

Optimal binary search tree

Problem: we are given sorted keys $k_1, \dots, k_n$, and probabilities $p_1, \dots, p_n$ that key i will be accessed at any point in time. Construct a binary search tree T that minimizes

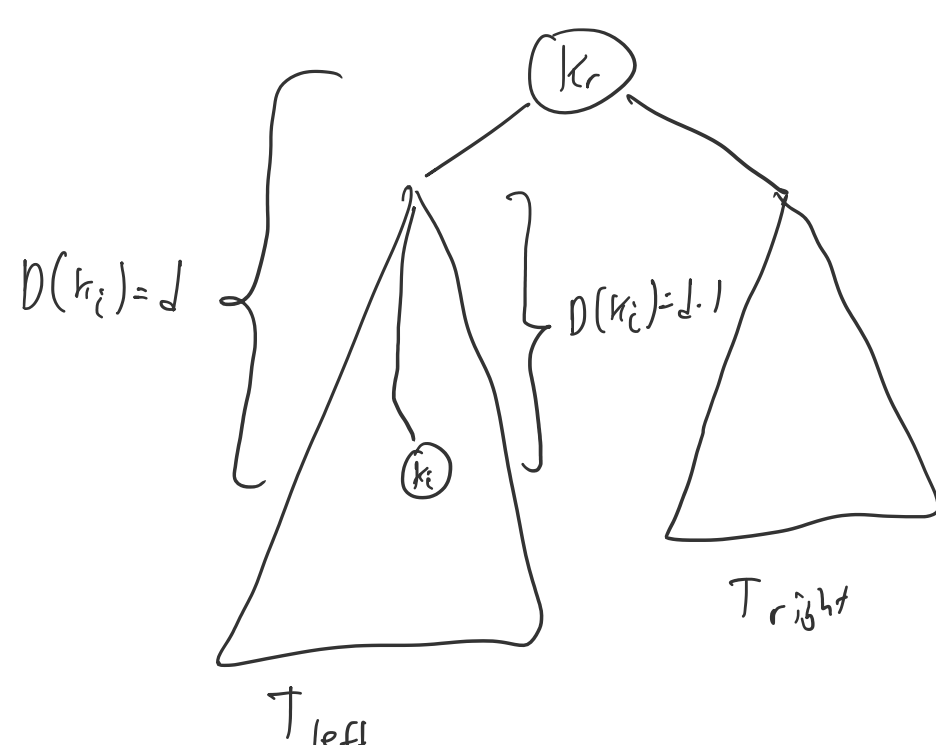
$$C(T) = \sum_{i=1}^n p_i (\overbrace{\text{Depth}(T, k_i) + 1}^{\text{distance from root to key } k_i}).$$

expected cost to find a key in tree by going down.

Let $D(T, k) = \text{Depth}(T, k)$ for brevity.

Let T be an optimal tree with root k_r .

$$\begin{aligned} C(T) &= p_r + \underbrace{\sum_{a=1}^{r-1} p_a (D(T, k_a) + 1)}_{\text{left subtree}} + \underbrace{\sum_{a=r+1}^n p_a (D(T, k_a) + 1)}_{\text{right subtree}} \\ &= p_r + \sum_{a=1}^{r-1} p_a + \sum_{a=1}^{r-1} p_a D(T, k_a) + \sum_{a=r+1}^n p_a + \sum_{a=r+1}^n p_a D(T, k_a) \\ &= \sum_{a=1}^n p_a + \sum_{a=1}^{r-1} p_a (D(T_{\text{left}}, k_a) + 1) + \sum_{a=r+1}^n p_a (D(T_{\text{right}}, k_a) + 1) \\ &= \sum_{a=1}^n p_a + C(T_{\text{left}}) + C(T_{\text{right}}) \end{aligned}$$



Recurrence: $C[i, j]$ = cost of optimal binary search tree on keys $k_i, \dots, k_j$.

$$C[i, j] = \begin{cases} 0 & \text{if } j < i \text{ (tree is empty)} \\ p_i & \text{if } i = j \text{ (tree is single node)} \\ \left(\sum_{a=i}^j p_a \right) + \min_r \{ C[i, r-1] + C[r+1, j] \} & \text{base cases} \end{cases}$$

↑ minimum over possible choices of root

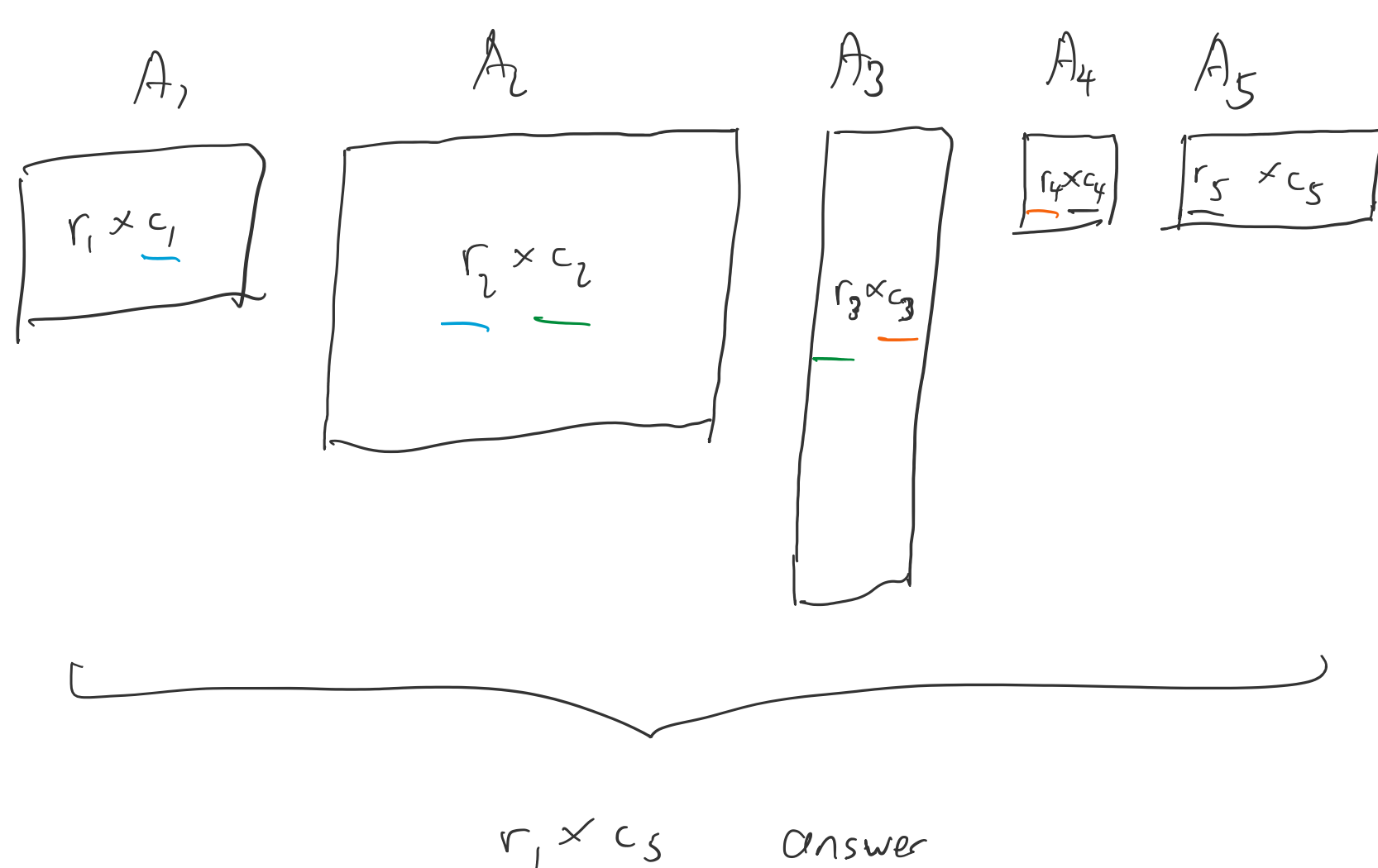
Want: $C[1, n]$. Can fill $C[i, j]$ matrix in increasing order of $j-i$.

$C[i, j]$

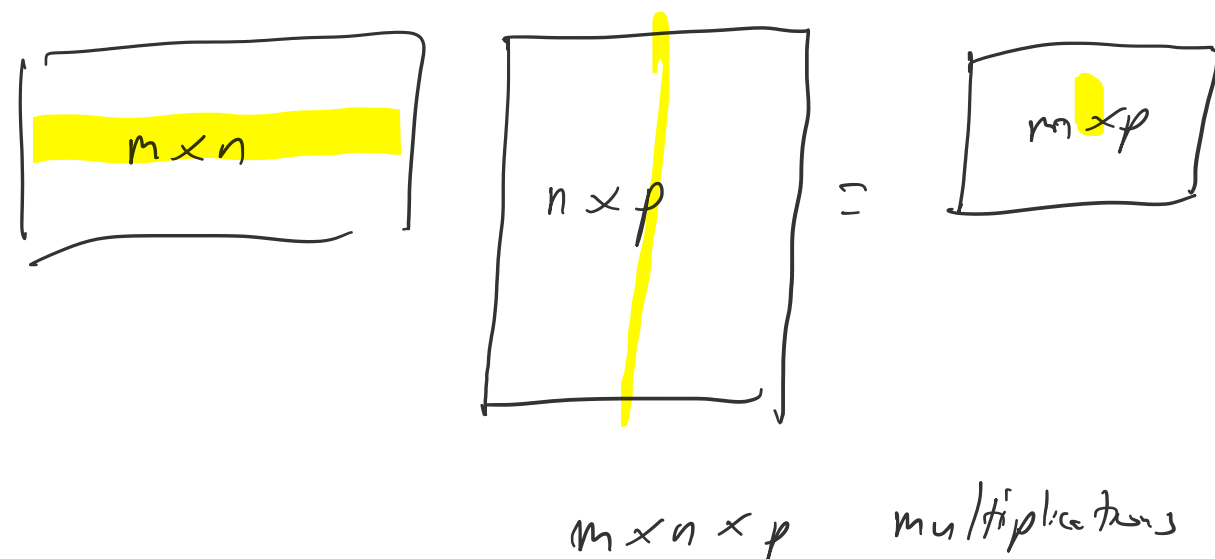
					p_6
				p_5	0
			p_4	0	0
		p_3	0	0	0
	p_2	0	0	0	0
p_1	0	0	0	0	0
i					j

Matrix-chain multiplication

Want to multiply matrices $A_1, A_2, \dots, A_n$:



Cost of multiplication:



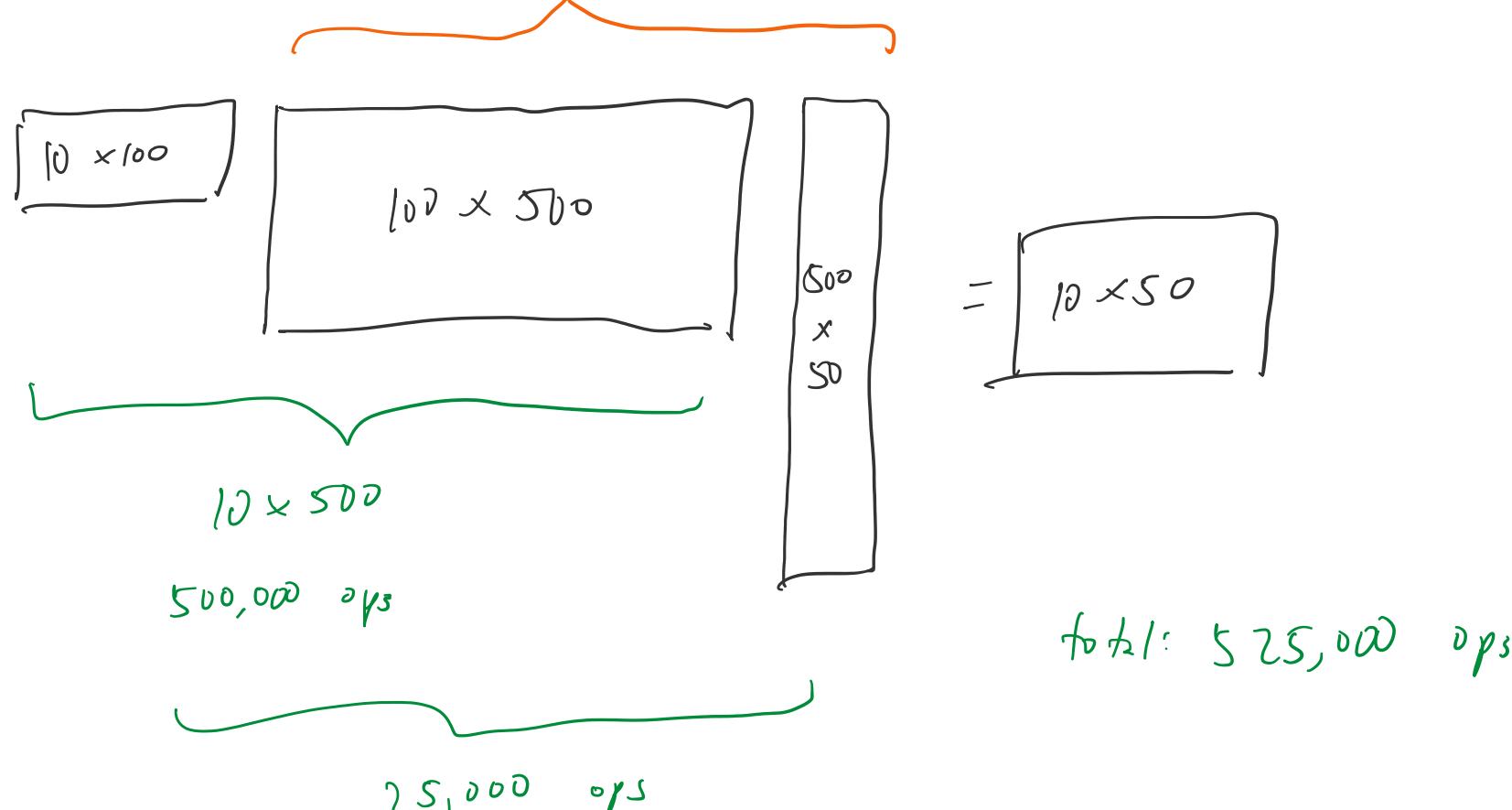
But $(A_1 A_2)(A_3 A_4)(A_5 A_6) = (A_1(A_2 A_3))(A_4(A_5 A_6)) = \dots$

All multiplication orders give same results but take different amounts of time.

50,000 ops $\rightarrow 100 \times 50 \rightarrow 2,500,000$ ops

total: 2,550,000 ops

Ex.



DP recurrence:

$$OPT(i, j) = \underbrace{OPT(i, t-1)}_{\substack{A_i \dots A_{t-1} \\ i=1 \dots t-1}} + \underbrace{OPT(t, j)}_{\substack{A_t \dots A_j \\ t=t \dots j}} + \text{multiplying the two matrices}$$

$$OPT(i, j) = \min_{i \leq t \leq j} \{ OPT(i, t-1) + OPT(t, j) + r_i c_{t-1} c_j \}$$

Base case: $OPT(i, i+1) = \text{mul}(A_i, A_{i+1}) = r_i c_i c_{i+1}$.