table $S_{i,j} = \begin{cases} M - c_i & j = i \\ S_{i,j-1} - c_j - 1 & i < j \end{cases}$

∞ if $S_{i,j} < 0$

2 compute $\text{BEST}_0, \text{BEST}_1, \ldots, \text{BEST}_n$

using

$\text{BEST}_0 = 0$

$\text{BEST}_n = \min \begin{cases} \text{BEST}_0 + S_{1,n}^2 \\ \text{BEST}_1 + S_{2,n}^2 \\ \cdots \\ \text{BEST}_{n-1} + S_{n,n}^2 \end{cases}$

3 keep track of choices & backtrack
to determine line breaks

Seam Carving

PROBLEM: REDUCE IMAGE

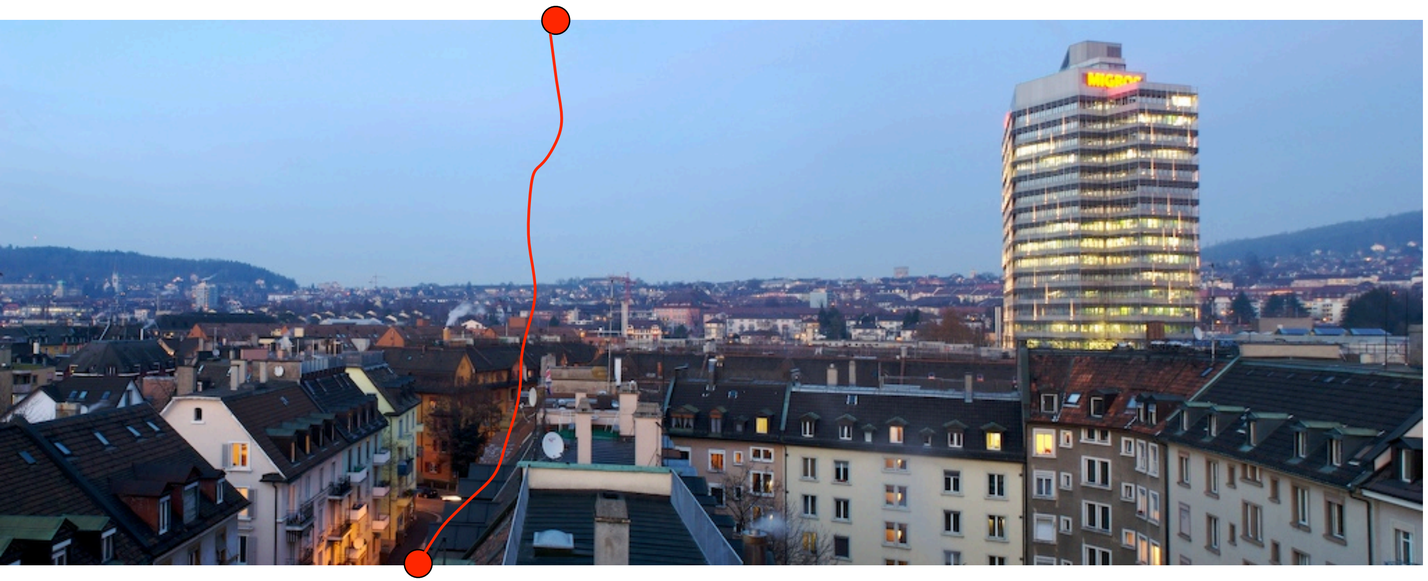
scaling: distortion
deleting column: distortion
delete the most invisible seam


Retarget col se...
Image Resizing
Shai Avidan
Mitsubishi Electric Research Lab
Ariel Shamir
The interdisciplinary Center & MERL

DEMO?

http://rsizr.com/

WHICH SEAM TO DELETE?

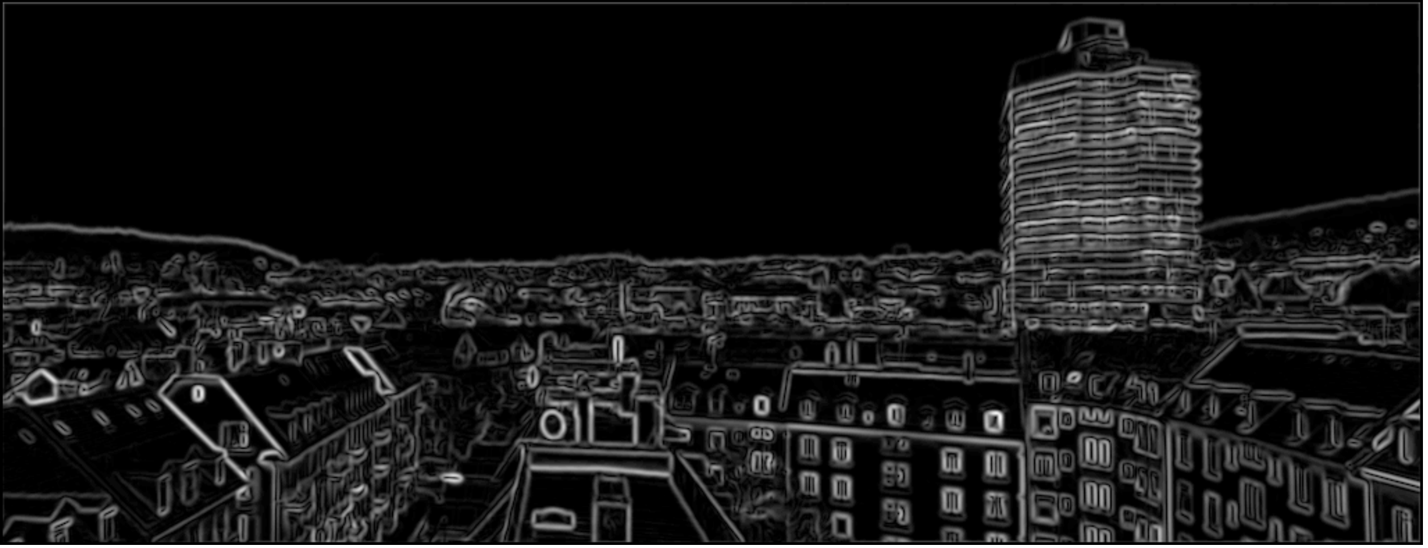


ENERGY OF AN IMAGE

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

"magnitude of gradient at a pixel"

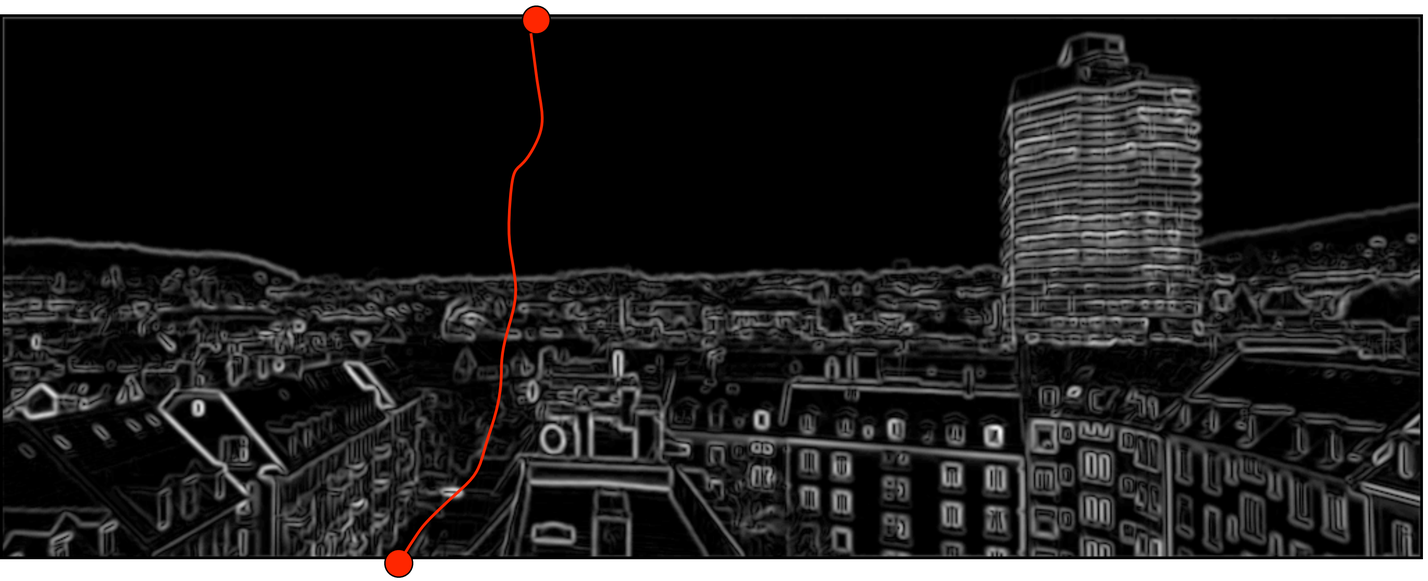
$$\frac{\partial}{\partial x} I_{x,y} = I_{x-1,y} - I_{x+1,y}$$



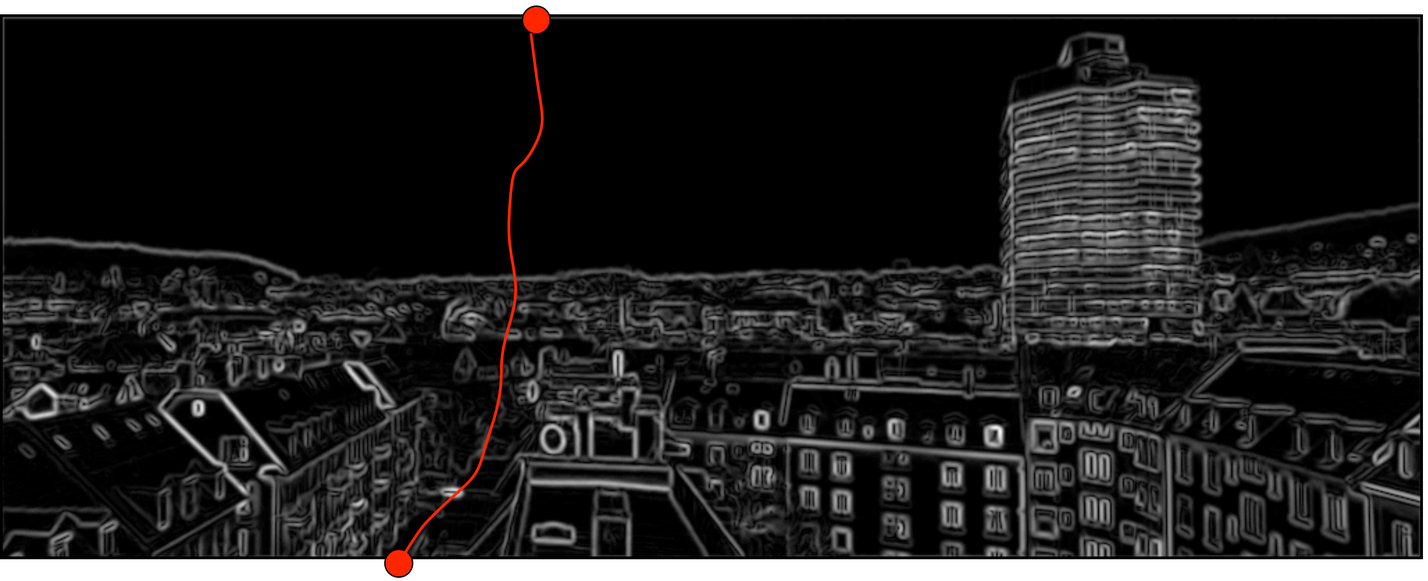
energy of sample image

thanks to Jason Lawrence for gradient software

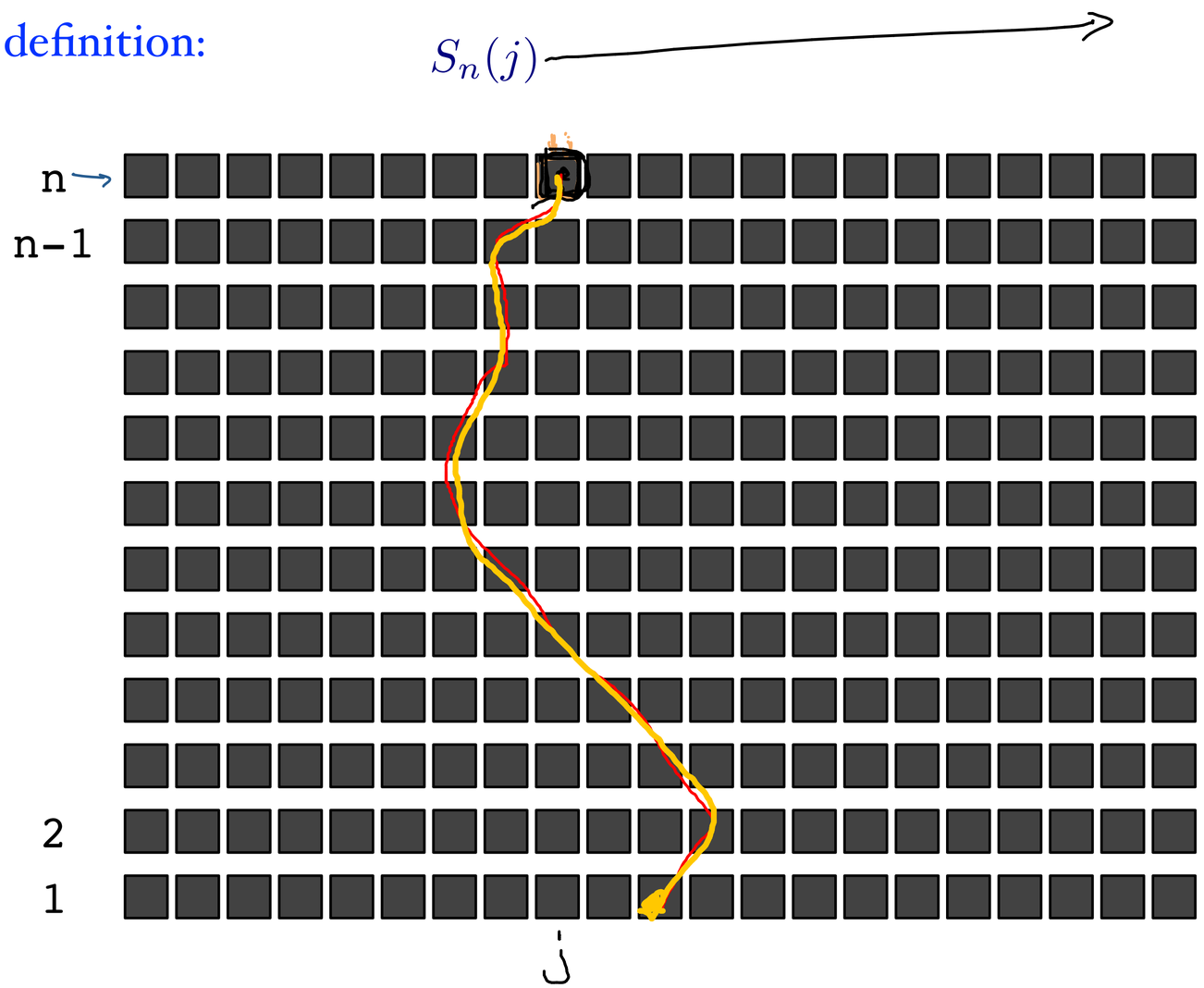
BEST SEAM HAS LOWEST ENERGY



FINDING LOWEST ENERGY SEAM?



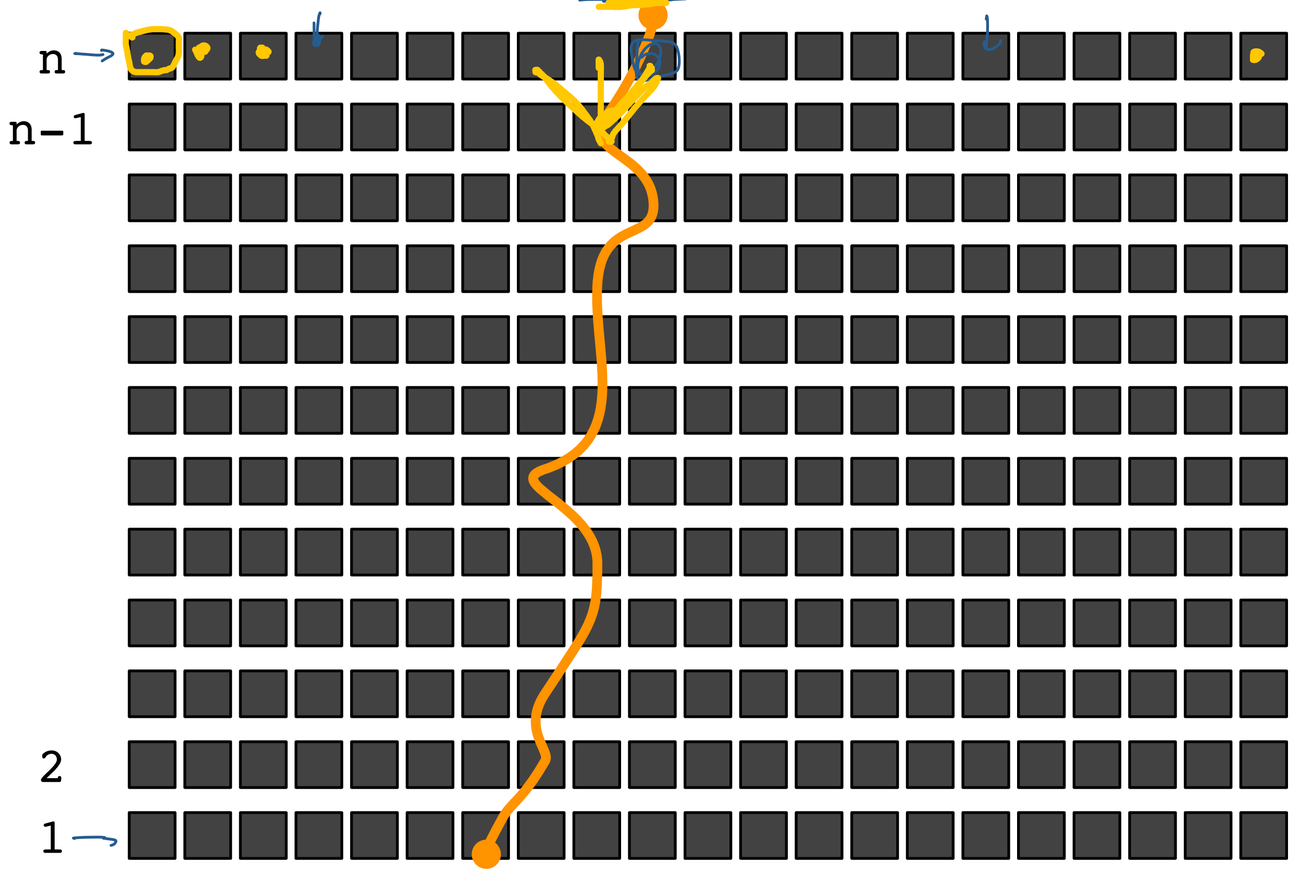
definition:



$S_n(j)$ = total energy of the best seam that ends @ pixel (n, j).

definition:

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

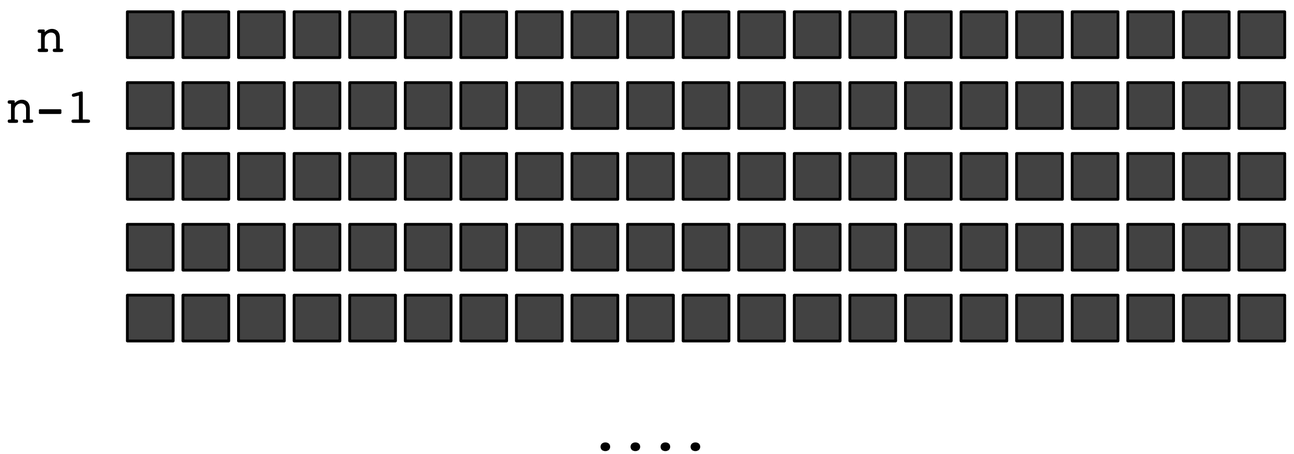


top row.

BEST SEAM TO DELETE HAS TO BE THE BEST AMONG

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

IDEA: COMPUTE + COMPARE

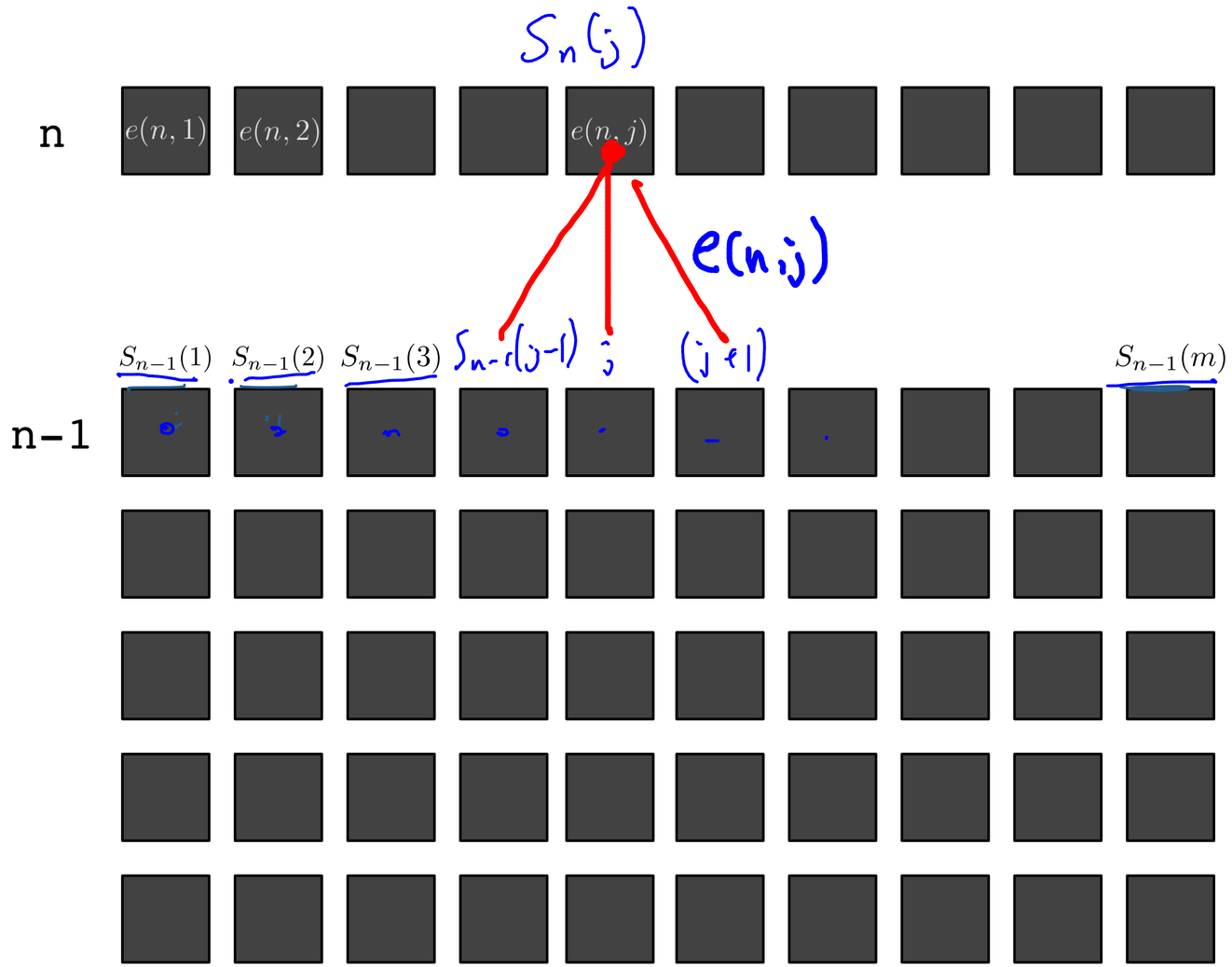


1. compute $S_n(1), \ldots S_n(n)$
2. find smallest among this set
3. Backtrack to recover the seam
4. Remove this SEAM

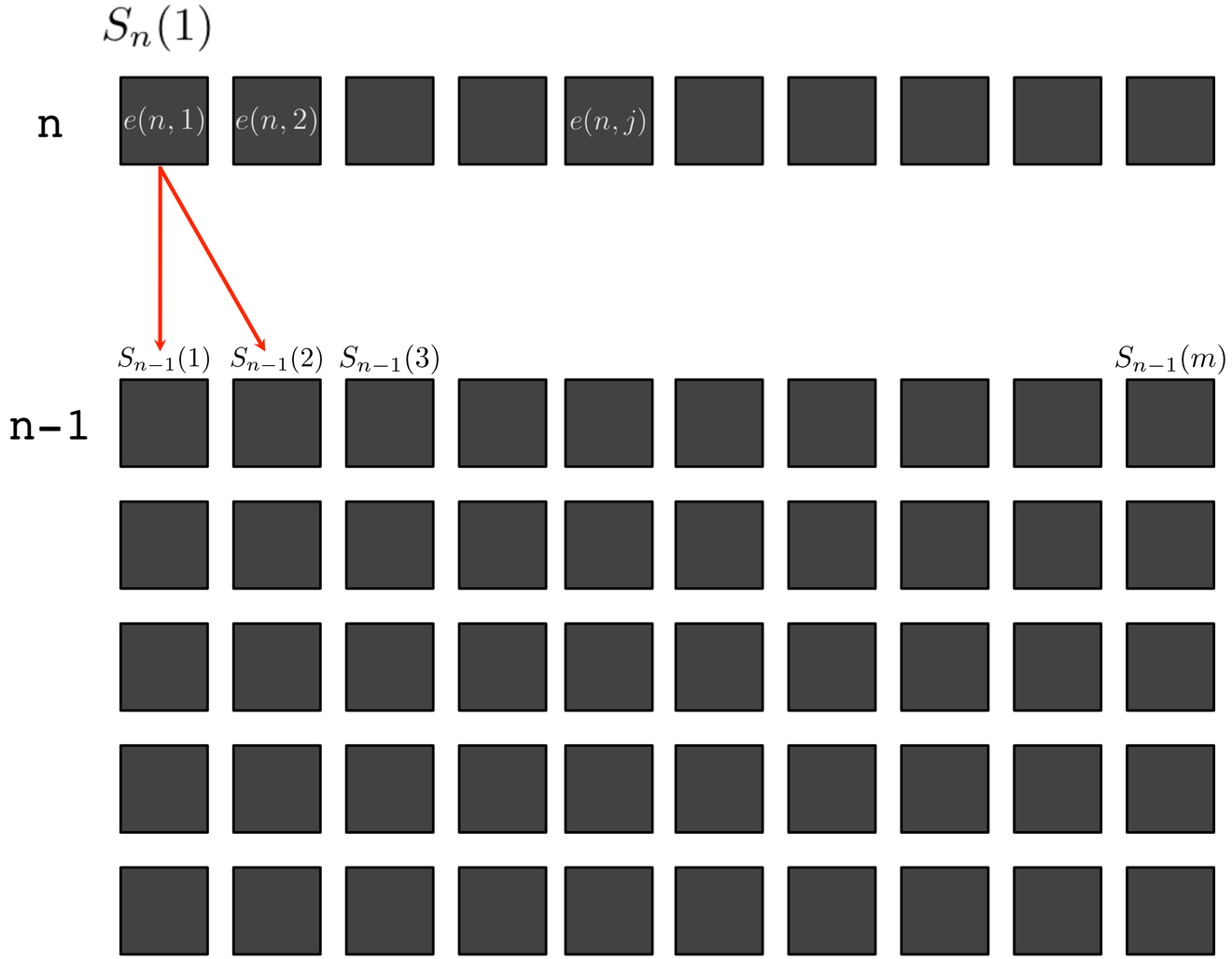
Repeat.

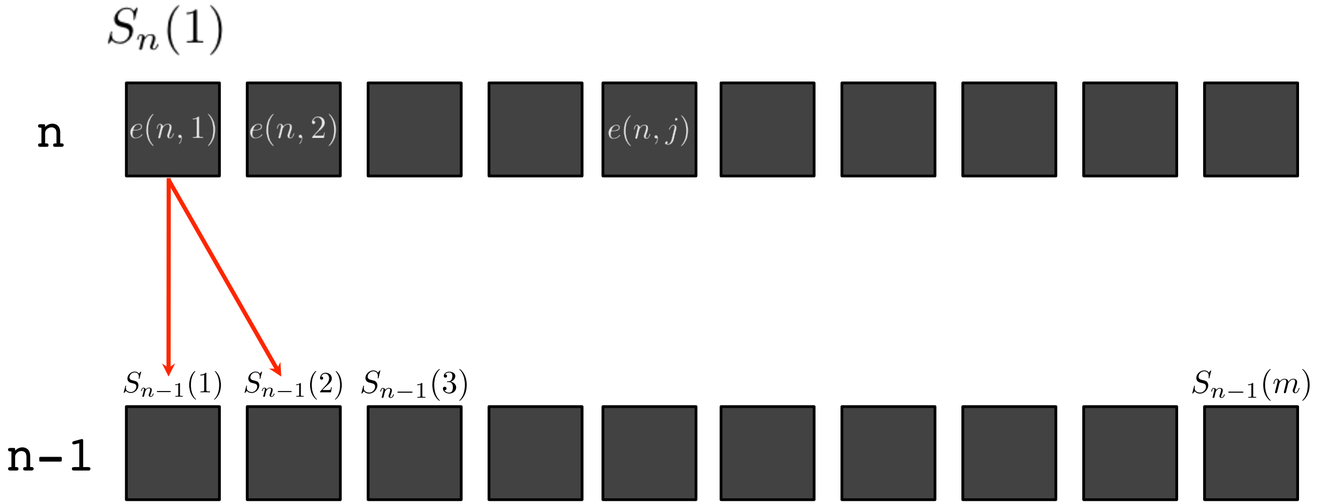
SMALLER PROBLEM APPROACH

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

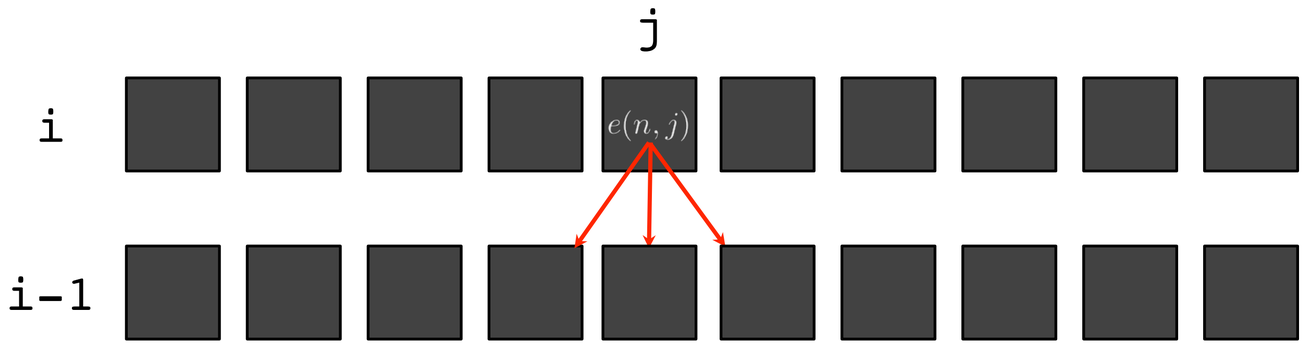


$$S_n(j) = \min \begin{Bmatrix} S_{n-1}(j-1) \\ S_{n-1}(j) \\ S_{n-1}(j+1) \end{Bmatrix} + e(n,j)$$

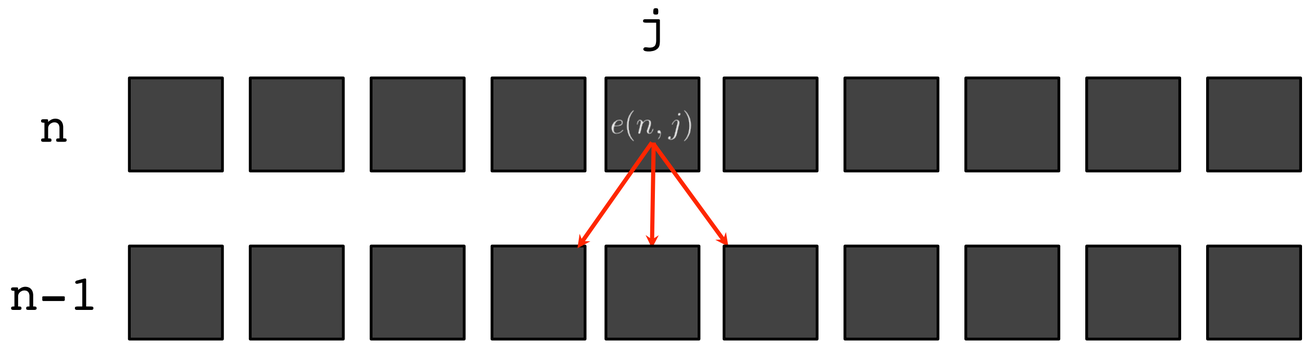

$S_n(1)$
n
$e(n,1)$
$e(n,2)$
$e(n,j)$
$S_{n-1}(1)$
$S_{n-1}(2)$
$S_{n-1}(3)$
$S_{n-1}(m)$
n-1



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



$$S_i(j) =$$



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

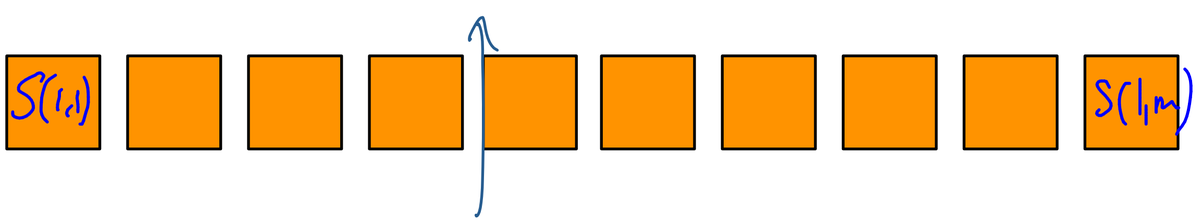
ALGORITHM

start at bottom of picture

ALGORITHM

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

1

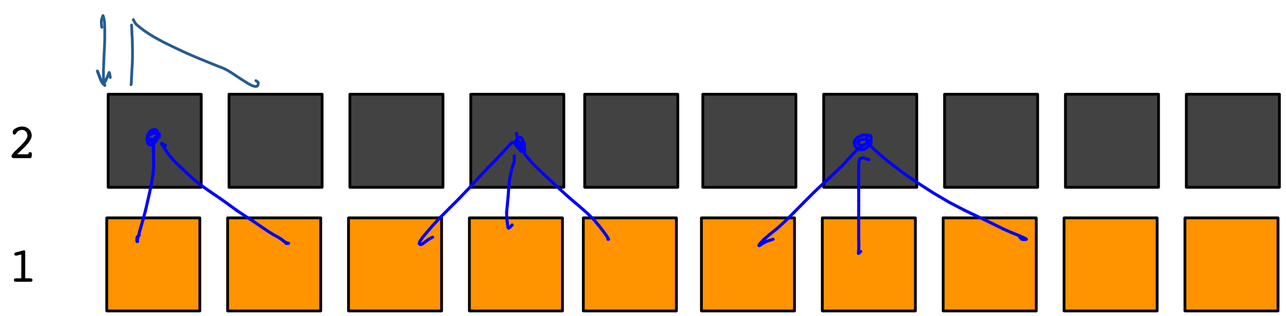


ALGORITHM

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

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

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



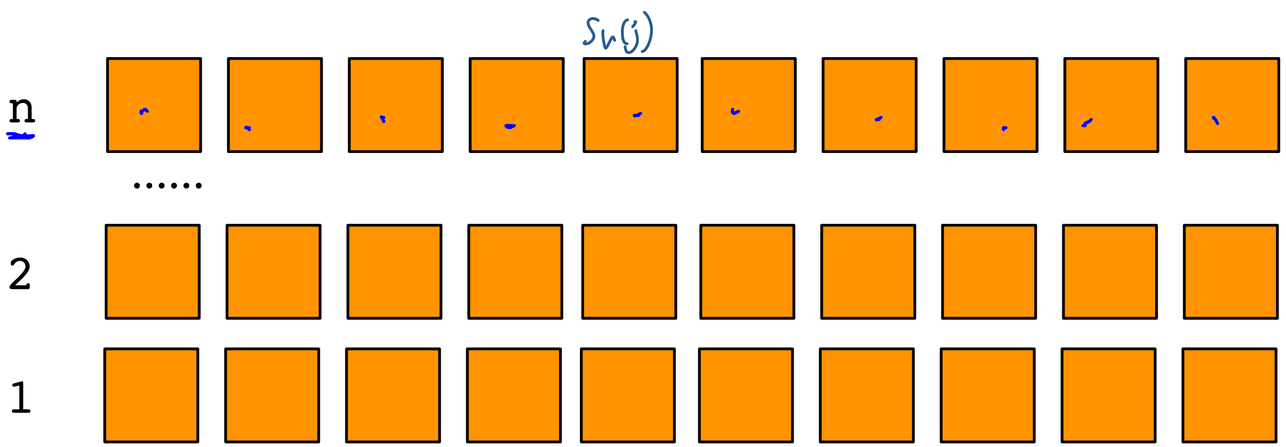
ALGORITHM

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

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

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

FIND MIN AMONG THE TOP ROW.



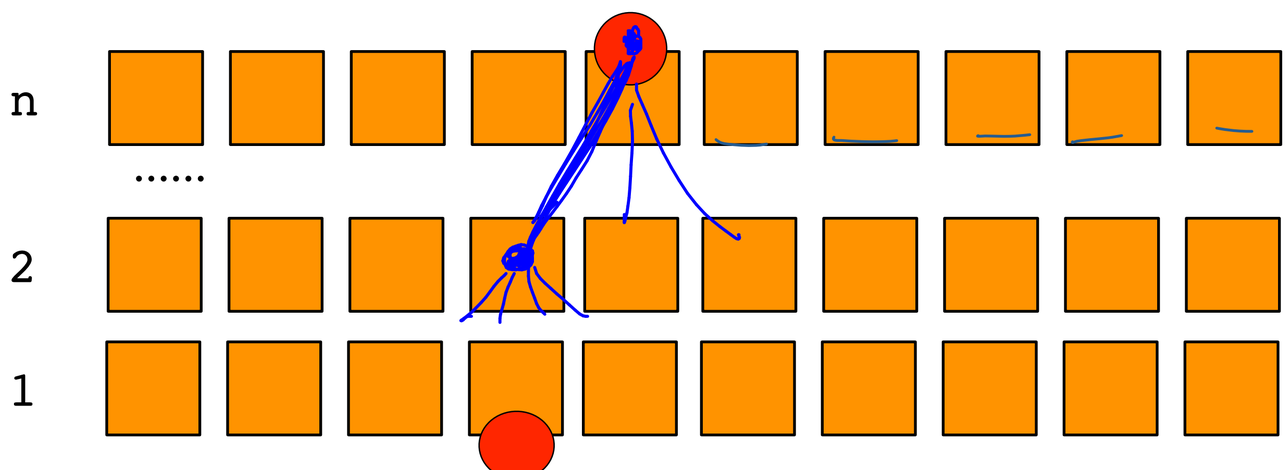
ALGORITHM

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 $S_{i+1}(\cdot)$

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

$\Theta(n \cdot m)$

pick best among top row, backtrack.

$\Theta(n+m)$

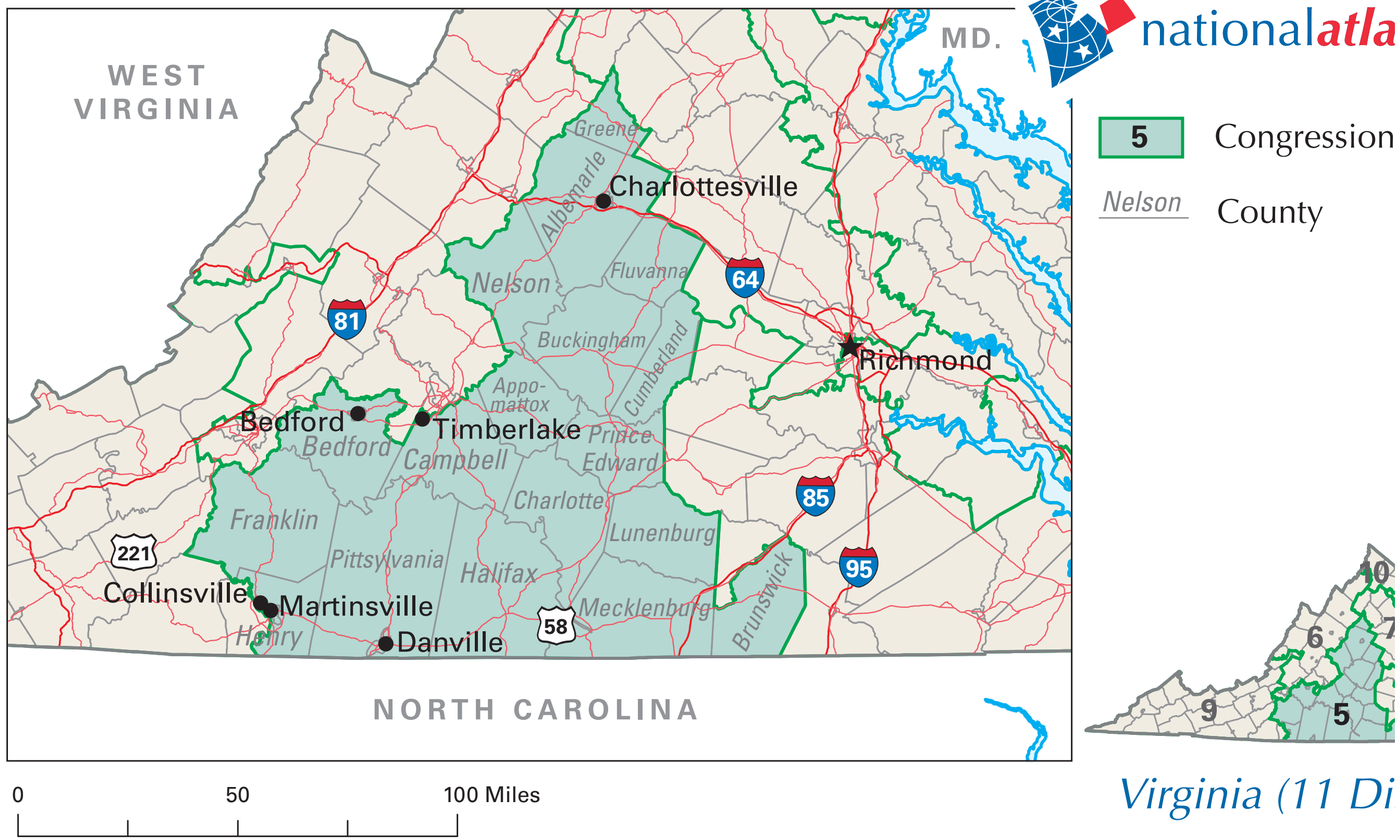
loop runs n - values per loop: m

overall $\Theta(n \cdot m)$

$\Rightarrow$ LINEAR IN THE SIZE of the image

Gerrymander

Congressional District 5



0 50 100 Miles

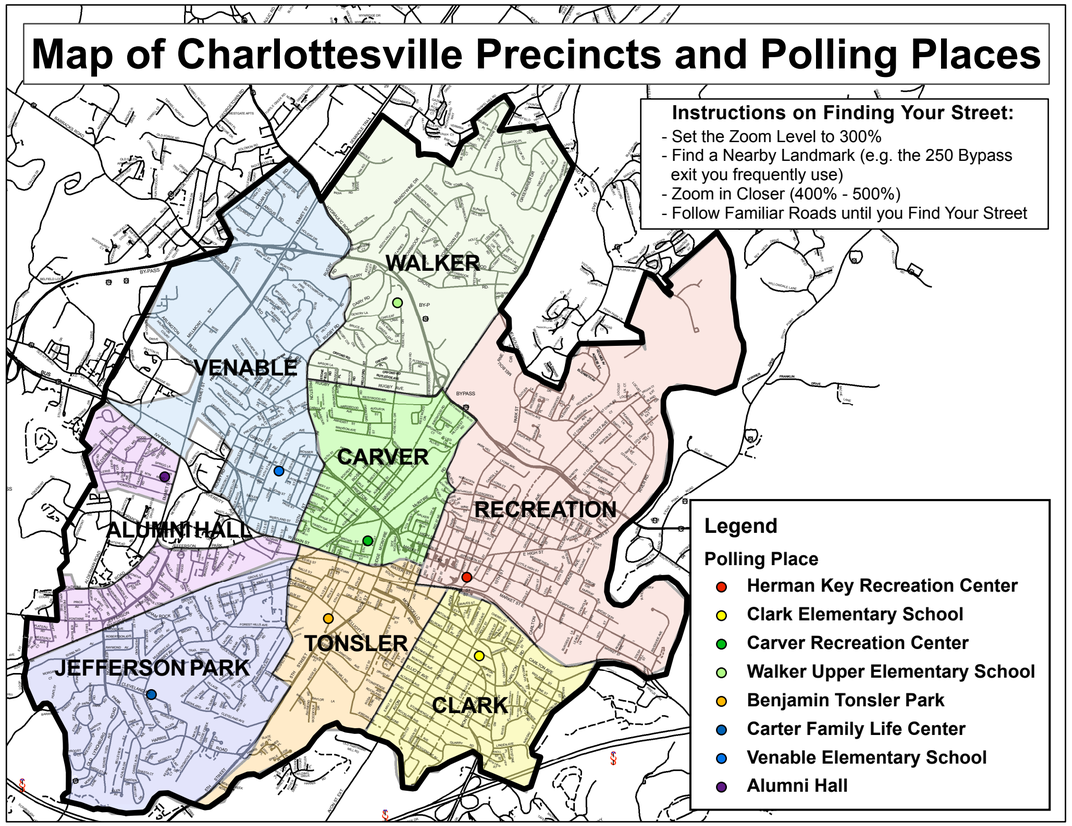
nationalatlas

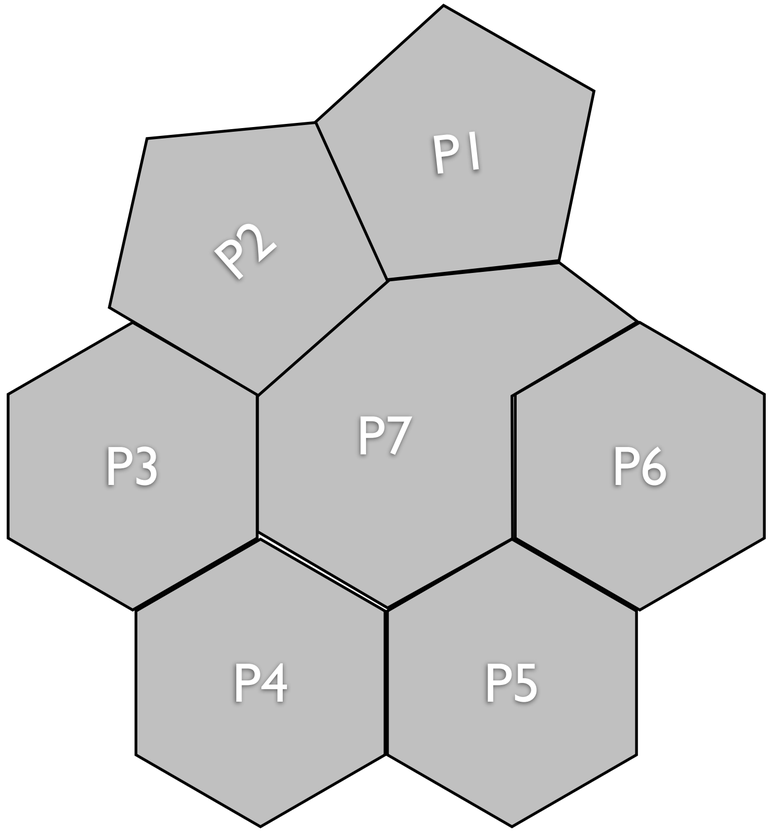
5 Congression

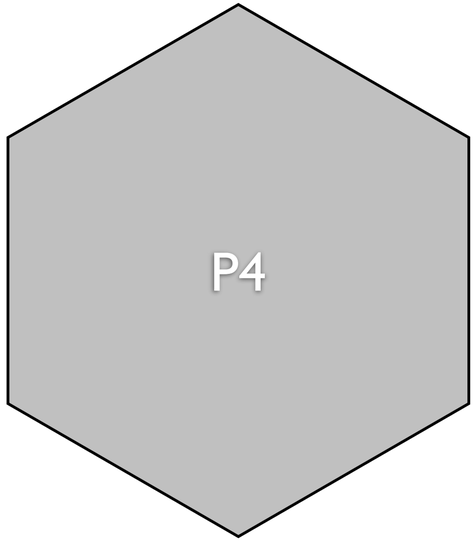
Nelson County

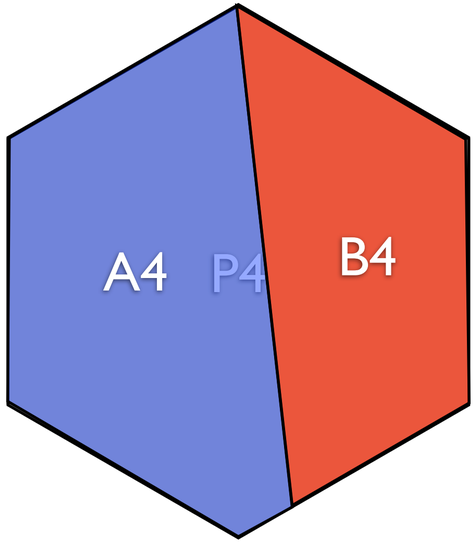
Virginia (11 Di

Map of Charlottesville Precincts and Polling Places




P1
P2
P3
P7
P6
P4
P5


P4


A4
P4
B4

GERRYMANDER PROBLEM

given: $P_1\ P_2\ \dots\ P_n$ M: # of people in each precinct

$A_1\ A_2\ \dots\ A_n$ (assume all P_i have same size)

output: $D_1\ D_2$ 2 districts such that both favor candidate A.

↑ set of precincts.

AND $|D_1| = |D_2|$ Lets assume that n is even.

GERRYMANDER PROBLEM

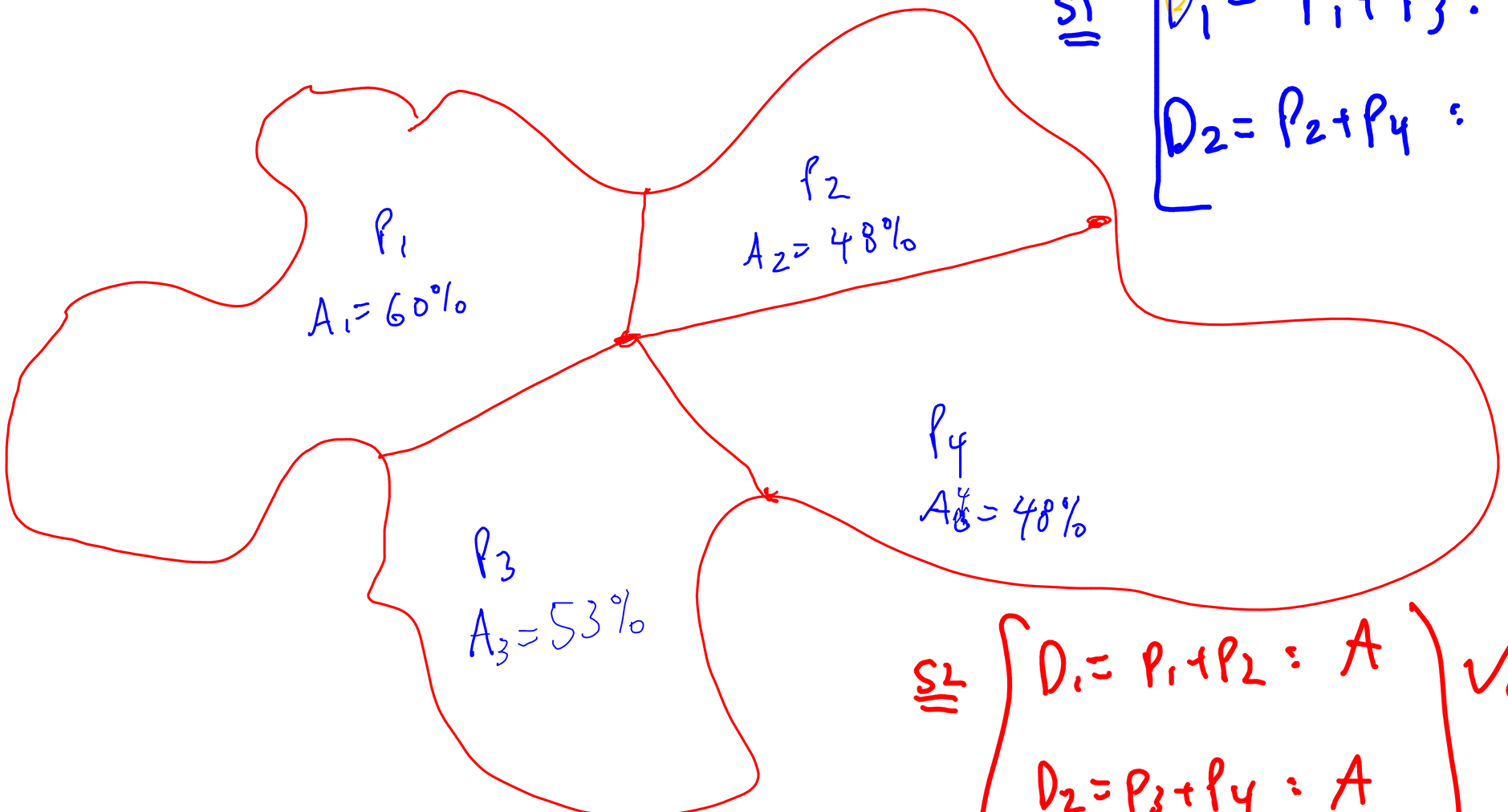
given: $P_1, P_2, \ldots, P_n \quad m$

$A_1, A_2, \ldots, A_n$ n is even

output: D_1, D_2

EXAMPLE

$n = 100$.



S1

$D_1 = P_1 + P_3 : A$

$D_2 = P_2 + P_4 : B$

S2

$D_1 = P_1 + P_2 : A$

$D_2 = P_3 + P_4 : A$

THE NEW TECHNIQUE

(1) Define the right sub-problem

1D Log 2D matrix 4D $S_{i,j,k,x,y}$

(a) ↱

(b) → Sub-problem is A BOOLEAN value

TRUE OR FALSE

(2) JUST LIKE SEAM CARVING,

RE-SCAN MATRIX TO FIND solution.

GERRYMANDER

imagine very last precinct and how it is assigned:

ASSIGN TO P_1

ASSIGN TO P_2

D_1' will have $|D_1|+1$ precincts

$D_1 + A_n$ – voters for A

D_2' will have $|D_2|$

" " D_2 – voters for A

D_1' will have $|D_1|$ – precincts

D_1 – A voters

D_2' will have $|D_2|+1$ – "

$D_2 + A_n$ = "

$S_{j,n,x,y}$

GERRYMANDER

$S_{j,k,x,y} =$ $\exists$ an assignment of j precincts in which

$|D_1| = k$ &

D_1 has x A-voters

D_2 has y A-voters.

TRUE or FALSE.

GERRYMANDER

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

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

↑ ASSIGN P_j to D_1

↑ ASSIGN P_j to D_2

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

GERRYMANDER(P,A,m)

initialize array S[0,0,0,0]

for $j = 1 \ldots n$

$k = 1 \ldots \frac{n}{2}$

$x = 0 \ldots n \cdot m$

$y = 0 \ldots n \cdot m$

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

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

GERRYMANDER(P,A,m)

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