Lecture 10 CS 124

Divide and Conquer

1. divide
2. conquer
3. combine

usually at low levels
can switch to another alg. to save overhead

ex: **Find min and max of an array**

[3, 4, 1, 6, 8]

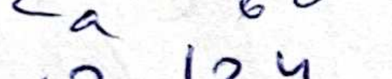
iterative: 2n comparisons

recursive: $T(n) = 2T\left(\frac{n}{2}\right) + 2$, $T(2) = 1$

$T(n) = \frac{3}{2}n - 2$, solve by induction or master theorem

ex: **integer multiplication**

$\Theta(n^2)$



$12\,|\,34$ $\quad x = 10^{n/2}a + b$

$\times\ 56\,|\,78$ $\quad y = 10^{n/2}c + d$

$x \cdot y = 10^n(a \cdot c) + 10^{n/2}(ad + bc) + bd$

$T(n) = 4T\left(\frac{n}{2}\right) + O(n)$ ← addition is $O(n)$ and multiplication by 10^x is just bit shifts

Master theorem

$= \Theta(n^2)$ ← down 4 sub-multiplications

$(a + b)(c + d) = ac + (bc + ad) + bd$

(bc + ad): difference

3 multiplications

$T(n) = 3T\left(\frac{n}{2}\right) + O(n)$

$\Theta(n^{\log_2 3}) \approx \Theta(n^{1.59})$ $\quad a = 3,\ b = 2,\ k = 1$ $\quad 3 > 2$

recurse down to machine instruction

there algs close to linear $\Theta(n^{1+\varepsilon})$, $\varepsilon > 0$

matrix multiplication

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$$

n

n

assume n is power of 2

Boring $c_{ik} = \sum_j a_{ij} \cdot b_{jk}$ $O(n^3)$

divided $T(n) = 8T\left(\frac{n}{2}\right) + O(n^2)$

$a=8, b=2, k=2$ $\quad 8 > 2^2$

$O(n^{\log_2 8}) = O(n^3)$

but we can get 7 multiplications

Strassen

$P_1 = A(F-H)$
$P_2 = (A+B)H$
$P_3 = (C+D)E$
$P_4 = D(G-E)$
$P_5 = (A+D)(E+H)$
$P_6 = (B-D)(G+H)$
$P_7 = (A-C)(E+F)$

lin. combos

$$\begin{bmatrix} P_5+P_4 & \\ -P_2+P_6 & \end{bmatrix}$$

more additions

$T(n) = 7T\left(\frac{n}{2}\right) + O(n^2)$

$O(n^{\log_2 7}) \approx O(n^{2.807})$

$O(n^{2+\varepsilon})$

2.796, 2.78, 2.522,
2.517, 2.496, 2.479
2.376, 2324, 2.3727

Dynamic Programming

String Reconstruction

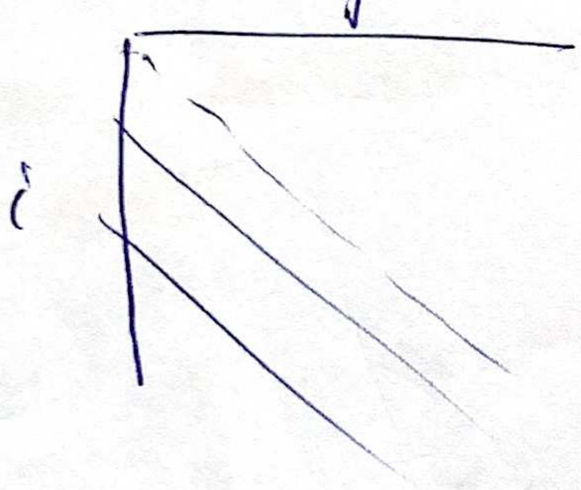
THESEARETHEREASONS

1) greedy + backtracking → exp. runtime

2) Div. Conc. with fixed splitting → exp. runtime

3) exhaustive splitting [redacted]
   ↳ looking for subproblem

$O(n^3)$




```
for d = 1 ... n-1
  for i = 1 ... n-d
    j = i + d
    for k = i+1 ... j-1
D(i,j) = D(i,k) and D(k,j)
```