More Dynamic Programming:

Longest common subsequence = LCS:

input: strings $X = x_1 x_2 \dots x_n$
& $Y = y_1 y_2 \dots y_m$

Goal: find length of longest string which is a subsequence of both X & Y

Example: X = BCDBCDA
Y = ABECBA
answer = 4 for BCBA

Application: used in unix diff for comparing files.

First step is to define the subproblem.
As before consider prefixes. In this case prefix of X & prefix of Y.

For i & j where $0 \le i \le n$ & $0 \le j \le m$ let:

$L(i,j)$ = length of LCS in $x_1, \dots, x_i$
& $y_1, \dots, y_j$

Base cases: $L(i,0)=0$

$L(0,j)=0$

For recurrence, two cases: $X_i = Y_j$ or $X_i \neq Y_j$

If $X_i \neq Y_j$ then:

$$L(i,j) = \max\{L(i,j-1), L(i-1,j)\}$$

If $X_i = Y_j$ then:

$$L(i,j) = \max\{1 + L(i-1,j-1), L(i,j-1), L(i-1,j)\}$$

(In fact, can argue that: $L(i,j) = 1 + L(i-1,j-1)$)

LCS(X,Y):

for $i = 0 \to n$, $L(i,0)=0$
for $j = 0 \to m$, $L(0,j)=0$
for $i = 1 \to n$
  for $j = 1 \to m$
    if $X_i = Y_j$ then
      $L(i,j) = \max\{1 + L(i-1,j-1), L(i,j-1), L(i-1,j)\}$
    else
      $L(i,j) = \max\{L(i,j-1), L(i-1,j)\}$

Return $(L(n,m))$

Running time: $O(nm)$.

How would you do longest common <u>substring</u>?

Chain Matrix Multiply

<u>Example:</u> for 4 matrices A, B, C, D
we want to compute $A \times B \times C \times D$ most efficiently.

Say A is size 50×20, B is 20×1,
C is 1×10, & D is 10×100.

Since matrix multiplication is associative we can compute:
$((A \times B) \times C) \times D$ or $(A \times B) \times (C \times D)$ etc.

Which is best?

For W of size $a \times b$ & Y of size $b \times c$
then $Z = WY$ is of size $a \times c$.

Define <u>cost</u> of multiplying $W \times Y$ is abc

Then, $((A \times B) \times C) \times D$ costs $(50)(20)(1) + (50)(1)(10) + (50)(10)(100)$
$= 51{,}500$

$(A \times B) \times (C \times D)$ costs $7{,}000$

$(A \times (B \times C)) \times D$ costs $60{,}200$

General problem:

For n matrices $A_1, A_2, \ldots, A_n$ where A_i is of size $m_{i-1} \times m_i$, what is the min cost for computing $A_1 \times A_2 \times \cdots \times A_n$?

First try: Use prefixes.

Let $C(i)$ = min cost for computing $A_1 \times A_2 \times \cdots \times A_i$

But the last multiplication will be of the form:

$$(A_1 \times \cdots \times A_j) \times (A_{j+1} \times \cdots \times A_i) \text{ for some } j.$$

Can look up optimal cost for

but not for this not a prefix!

Need to use substrings.

For $1 \le i \le j \le n$,

let $C(i,j)$ = min cost of computing $A_i \times \cdots \times A_j$.

Base case: $C(i,i) = 0$.

For $i<j$, try all l for the split where $i \le l \le j-1$

then multiply: $(A_i \times \cdots \times A_l) \times (A_{l+1} \times \cdots \times A_j)$

costs $C(i,l)$ costs $C(l+1,j)$

costs $m_{i-1} m_l m_j$

$$C(i,j) = \min_{i \le l \le j-1} \{ C(i,l) + C(l+1,j) + m_{i-1} m_l m_j \}$$

How do you fill the table?

By width $s = j - i$.

ChainMultiply$(m_0, m_1, \ldots, m_n)$:

```
For i=1 → n, C(i,i)=0
For s=1 → n-1,
    For i=1 → n-s,
        Let j=i+s
        C(i,j)=∞
        For l=i → j-1
            cur= m_{i-1} m_l m_j + C(i,l) + C(l+1,j)
            if C(i,j) > cur then C(i,j)=cur
Return(C(1,n))
```

Running time: $O(n^3)$.