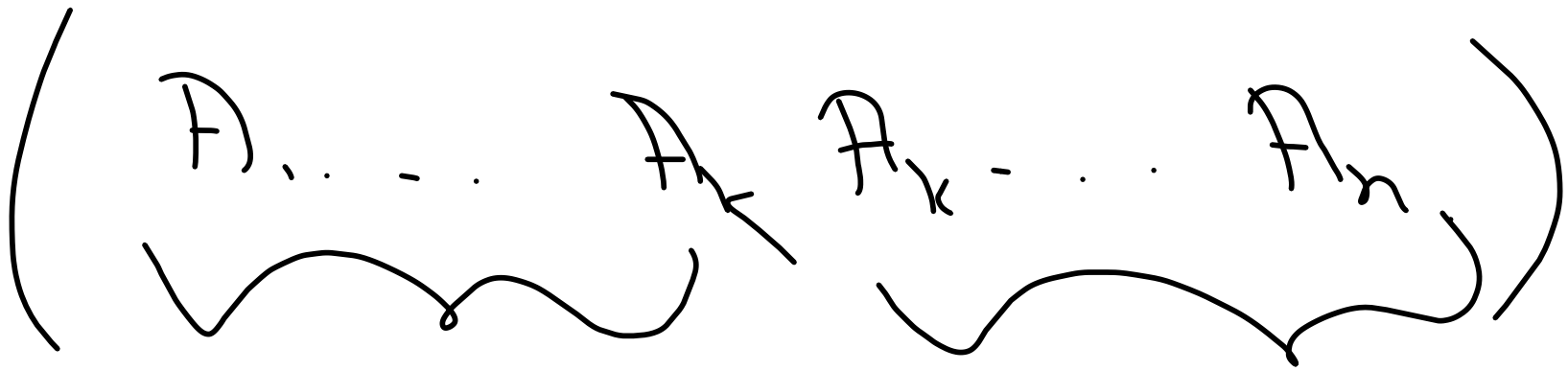
Matrix-Chain-Multiplication Problem

Counting the no. of parenthesizations

Let P(n) be the no. of ways of fully parenthesizing a product n matrices

The product can be fully parenthesized by breaking it between A_k and A_{k+1} then independently parenthesize the 2 subsequences to obtain the final parenthesization



Thus we have the following rec:

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ \sum_{k=1}^{n-1} T(k)T(n-k) & \end{cases}$$

The soln is $T(n) = C(n-1)$,

where $C(n) = \frac{1}{n+1}\binom{2n}{n}$, is the n^{th} Catelan no.

The matrix-chain-multiplication Problem

Given a chain $A_1, \ldots, A_n$ of n matrices, where A_i is of dimension $p_{i-1} \times p_i$, $1 \le i \le n$, fully parenthesize the product in a way that minimizes the no. of scalar multiplications

Structure of optimal Solution

Let $A_{[i,j]}$ denote the product $A_i \cdot A_{i+1} \cdots A_j$, $i \le j$

The optimal parenthesization of $A_{[1,n]}$ breaks the seqn between A_k & A_{k+1} for some k, $1 \le k < n$

Thus for some k, we form the products $A_{[1,k]}$, $A_{[k+1,n]}$ & then

multiply these two. Thus the optimal cost of forming the product is the cost of obtaining $A[1,r]$ + the cost of obtaining $A[r+1,n]$ + the cost of multiplying these two

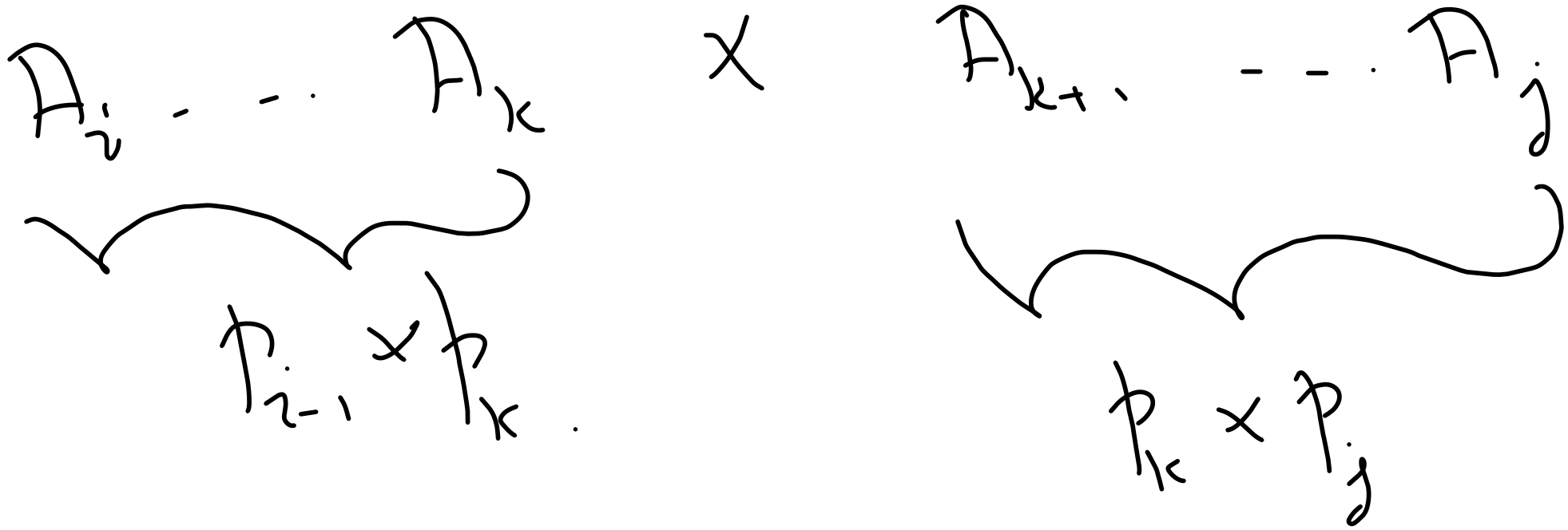
This gives rise to a recurrence relation as follows:

Thus the subproblems that we consider are of the form $A_{[i,j]}$, $i \le j$

Let $m[i,j]$ be the min no. of multiplications used in forming $A_{[i,j]}$, $i \le j$

Then $m[i,j]$ satisfies the following

$$m[i,j] = \begin{cases} 0 & \text{if } i = j \\ \min_{i \le k < j} \{ m[i,k] + m[k+1, j] + d_{i-1} d_k d_j \} \end{cases}$$


$A_i \cdots A_k$
$p_{i-1} \times p_k$
$\times$
$A_{k+1} \cdots A_j$
$p_k \times p_j$

To solve by dynamic programming method we shall use the above recurrence relation by a bottom-up method by constructing tables.

We assume that each A_i is of dimension $p_{i-1} \times p_i$ & the input is a seq $p = (p_0, \ldots p_n)$ of length $length(p) = n+1$. Our procedure uses an auxiliary table $m[1..n, 1..n]$ that stores $m[i,j]$ and an auxiliary table

that stores $s[i,j]$

$s[1..n, 1..n]$, that records which index k was used while computing $m[i,j]$

```
Matrix-Chain-Product(p).
1.  n ← length(p) - 1
2   for i ← 1 to n
3.      do m[i,i] = 0
4.  for l ← 2 to n
5.      do for i ← 1 to n-l+1
6.          do j ← i+l-1
7.              m[i,j] ← ∞
8.          for k ← i to j-1
9.              do q ← m[i,k] + m[k+1,j]
                       + p_{i-1} p_k p_j
10.             if q < m[i,j]
11                  then m[i,j] ← q
12                      s[i,j] ← k.
13.     return m and s.
```

In the first execution of the for loop the alg. computes m[i, i+1] for i = 1, ..., n-1.

In the 2nd execution it computes m[i, i+2] for i = 1, --- , n-2 and so forth.

The alg. has 3 for loops nested 3 deep.

Each loop is executed at most n times

at so the complexity is $O(n^3)$

Ex Let $R(i,j)$ is the no. of times $m[i,j]$ is called while computing the table m. Show that

$$\sum_i \sum_j R(i,j) = \frac{n^3 - n}{3}$$

Obtaining the optimal solution

Matrix-Chain-Multiply(A, s, i, j)

$A_i \cdots A_x \cdots A_j$

1 if $j > i$

do { $X \leftarrow$ Matrix-Chain-Multiply(A, s, i, s[i,j])

$Y \leftarrow$ Matrix-Chain-Multiply(A, s, s[i,j]+1, j)

Return $X \times Y$.

else return A_i

3. Subset sums & Knapsacks.

Subset-sum problem

Given a bound W & n items $\{1, \ldots n\}$ with non-negative wts $w_1 \ldots w_n$, find a subset $S \subseteq \{1, \ldots, n\}$ s.t.

(i) $\sum_{i \in S} w_i \leq W$

(ii) $\sum_{i \in S} w_i$ is as large as possible.

Knapsack Problem

Given a set of n items $\{1 \dots n\}$ with weights $w_1 \dots w_n$ & values $v_1 \dots v_n$ respectively, find $S \subseteq \{1 \dots n\}$ s.t.

(1) $\sum_{i \in S} w_i \leq W$

& (2) $\sum_{i \in S} v_i$ is as large as possible.

Structure of optimal solutions.

Let $S \subseteq \{1, \ldots, n\}$ be an optimal soln

If $n \notin S$ then clearly S is an optimal soln of the corresp Knapsack with item $1 \ldots n-1$ & W. limit W.

If $n \in S$, then $S' = S - \{n\}$
then S' is an optimal soln
of the knapsack with items $1 \ldots n-1$
& W-limit $W - w_n$.

We shall assume that wts are integers
& also the bd. W.

We shall consider the knapsack prob.
with i items $1 \ldots i$, a W-limit w, where
$1 \leq w \leq W$

Let $M[i, w]$ be the wt. of the optimal sln for the Knapsack with items $1, 2, \ldots, i$ & wt-limit w.

$$M[i,w] = \begin{cases} M[i-1, w] & \text{if } w < w_i \\ \text{Max}\{M[i-1, w] \; , \; w_i + M[i-1, w-w_i]\} \end{cases}$$

Subset-Sum (n, W)

Array $M[0..n; 0..W]$

Initialize $M[0, w] = 0$ for all $w = 0, \ldots W$

for $i \leftarrow 1$ to n

do if $W < w_i$

then $M[i, w] \leftarrow M[i-1, 0]$

else $M[i, w] \leftarrow \max\{M[i-1, w),$
$w_i + M[i-1, w - w_i]\}$

Ex Find the subset $S \subseteq \{1, \dots n\}$ for which $\sum_{i \in S} w_i$ is maximized.

Ex Modify the alg. to solve the knapsack problem.