6.006
Rivest
L21.1
11/18/08

Admin:

Readings: 15.1 - 15.4

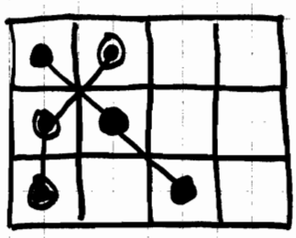
Outline: (Dynamic Programming 3/4)

- ☐ Image compression by seam-carving
- ☐ Matrix chain multiplication

6.006
Rivest
L21.2
11/18/08

Seams in a grid

Given an m x n array:



m = 3 rows
n = 4 columns

a (vertical) seam is a selection of one element per row, s.t. elements in adjacent rows are adjacent vertically or diagonally. (Horizontal seam has one per column.)

How many seams are there?

$S(m,n)$ = # (vertical) seams in an m x n grid

Consider moving king from top row to bottom row. Always have at most 3 moves from row i to row $i+1$, and always have at least 2 moves. There are n places to start off, so

$$n \cdot 2^{m-1} \leq S(m,n) \leq n \cdot 3^{m-1}$$

(exponentially many!)

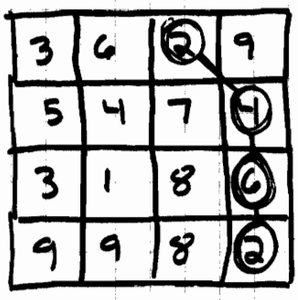
6.006
Rivest
L21.3
11/18/08

Cheap seams

Suppose each cell has <u>cost</u> c_{ij} for including it in seam.

Cost of seam = sum of costs of cells it contains.

Example:



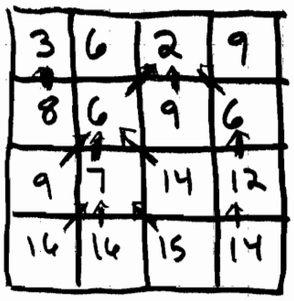
Cheapest seam has
cost $2+4+6+2=14$

Computing cheapest seam using D.P.

Given costs c_{ij} for $0 \le i < m$, $0 \le j < n$ (0-origin indexing)

one subproblem for each cell (i,j): what is cost of cheapest seam running from 0th row down to cell (i,j) (including (i,j))

Let V_{ij} be that cost:



$$V(0,j) = C(0,j)$$

$$V(i,j) = C(i,j) + \min(V(i-1,j-1), V(i-1,j), V(i-1,j+1))$$

($=\infty$ if outside array)

$B(i,j)$ = "↖" or "↑" or "↗" depending (backptrs)

Take min along bottom row.

6.006
Rivest
L21.4
11/18/08

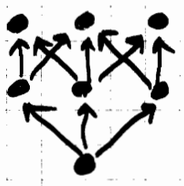
Subproblem dependence graph

vertices = $m \cdot n$

edges = $3(m-1)n - 2(m-1) = 3mn - 3n - 2m + 2$
$= \Theta(m \cdot n)$

(could add one dummy vertex at bottom,..)

This graph is a DAG

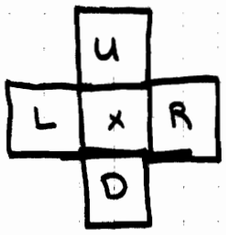


D.P. time = $\Theta(V+E) = \Theta(m \cdot n)$

Application: Image-Resizing (Avidan + Shamir) [4 min 27 sec]

* Play video (link from his home page is better than Youtube)

Cost fn:



$$\text{cost}(x) = \Delta(L,R) + \Delta(U,D)$$

where

$$\Delta(A,B) = \sum_{c=R,G,B} |A_c - B_c|$$

many other cost fns possible...

Have fun on this homework!

6.006
Rivest
L21.5
11/18/08

Matrix chain multiplication

$$A_1 \cdot A_2 \cdot \cdots \cdot A_n$$

where A_i is an $r_i \times c_i$ matrix.
Here $c_i = r_{i+1}$ for compatibility.

Matrix mult is associative, so we can do this many ways:

$$((A_1 \cdot A_2) \cdot (A_3 \cdot A_4))$$

$$A_1 \cdot (A_2 \cdot (A_3 \cdot A_4))$$

$$\vdots$$

(5 ways to do this)

They can have <u>dramatically</u> different costs.

Consider $A_1 \times A_2$ ($r_1 \times c_1$, $r_2 \times c_2$)

Cost $= r_1 c_1 c_2$
$= r_1 r_2 c_2$
(no choice)

but $(A_1 \times A_2) \times A_3$ (1×1000, 1000×1, 1×1000; 1×1)

cost $= 1 \cdot 1000 \cdot 1 + 1 \cdot 1 \cdot 1000 = 2000$

$A_1 \times (A_2 \times A_3)$ (1×1000, 1000×1, $1 \cdot 1000$; $1000 \cdot 1000$)

cost $= 1000 \cdot 1 \cdot 1000 + 1 \cdot 1000 \cdot 1000 = 2{,}000{,}000$

$\times 1000$ different!

We can find optimal parenthesization using D.P.:

subproblem = find cost of computing product $A_i \cdot A_{i+1} \cdots A_j = A_{i..j}$

call this cost m_{ij}

Then

$$m_{ii} = 0 \qquad \text{(only one matrix, no mpy's needed)}$$

$$m_{ij} = \min_{i \le k < j} \left(m_{ik} + m_{k+1,j} + r_i c_k c_j \right)$$

Can compute "bottom-up" or "top-down"

<u># vertices</u> in subproblem dependence graph is $n + \binom{n}{2} = \Theta(n^2)$

<u># edges</u> $= \sum_{1 \le i \le j \le n} (j-i) \cdot 2$ since m_{ij} depends on $(j-i) \cdot 2$ smaller subproblems

$$= \Theta(n^3)$$

(Don't confuse cost of computing optimal parenthesization with cost of then computing matrix chain product!)