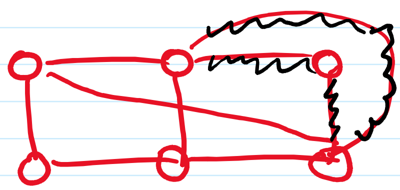
Tuesday, September 8, 2026 1:46 PM

Last Time: Matrix Multiplication

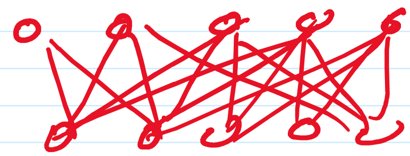
Strassen $O(n^{2.81})$ time

current record $O(n^{2.371177})$

Appl'n: Given undir. graph $G=(V,E)$, with n vertices, decide $\exists$ cycle of length 3

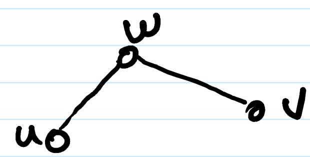


yes



brute force alg'm: $O(n^3)$ better?

Let $a_{uv} = \begin{cases} 1 & \text{if } uv \in E \\ 0 & \text{else} \end{cases}$



for $u \in V$ do

for $v \in V$ do

$c_{uv} = \boxed{\sum_{w \in V} a_{uw} a_{wv}}$

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

if $c_{uv} > 0$ & $uv \in E$ return yes

return no

$\Rightarrow \underline{O(n^{2.81})}$ time

DYNAMIC PROGRAMMING (DP)

- define subproblems
- derive recursive formula to express ans to subproblems in terms of ans to smaller subproblems
- evaluate formula bottom-up using a table

"Line Break" Problem

"derive recursive formula to express ans to subproblems

L

Given sequence of n numbers $\langle a_1, \ldots, a_n \rangle$ and L,

split into subseqs $\langle a_1, \ldots, a_{i_1} \rangle$, $\langle a_{i_1+1}, \ldots, a_{i_2} \rangle$, $\ldots$, $\langle a_{i_{k-1}+1}, \ldots, a_n \rangle$

$\langle 5, 9, \ldots \rangle$

$1 = i_0 < i_1 < \cdots < i_{k-1} < i_k = n.$

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

Define Subproblems: for $i = 0, \ldots, n$,

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

Want $P(n)$.

Recursive formula:

$a_1 \ldots \ldots a_j \mid a_{j+1} \ldots a_i$

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

idea: then $P(i) = P(j) + (L - a_{j+1} - \cdots - a_i)$

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

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

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

if we evaluate formula recursively,
worst-case runtime $T(i) \geq T(i-1) + T(i-2) + \cdots$
$\Rightarrow$ exponential.

Instead, evaluate in increas. i & use table

Pseudocode:

$P[0] = 0$
for $i = 0$ to n
$$P[i] = \min_{\substack{j \in \{0,\dots,i-1\}: \\ a_{j+1}+\dots+a_i \le L}} \left(P[j] + (L - \overbrace{a_{j+1} - \cdots - a_i}^{s_i - s_j})^2 \right)$$
pred[i] = j that achieves this min

$\cancel{O(n)}$ per j
$O(1)$

return $P[n]$.

prefix sum:
$s_i = a_1 + \cdots + a_i$

Analysis:

$O(n)$ subprobs.
each requires $\cancel{O(n^2)}$ $O(n)$ time
$\Rightarrow \boxed{\cancel{O(n^3)}}$ total time
$\boxed{O(n^2)}$

$O(n)$ space

$O(n)$ space

How to output opt sol'n:

```
output-sol(i):
    if i = 0 return
    output-sol(pred[i]), output pred[i])
```

$\}$ $O(n)$ additional time

Alternative 1: forward version

suffix

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

Want $P(1)$

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

Alternative 2: graph version

define graph where

vertices are $i \in \{1, .., n+1\}$

"states"

place edge $i \to j$ of weight $(L - a_i - \cdots - a_{j-1})^2$

if $a_i + .. + a_{j-1} \leq L$.

"transition"

shortest path from 1 to $n+1$ in DAG

$O(n)$ vertices

$O(n^2)$ edges