Analysis of Algorithms

- Design & analysis of algorithms
  - proofs
  - theory
  - eye to practice

Matrix Multiplication

$$\begin{bmatrix} 3 & 1 & 1 \\ 2 & 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \\ 2 & 6 \end{bmatrix} = \begin{bmatrix} 5 & 9 \\ 10 & 24 \end{bmatrix}$$

pseudocode

input A $n \times m$

B $m \times p$ $n = m = p$

output C $n \times p$

$C = A \cdot B$

```
for i = 1 to n
    for j = 1 to ~~m~~ p
        C[i,j] = 0
        for k = 1 to ~~p~~ m
            C[i,j] += A[i,k] · B[k,j]
```

Running time = $O(nmp)$

$O(n^3)$

Lower bound of

$\Omega(n^2)$

Can you do better than

$O(n^3)$?

Use recursion?

$$\left(\begin{array}{c:c} a & b \\ \hline c & d \end{array}\right)\left(\begin{array}{c:c} e & g \\ \hline f & h \end{array}\right) = \left(\begin{array}{c:c} r & s \\ \hline t & u \end{array}\right)$$

$r = ae + bf$
$s = ag + bh$
$t = ce + df$
$u = cg + dh$

2 $n \times n$ mat. mult.

$\rightarrow$ 8 $\frac{n}{2} \times \frac{n}{2}$ mat. mult. $\left(\frac{n}{2}\right)^3$

4 $\frac{n}{2} \times \frac{n}{2}$ mat. additions $\left(\frac{n}{2}\right)^2$

Recurrences

$T(n)$ = time to mult. 2 $n \times n$ matrices

$$T(n) = \overset{7}{\cancel{8}}\, T\left(\frac{n}{2}\right) + \overset{22}{\cancel{4}}\left(\frac{n}{2}\right)^2$$

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

$$= O(n^3) \rightarrow O(n^{\lg_2 7}) = O(n^{2.81}) \rightarrow O(n^{2.37\ldots})$$

fewer mults, but more additions

$\exists$ faster alg. $O(n^{2.37\ldots})$

Maximum Subsequence Sum

Given n numbers $a_1 \cdots a_n$
(at least one is positive)

compute

$$\max_{1 \le l \le u \le n} \sum_{k=l}^{u} a_k .$$

$$-2 \quad \underbrace{11 \quad -4 \quad 13}_{20} \quad -5 \quad 4 \quad -2$$

n^2 subsequence

each consists of $\leq n$.

terms

Straightforward $O(n^3)$

alg.

only look sub seq. that start/end w a number > 0

Compute all n^2 subseq.

$$S[x,y] = S[x,y-1] + a_y.$$

replace a loop w/ 1 addition

$O(n^2)$ | $O(n)$