DYNAMIC PROGRAMMING

PROGRAMMING — TELEVISION PROG.

ESTAB A SCHEDULE

1) PRINCIPLE OF OPTIMALITY

2) MEMOIZATION

$$F_0 = F_1 = 1$$

$$F_i = F_{i-1} + F_{i-2}$$

function F(i) ....

if $i \le 1$ return 1

else return F(i-1) + F(i-2)

If F[i] exists
return F[i]
else

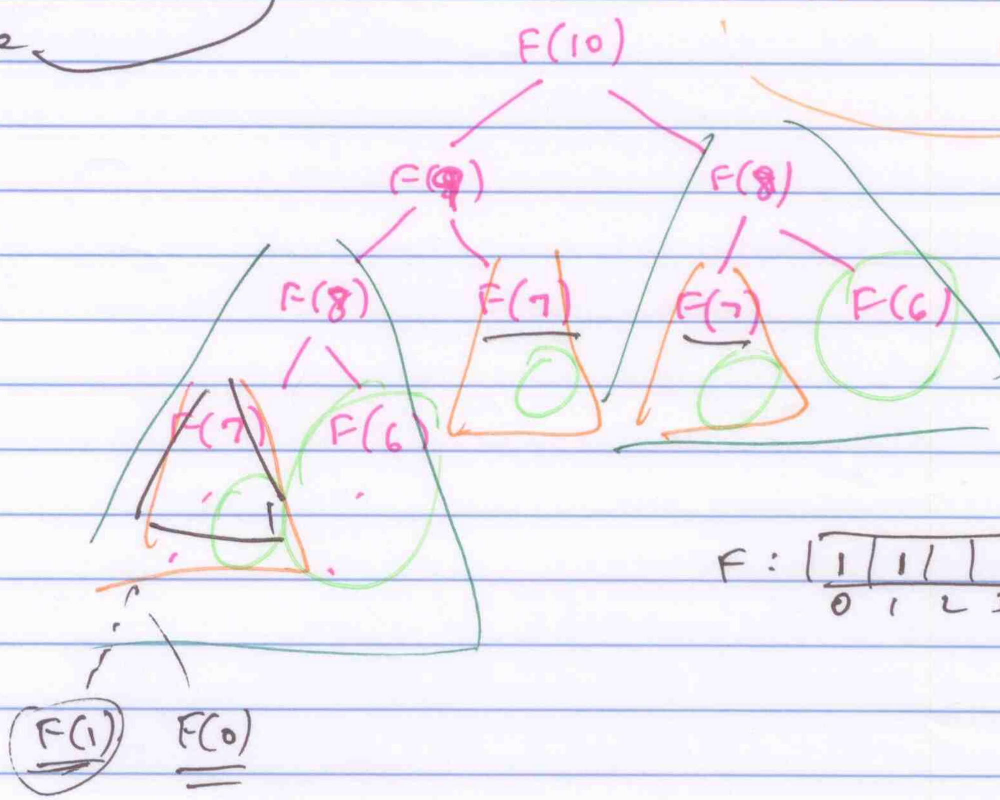
$T(i)$

$i \le 1$ — 0

$i > 1$ $\quad 1 + T(i-1) + T(i-2)$

$\Downarrow$

$$T(i) = \Theta(\varphi^i)$$



F: | 1 | 1 | | | | | | ...

0 1 2 3 4 5 ...

"MEMOIZED"

```
function F(i)
    if F[i] exists then return F[i]
    else { if i≤1 F[i]=1
           F[i] = F(i-1) + F(i-2)
           return F[i]
```

F: | 1 | 1 | 2 | 3 | 5 | 8 | 13 | | | ....

TRIVIAL (UNNEEDED)

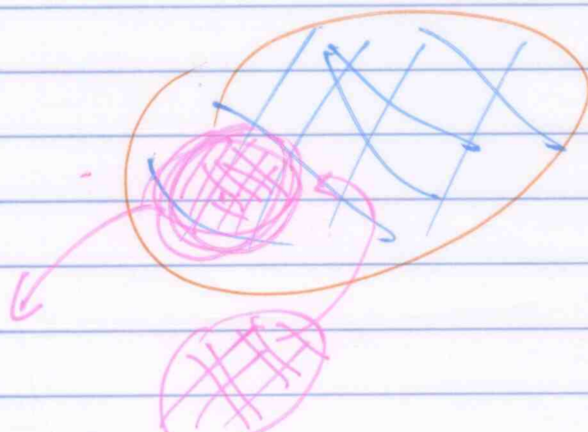
```
F[0]=F[1]=1
for i = 2..n
    F[i] = F[i-1] + F[i-2]
```

n-1

PRINC OF OPT

TREE DRAWING

SUBSOLUTIONS TO AN OPTIMAL SOLUTION ARE THEMSELVES OPTIMAL



ROD CUTTING

n

1	p_1
2	p_2
$\vdots$	
n	p_n

← READ

$$(A_1((A_2\ A_3)(\cdot\ \cdot)))\ A_n$$

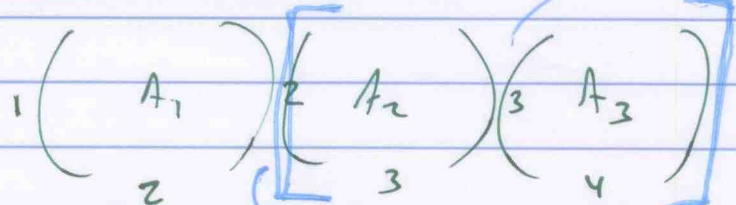
$$\boxed{p_0 \times p_1}\ \boxed{p_1 \times p_2}\ \boxed{p_2 \times p_3} \cdots \boxed{p_{n-1} \times p_n}$$

n_2 mults
n_2 adds

$n_1 \times n_2$ times $n_2 \times n_3$ → $n_1 \times n_3$

$n_1 n_2 n_3$

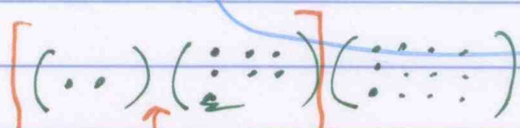
Matrix mult is associative



$2 \cdot 3 \cdot 4 = 24$

$1 \times 2 \times 4 = 8$

32



$1 \times 2 \times 3$

$1 \times 3 \times 4$

$6 + 12 = 18$

n-1 choices for k

$$\overbrace{(A_1 \mid \underset{k}{A_2} \mid A_3 \ldots)}^{}\ \overbrace{(\ldots \mid A_n)}^{n-k}$$

$p_0 \times p_1 \quad p_1 \times p_2 \quad p_2 \times p_3 \quad \ldots \quad p_{n-1} \times p_n$

final mult

k = 1
OR
k = 2
⋮
k = n-1

rule of sum

How many ways to parenthesize?

rule of product

$$P(n) = \sum_{k=1}^{n-1} P(k)P(n-k)$$

$P(1) = 1$

$P(2) = 1$

→ CATALAN NUMBERS

$P(3) = 2$

$P(4) = 5$

$P(5) = 14$

$P(6) = 42$

⋮

$$P(n) = \binom{2n-2}{n-1} / n$$

STIRLING'S FORMULA.

$n!$ — exponential

$\binom{2n-2}{n-1}$ — exponential

PRINC OF OPT — HOLDS!

$$\overbrace{(A_1(A_2)A_3)}^{A_1 \ldots A_k}\overbrace{(A_n)}^{A_{k+1} \ldots A_n}$$

k = i
k = i+1
k = i+2
⋮
k = j-1

worse

better

$p_i \times p_{k-1}$ k $p_{k-1} \times p_j$

C(i,j) = opt cost to mult

Best way?

$$C(i,j) = \min_{\boxed{i \le k \le j-1}} \left[C(i,k) + C(k+1,j) + p_i p_k p_j \right]$$

min = 0
for k = i … j-1 do
if < min then ...

$$C(i,i) = 0$$

function $\boxed{c(i,j)}$ if $i = j$ return 0

~~min ← -∞~~

~~min ← ∞~~

for $h = i \ldots j-1$ do

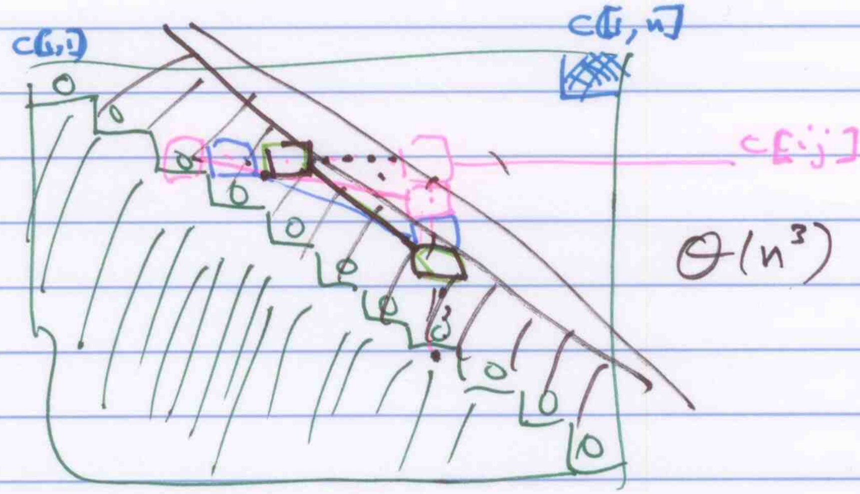
if $c(i,h) + c(h+1,j) + \boxed{p_i p_h p_j}$ < min then

~~min ←~~

~~min ← h~~

return $\boxed{h}$ cost.

MEMOIZE — menos $c[i,j]$ $i \le j$



$\Theta(n^3)$