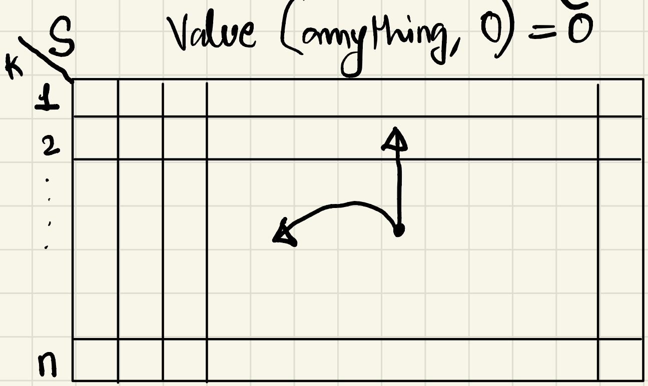
DYNAMIC PROGRAMMING

- MASTER RECURRENCE $\binom{n}{m}$ = choose m items from n = $\overset{\text{include } m^{th}}{\binom{n-1}{m-1}} + \overset{\text{Don't include } m^{th}}{\binom{n-1}{m}}$

- 0-1 Knapsack
  - → values : $v_1, v_2, \dots, v_n$
  - → wts : $w_1, w_2, \dots, w_n$
  - → Goal : Maximize value within capacity Z
  - → RECURRENCE : $Value(S, k)$:= Max value I can get in a knapsack of size S, when I am allowed to choose from items 1 through k.
  - → GOAL : $Value(Z, n)$

$$Value(S, k) = \max \left\{ \overset{\text{include } k^{th}}{v_k + Value(S - w_k, k-1)} \;,\; \overset{\text{don't include}}{Value(S, k-1)} \right\} \qquad \Theta(Zn)$$

$Value(\text{anything}, 0) = 0$ $\qquad$ $Value(-ve, \text{anything}) = 0$



top to bottom, left to right

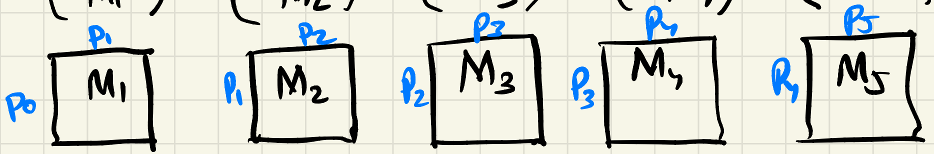
- **COIN CHANGE**
  - coins $d_1, \dots, d_k$ (infinite supply)
  - Goal: Minimum # coins needed to make amt A
  - RECURRENCE: $C(a)$: Min # of coins needed to make the change a
  - GOAL: $C(A)$

$$C(a) = \min_{d_i \leq a} \{1 + C(a-d_i)\} \qquad \Theta(n,k)$$

Base case $C(\leq 0) = 0$

- **MATRIX CHAIN MULTIPLY**

$${}_{10}(M_1)^{20} \quad {}_{20}(M_2)^{10} \quad {}_{10}(M_3)^{15} \quad {}_{15}(M_4)^{20} \quad {}_{20}(M_5)^{15}$$

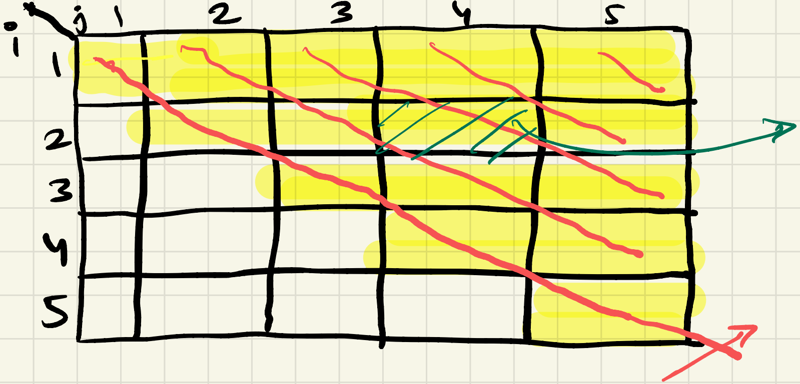


$M(i,j) :=$ Min # of multiplications to multiply matrices i through j (inclusive)

Goal: $M(1,5)$

RECURRENCE:

$$M(i,j) = \min_{i \leq k < j} \{ \underbrace{M(i,k)}_{(p_{i-1},\, p_k)} + \underbrace{M(k+1,j)}_{(p_k,\, p_j)} + p_{i-1} p_k p_j \}$$



$$M(1,4) \xrightarrow{\text{split1}} (M_1)(M_2\ M_3\ M_4)$$

$$M(1,4) \xrightarrow{\text{split2}} (M_1\ M_2)(M_3\ M_4)$$

$$M(1,4) \xrightarrow{\text{split3}} (M_1\ M_2\ M_3)(M_4)$$