HW1 due tomorrow (Wed) at 9pm
(no late HW)

Co-instructor: Bhaskar Ray Chaudhury

Matrix Multiplication

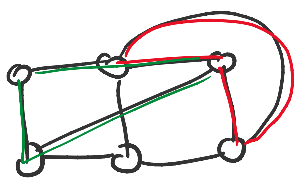
$O(n^{2.81})$ Strassen

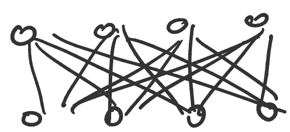
$\downarrow$

$n^{2.373}$

Application Given graph $G=(V,E)$, with n vertices,

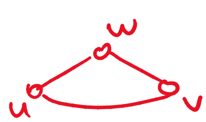
decide $\exists$ cycle of length 3





naive / brute-force alg'm: $O(n^3)$ time

let $a_{uv} = \begin{cases} 1 & \text{if } uv \in E \\ 0 & \text{else} \end{cases}$ (adjacency matrix)



$\exists w \in V$ s.t. $uw \in E \wedge wv \in E$

$\underbrace{a_{uw}=1 \wedge a_{wv}=1}$

$a_{uw}a_{wv}=1$

$c_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}$

$$\sum_{w \in V} a_{uw}a_{wv} > 0$$

for $u \in V$ do
  for $v \in V$ do
    Compute $c_{uv} = \sum_{w \in V} a_{uw}a_{wv}$ ← by matrix mult.
    if $c_{uv} > 0$ & $uv \in E$

return yes

return no

$\Rightarrow \boxed{O(n^{2.81})}$ time by Strassen

DYNAMIC PROGRAMMING (DP)

① define subproblems

② derive recursive formula to express ans to subproblem (opt value) in terms of answers to smaller subproblems

③ evaluate formula bottom-up using a <u>table</u>

Line Break Problem

it is a
good
algorithm

$\langle 3,3,2,5,10\rangle$, $L=11$

$\hookrightarrow \langle 3,3,2\rangle, \langle 5\rangle, \langle 10\rangle$

Given sequence $\langle a_1,\dots,a_n\rangle$ and L,

split into subseqs $\langle a_1,\dots,a_{i_1}\rangle, \langle a_{i_1+1},\dots,a_{i_2}\rangle,$

$\dots, \langle a_{i_{k-1}+1},\dots,a_n\rangle,$

$1=i_0<i_1<\dots<i_{k-1}<i_k=n,$

s.t. each subseq has sum $\leq L$

to minimize penalty $\sum_{j=1}^{k}\left(L-(a_{i_{j-1}+1}+\dots+a_{i_j})\right)^2$

① For each $i=0,\dots,n$,

define $P(i)$ = min penalty for the input sequence $\langle a_1,\dots,a_i\rangle$

Want $P(n)$.

② Recursive formula:

if last subseq we split into $\langle a_{j+1},\dots,a_i\rangle$,

if last subseq we split into $\langle a_{j+1}, \ldots, a_i \rangle$,
then $P(i) = P(j) + (L - (a_{j+1} + \ldots + a_i))^2$

but don't know j, so try all & take min

$$\Rightarrow \boxed{P(i) = \min_{\substack{j \in \{0,\ldots,i-1\}: \\ a_{j+1}+\ldots+a_i \le L}} \left(P(j) + (L - (a_{j+1}+\ldots+a_i))^2 \right)}$$

Base case: $P(0) = 0$.

If we evaluate formula recursively,

$$T(i) = T(i-1) + T(i-2) + \ldots + T(0)$$

$\Rightarrow$ exponential!

Instead, evaluate in increas. i & use table

Pseudocode:

$s_0 = 0$
for $i = 1$ to n
$s_i = s_{i-1} + a_i$

$P[0] = 0$
for $i = 1$ to n
$$P[i] = \min_{\substack{j \in \{0,\ldots,i-1\}: \\ a_{j+1}+\ldots+a_i \le L}} \left(P[j] + (L - (a_{j+1}+\ldots+a_i))^2 \right)$$

$(s_i - s_j)^2$

$(a_1 + \ldots + a_i) - (a_1 + \ldots + a_j)$

Pred$[i] = j$ that attains the above min

return $P[n]$.

$\Rightarrow O(n^3)$ time

improves to $\boxed{O(n^2)}$ by using prefix sums by using prefix sums

$\boxed{O(n)}$ space

How to output opt sol'n:

```
output-sol(i):
    if i=0 return
    j = pred[i], output-sol(j)
    output ⟨ a_{j+1}, .., a_i ⟩.

Call output-sol(n).
```

additional $O(n)$ time

Alternative 1: "forward" version

for $i = 1, .., n+1$,

define $P(i)$ = min penalty for sequence $\langle a_i, .., a_n \rangle$

Want $P(1)$.

$$P(i) = \min_{\substack{j \in \{i+1, .., n+1\} \\ a_i + .. + a_{j-1} \leq L}} \left(P(j) + (L - (a_i + .. + a_{j-1}))^2 \right)$$

Alternative 2: graph version . . .