Lec23-more-DP

Tuesday, October 29, 2024 11:46 PM

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

distance from root to key k_i.

$$C(T) = \sum_{i=1}^{n} p_i \left(\text{Depth}(T, k_i) + 1 \right)$$

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.

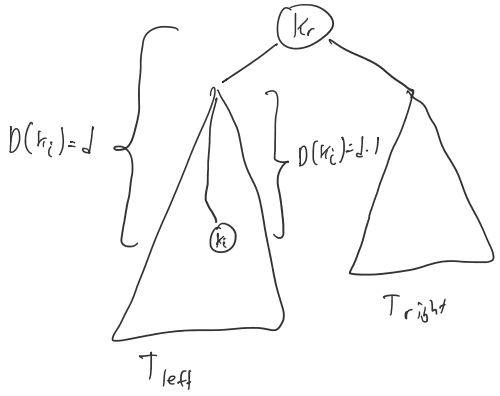
$$C(T) = p_r + \sum_{a=1}^{r-1} p_a \left(D(T, k_a) + 1 \right) + \sum_{a=r+1}^{n} p_a \left(D(T, k_a) + 1 \right)$$

left subtree — right subtree

$$= p_r + \sum_{a=1}^{r-1} p_a + \sum_{a=1}^{r-1} p_a D(T, k_a) + \sum_{a=r+1}^{n} p_a + \sum_{a=r+1}^{n} p_a D(T, k_a)$$

$$= \sum_{a=1}^{n} p_a + \sum_{a=1}^{r-1} p_a \left(D(T_{left}, k_a) + 1 \right) + \sum_{a=r+1}^{n} p_a \left(D(T_{right}, k_a) + 1 \right)$$

$$= \sum_{a=1}^{n} p_a + C(T_{left}) + C(T_{right})$$

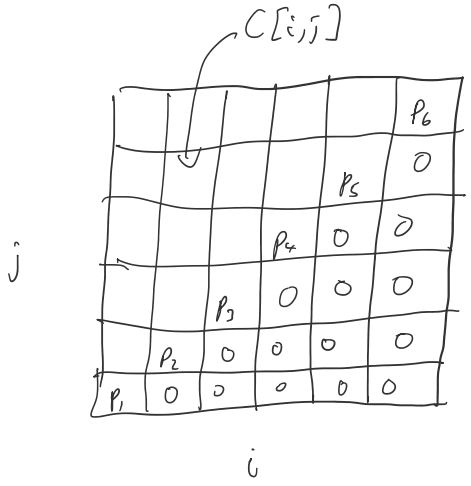


Recurrence: $C[i, j]$ = cost of optimal binary search tree on keys $k_i, \dots, k_j$.

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

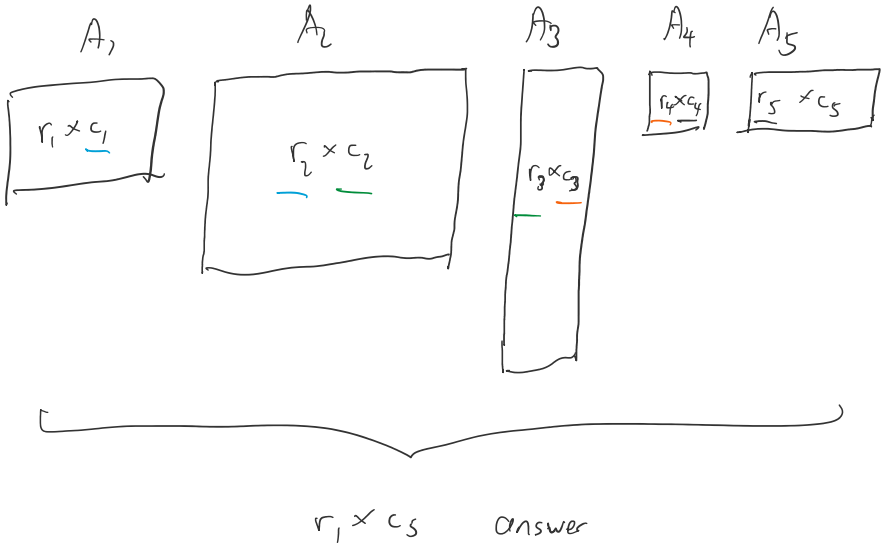
↑ minimum over possible choices of root

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

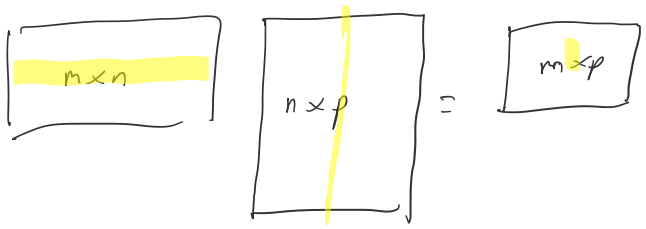


Matrix-chain multiplication

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



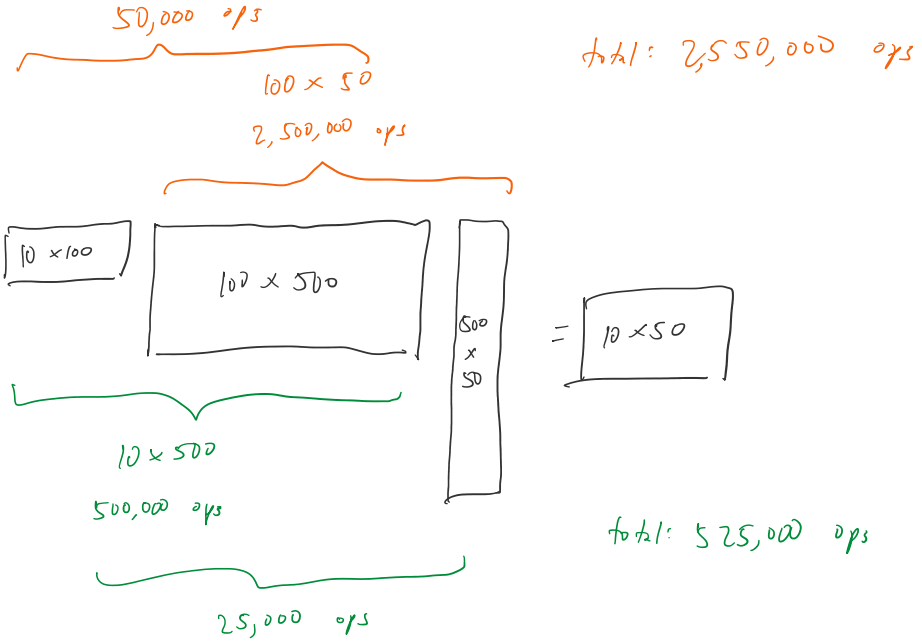
Cost of multiplication:



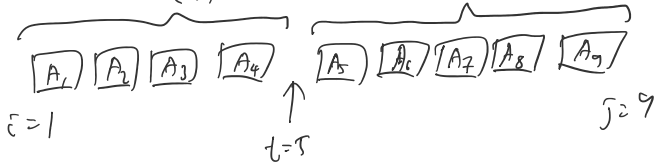
$m \times n \times p$ multiplications

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

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



DP recurrence: $OPT(i,j) = \min_{i \le t \le j} \{ OPT(i,t-1) + OPT(t,j) + \text{multiplying the two matrices} \}$



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

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