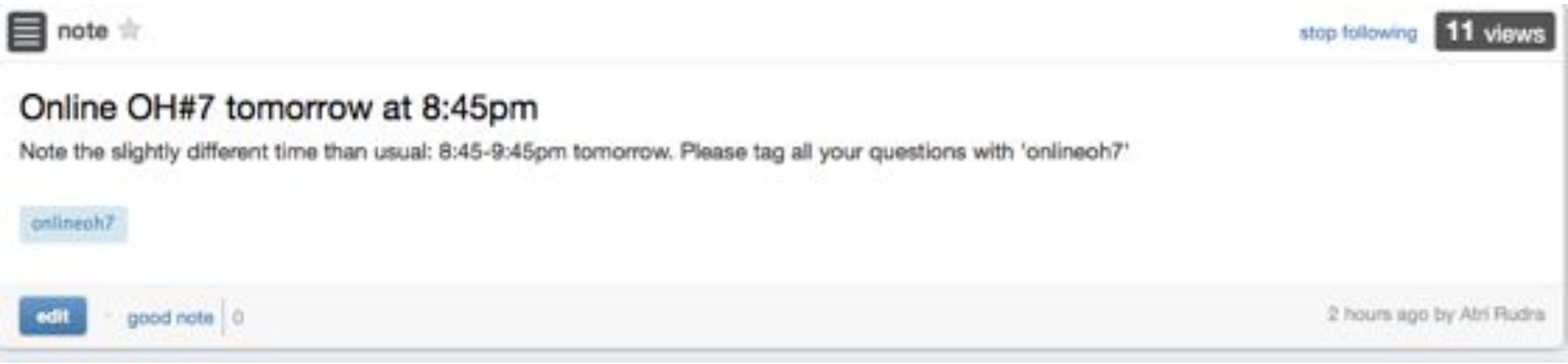
Lecture 29

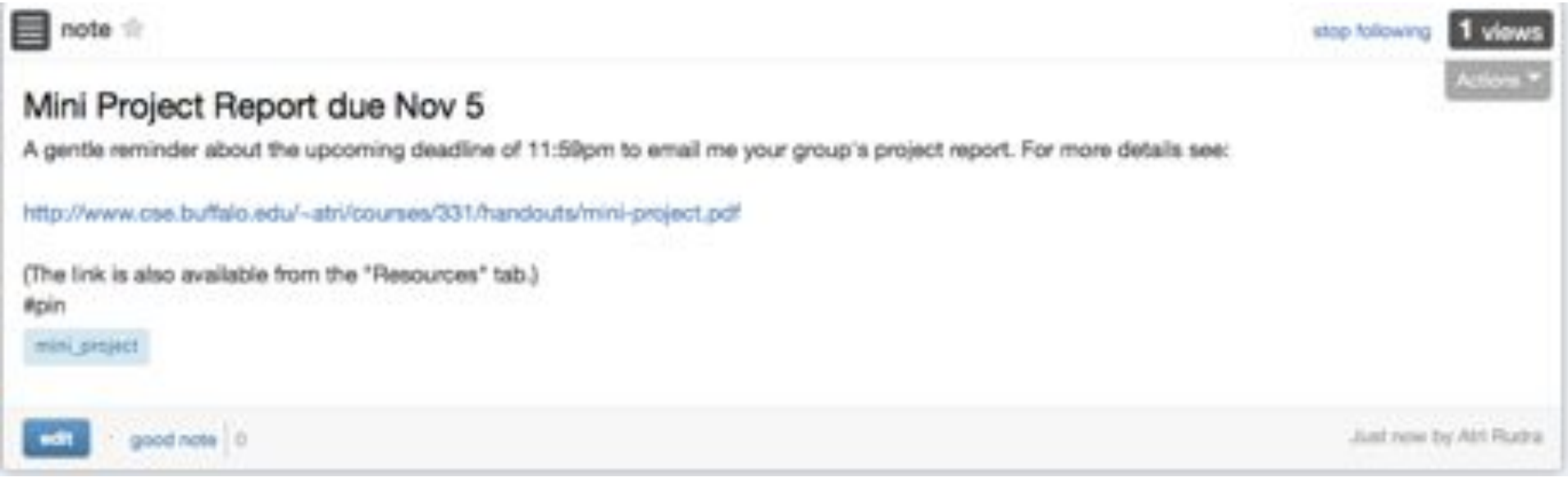
CSE 331

Nov 5, 2014

Online OH tonight @8:45pm



Mini project report due TONIGHT



Divide and Conquer

Divide up the problem into at least two sub-problems

Recursively solve the sub-problems

“Patch up” the solutions to the sub-problems for the final solution

Improvements on a smaller scale

Greedy algorithms: exponential → poly time

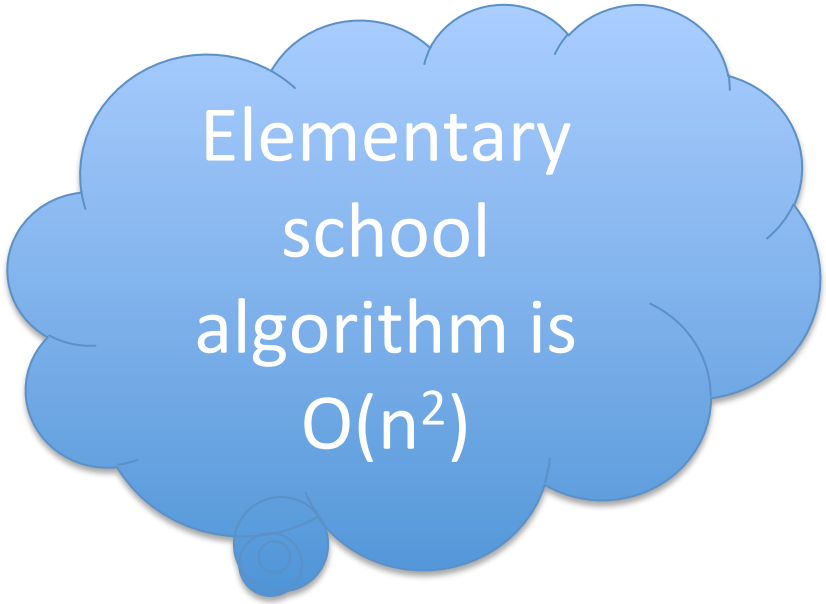
(Typical) Divide and Conquer: $O(n^2)$ → asymptotically smaller running time

Multiplying two numbers

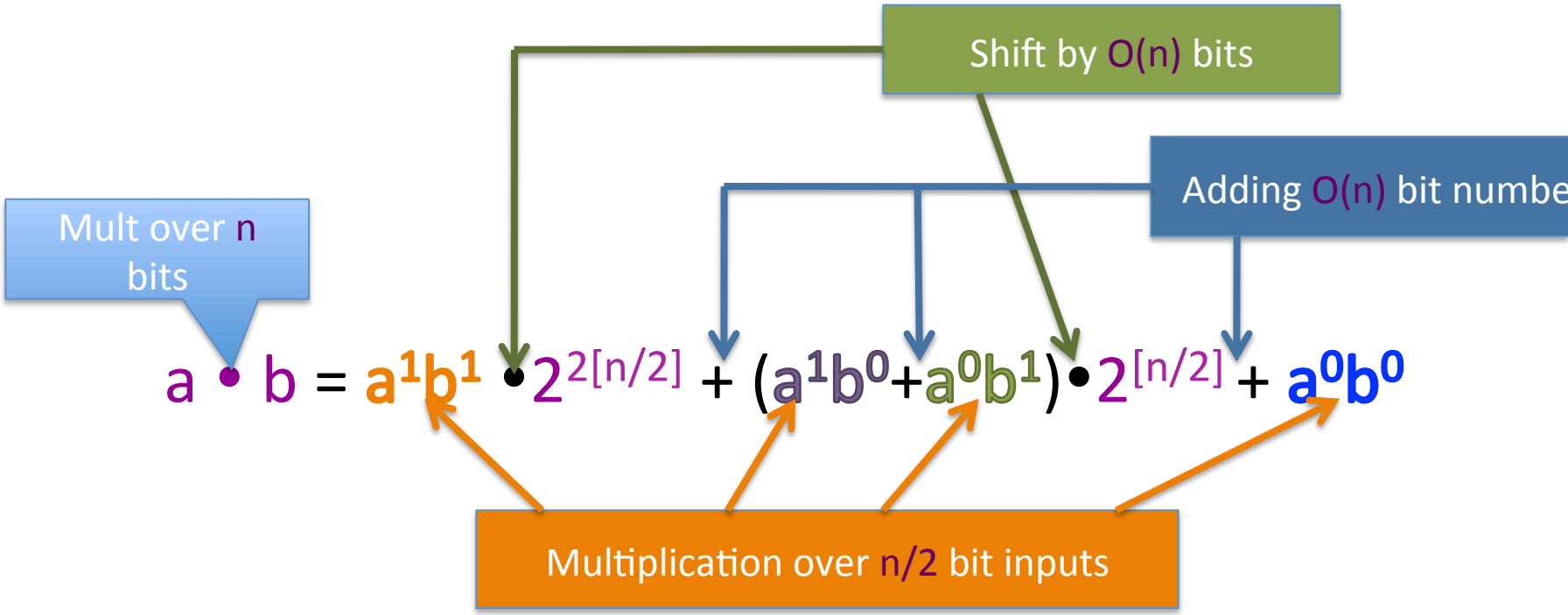
Given two numbers a and b in binary

$a=(a_{n-1},..,a_0)$ and $b = (b_{n-1},\ldots,b_0)$

Compute c = a x b



The current algorithm scheme



$T(n) \leq 4T(n/2) + cn$

$T(1) \leq c$

$T(n)$ is $O(n^2)$

The key identity

$$a^1b^0+a^0b^1 = (a^1+a^0)(b^1+b^0) - a^1b^1 - a^0b^0$$

The final algorithm

Input: $a = (a_{n-1},..,a_0)$ and $b = (b_{n-1},\ldots,b_0)$

Mult (a, b)

If $n = 1$ return $a_0 b_0$

$a^1 = a_{n-1},\ldots,a_{[n/2]}$ and $a^0 = a_{[n/2]-1},\ldots, a_0$

Compute b^1 and b^0 from b

$x = a^1 + a^0$ and $y = b^1 + b^0$

Let p = Mult (x, y), D = Mult (a^1, b^1), E = Mult (a^0, b^0)

$F = p - D - E$

return $D \bullet 2^{2[n/2]} + F \bullet 2^{[n/2]} + E$

$T(1) \le c$

$T(n) \le 3T(n/2) + cn$

$O(n^{\log 3}) = O(n^{1.59})$ run time

All **green** operations are $O(n)$ time

$$a \bullet b = a^1b^1 \bullet 2^{2[n/2]} + ((a^1+a^0)(b^1+b^0) - a^1b^1 - a^0b^0) \bullet 2^{[n/2]} + a^0b^0$$