WEEK 12
LONGEST COMMON SUBSEQUENCE
MATRIX CHAIN MULTIPLICATION

2024-04-04

LONGEST COMMON SUBSEQUENCE

- Common subsequence between 2 sequences

c a s t

c h a r t

- Matching in an ascending order. Some objects might be skipped.

NOT THE LONGEST COMMON SUBSTRING!!!

c a s t

c h a r t

Longest common substring

Either C, A, or T alone

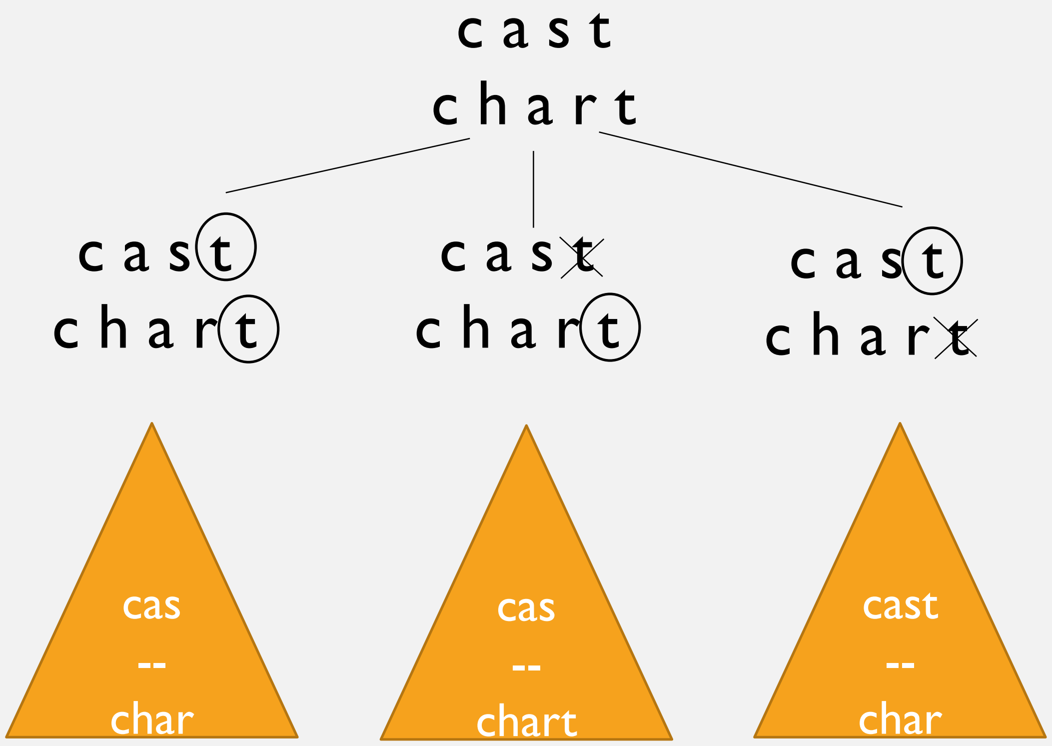
Longest common subsequence

All of CAT

WHEN DO WE USE L.C.S.?

- Find an update between two versions of documents
- Plagiarism check
- DNA sequence comparison

RECURSIVE BACKTRACKING



- Bird
  - Do we skip the last object of each sequence in a matching
    - Not skip at all
    - Skip for the first sequence
    - Skip for the second sequence
    - ~~Skip both (included in the two cases above)~~
- Friend
  - Find the longest common subsequence for the remaining parts of both sequences.

SUBPROBLEMS

- Friends has to solve the smaller problems
  - Find the longest common subsequence for the remaining parts of both sequences.
- If the last object for each sequence has been removed, friends have to solve LCS of the sequences with 1 object shorter.

	-	C	A	S	T
-					
C					
H					
A					
R					
T					

+1

Length of LCS for CHAR and CAS

Length of LCS for CHART and CAST

c a s t

c h a r t

TABLE FILLING

- Value in the table at i, j is the length of the longest possible common subsequence for the first i, j objects from sequence 1 and 2 respectively.
- Refer to different cases:
  - Not skip at all (use the last object in both sequences in the matching)
  - Skip for the first sequence
  - Skip for the second sequence

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0					
A	0					
S	0					
T	0					

Find the longest common subsequence between CAST and CHART

Smallest inputs are when one of the sequences is empty.

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0					
A	0					
S	0					
T	0					

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0	1	1	1	1	1
A	0					
S	0					
T	0					

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0	1	1	1	1	1
A	0	1				
S	0					
T	0					

- ~~Not skip at all~~ (unavailable since $H \neq A$)
- Skip for the first sequence
  - Refer to C and CA
- Skip for the second sequence
  - Refer to CH and C

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0	1	1	1	1	1
A	0	1	1			
S	0					
T	0					

- Not skip at all $(A = A)$
  - Refer to CH and C
  - Add one matching $(A = A)$
- Skip for the first sequence
  - Refer to CA and CH
- Skip for the second sequence
  - Refer to CHA and C

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0	1	1	1	1	1
A	0	1	1	2		
S	0					
T	0					

- ~~Not skip at all~~ $(R \neq A)$

- Skip for the first sequence
  - Refer to CHA and CA

- Skip for the second sequence
  - Refer to CHAR and C

MAIN EQUATION

$$opt_{i,j} = \max(opt_{i-1,j}, opt_{i,j-1}, opt_{i-1,j-1} + 1\ )$$

Exists only when $s_1[i] = s_2[j]$

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0	1	1	1	1	1
A	0	1	1	2		
S	0					
T	0					

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0	1	1	1	1	1
A	0	1	1	2	2	2
S	0					
T	0					

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0	1	1	1	1	1
A	0	1	1	2	2	2
S	0	1	1	2	2	2
T	0					

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0	1	1	1	1	1
A	0	1	1	2	2	2
S	0	1	1	2	2	2
T	0	1	1	2	2	3

TRACE FOR THE OPTIMAL MATCHING

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0	1	1	1	1	1
A	0	1	1	2	2	2
S	0	1	1	2	2	2
T	0	1	1	2	2	3

- If the value in the cell is the same as the one on top or on the left, it borrows the previous optimal matching with no progress.
- Progress is made only when the reference is from the diagonal direction.
  That is where the object is chosen.

LOOP INVARIANT

- Value in the table at i, j is the length of the longest possible common subsequence for the first i, j objects from sequence 1 and 2 respectively.

	-	C	H	A	R	T
-	0	0	0	0	0	0
C	0	1	1	1	1	1
A	0	1	1	2	2	2
S	0					
T	0					

Length of LCS for CA and CHAR

STRONG INDUCTION PROOF (SKETCH)

- Basic step
  - We have the optimal matching for the smallest input (when at least one sequence is empty).
- Inductive step
  - Assume that we have the solution for the optimal matching for every smaller subsequences in the table.
  - We have to show that the one-unit-longer input gets the optimal matching as well.

INDUCTIVE STEP

- Assume that we have the solution for the optimal matching for every smaller subsequences in the table.
- There are only 3 cases that the new value will be added to the table.
- In each case, there is a new object added to at least one of the sequences.
  - Both sequences have new object – only when these new objects are the same
  - First sequence has new object
  - Second sequence has new object
- Choose the max among all three cases, there is no other way to get better length.

TIME COMPLEXITY

- For two sequences with length m and n, the algorithm has to fill the table with size $(m+1)(n+1)$.
- Task in each step is at most 3, so it is constant.
- Time complexity = $O(mn)$

MATRIX CHAIN MULTIPLICATION

- Minimize cost of the multiplication, not to get the solution for real.
- Actually, it is working on the size of the matrix, not the multiplication process.

$$\begin{bmatrix} 1 & 3 \\ 2 & -4 \end{bmatrix} \times \begin{bmatrix} 2 & 5 & 7 \\ 4 & 0 & -2 \end{bmatrix} \times \begin{bmatrix} 1 \\ 4 \\ 6 \end{bmatrix} \times \begin{bmatrix} 3 & 7 & -2 & 9 \end{bmatrix}$$

RECALL: COST OF MATRIX MULTIPLICATION

- Multiply $A_{m \times n} \times B_{n \times r}$
- There are $m \times r$ elements in the product
- Each of them is the sum of n scalar product pairs.
- Cost of multiplication = $O(m \times n \times r)$

$$\begin{bmatrix} 2 & 5 & 7 \\ 4 & 0 & -2 \end{bmatrix} \times \begin{bmatrix} 1 \\ 4 \\ 6 \end{bmatrix}$$

ASSOCIATIVE PROPERTY OF MATRIX MULTIPLICATION

$$(A \times B) \times C = A \times (B \times C)$$

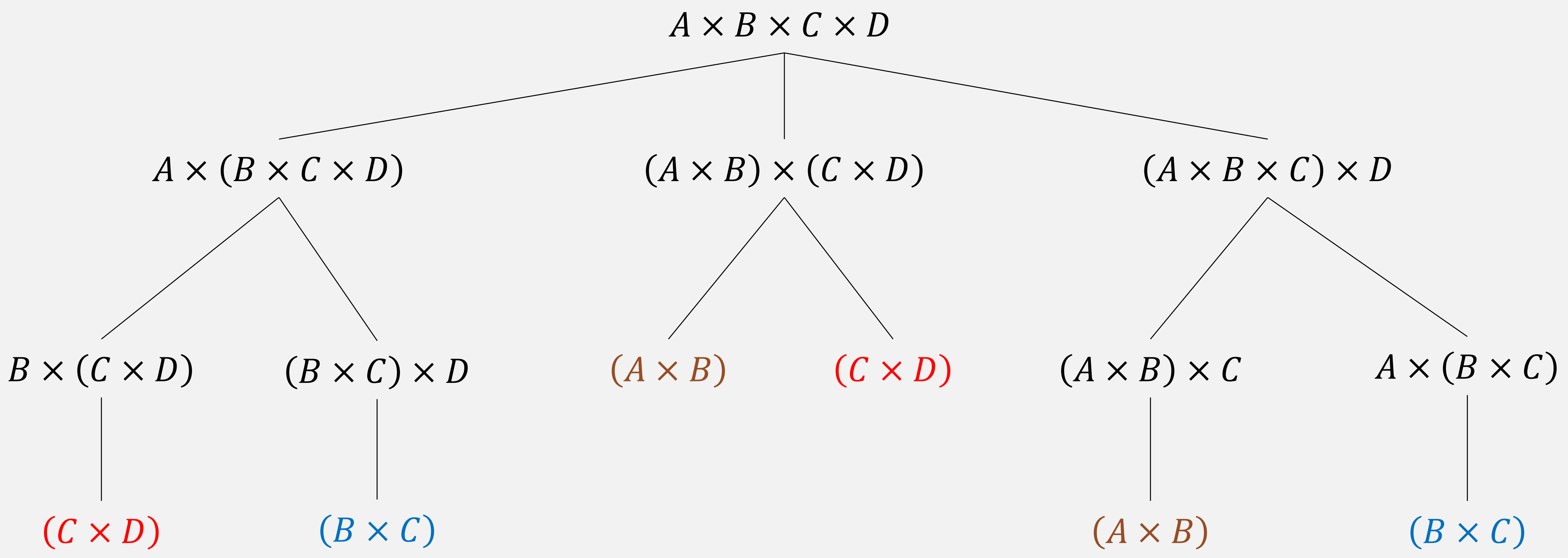
$$A \times B \times C \times D \ = A \times (B \times C \times D) = (A \times B) \times (C \times D) = (A \times B \times C) \times D$$

$$\underbrace{\underbrace{\begin{bmatrix} 1 & 3 \\ 2 & -4 \end{bmatrix} \times \begin{bmatrix} 2 & 5 & 7 \\ 4 & 0 & -2 \end{bmatrix}}_{\substack{\text{dim} = 2 \times 3 \\ cost = 2 \times 2 \times 3 = 12}} \times \begin{bmatrix} 1 \\ 4 \\ 6 \end{bmatrix}}_{\substack{\text{dim} = 3 \times 1 \\ cost = 12 + (2 \times 3 \times 1) = 18}}$$

$$\underbrace{\begin{bmatrix} 1 & 3 \\ 2 & -4 \end{bmatrix} \times \underbrace{\begin{bmatrix} 2 & 5 & 7 \\ 4 & 0 & -2 \end{bmatrix} \times \begin{bmatrix} 1 \\ 4 \\ 6 \end{bmatrix}}_{\substack{\text{dim} = 2 \times 1 \\ cost = 2 \times 3 \times 1 = 6}}}_{\substack{\text{dim} = 3 \times 1 \\ cost = 6 + (2 \times 2 \times 1) = 10}}$$

Good strategy: If we can wrap the larger dimension in between two smaller size dimensions, the cost can be reduced.

DIFFERENT WAYS TO GET CHAIN MULTIPLICATION



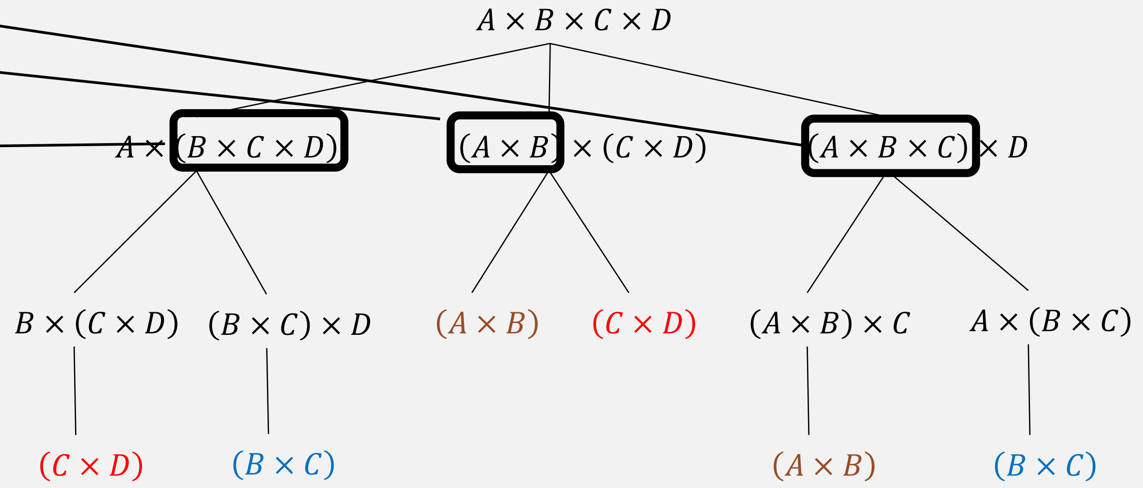
Redundancies of shorter chain multiplications

SUBPROBLEMS

- Once we found the optimal multiplication for first k matrices that is a part of the chain multiplication, we keep it.
- Don't forget that there must be the latter part of the chain that we must take into account in order to get the entire chain.
- Table storing the cost must have all the optimal cost for any chain partition.
- Row i and column j stores the optimal cost for the chain multiplication started at matrix i and ends up with matrix j.

OPTIMAL COST FOR SUBPROBLEMS

	A	B	C	D
A				
B				
C				
D				



OPTIMAL COST TABLE

	A	B	C	D
A				
B				
C				
D				

Row i and column j stores the optimal cost for the chain multiplication started at matrix i and ends up with matrix j.

- Only the upper half is needed
- The main diagonal is the cost of the chain having only one matrix. Then cost = 0.
- One line above the main diagonal contains the cost for the chain having 2 matrices.

REFERENCE TABLE

	A	B	C	D
A				
B				
C				
D				

- Store the ending position of the first component that has been chosen to the calculation of the optimal cost.

$$\begin{bmatrix}1 & 3\\2 & -4\end{bmatrix} \times \begin{bmatrix}2 & 5 & 7\\4 & 0 & -2\end{bmatrix} \times \begin{bmatrix}1\\4\\6\end{bmatrix} \times \begin{bmatrix}3 & 7 & -2 & 9\end{bmatrix}$$

$$A_{2\times 2} \times B_{2\times 3} \times C_{3\times 1} \times D_{1\times 4}$$

	A	B	C	D
A	0			
B		0		
C			0	
D				0

	A	B	C	D
A	0			
B		1		
C			2	
D				3

$$A_{2\times2} \times B_{2\times3} \times C_{3\times1} \times D_{1\times4}$$

	A	B	C	D
A	0			
B		0		
C			0	
D				0

- Optimal cost for $A_{2\times2} \times B_{2\times3}$
- There is only one way to do it.
- Cost $= 2 \times 2 \times 3 = 12$

	A	B	C	D
A	0	0		
B		1		
C			2	
D				3

$$A_{2\times2} \times B_{2\times3} \times C_{3\times1} \times D_{1\times4}$$

	A	B	C	D
A	0	12		
B		0		
C			0	
D				0

- Optimal cost for $B_{2\times3} \times C_{3\times1}$
- There is only one way to do it.
- Cost $= 2 \times 3 \times 1 = 6$

	A	B	C	D
A	0	0		
B		1	1	
C			2	
D				3

$$A_{2\times2} \times B_{2\times3} \times C_{3\times1} \times D_{1\times4}$$

	A	B	C	D
A	0	12		
B		0	6	
C			0	12
D				0

	A	B	C	D
A	0	0		
B		1	1	
C			2	2
D				3

$A_{2\times2} \times B_{2\times3} \times C_{3\times1} \times D_{1\times4}$

	A	B	C	D
A	0	12		
B		0	6	
C			0	12
D				0

$(A \times B \times C)$

$(A \times B) \times C$ $\quad$ $A \times (B \times C)$

- $(A \times B)_{2\times3} \times C_{3\times1}$
  - Additional cost = $2 \times 3 \times 1 = 6$
  - Total cost = $Cost_{A\times B} + Cost_C + Cost_{(A\times B)\times C}$
    $= 12 + 0 + 6 = 18$
- $A_{2\times2} \times (B \times C)_{2\times1}$
  - Additional cost =
  - Total cost =

$A_{2\times2} \times B_{2\times3} \times C_{3\times1} \times D_{1\times4}$

	A	B	C	D
A	0	12	10	
B		0	6	
C			0	12
D				0

$(A \times B \times C)$

$(A \times B) \times C$

Total cost $= 18$

$A \times (B \times C)$

Total cost = 10

	A	B	C	D
A	0	0	0	
B		1	1	
C			2	2
D				3

The end of the first partition is the matrix index 0

$$A_{2\times 2} \times B_{2\times 3} \times C_{3\times 1} \times D_{1\times 4}$$

	A	B	C	D
A	0	12	10	
B		0	6	
C			0	12
D				0

$(B \times C \times D)$

$(B \times C) \times D$ $\quad\quad$ $B \times (C \times D)$

$$A_{2\times 2} \times B_{2\times 3} \times C_{3\times 1} \times D_{1\times 4}$$

	A	B	C	D
A	0	12	10	
B		0	6	14
C			0	12
D				0

$(B \times C \times D)$

$(B \times C) \times D$

Cost = 14

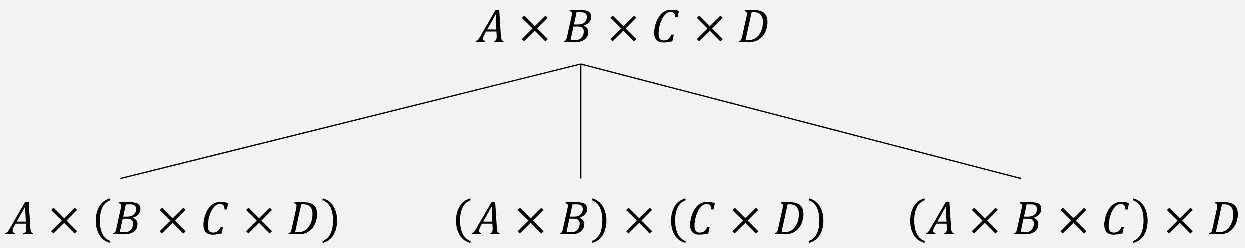
$B \times (C \times D)$

Cost = 36

	A	B	C	D
A	0	0	0	
B		1	1	2
C			2	2
D				3

The end of the first partition is the matrix index 2

$$A_{2\times 2} \times B_{2\times 3} \times C_{3\times 1} \times D_{1\times 4}$$



	A	B	C	D
A	0	12	10	
B		0	6	14
C			0	12
D				0

$$A_{2\times 2} \times B_{2\times 3} \times C_{3\times 1} \times D_{1\times 4}$$

$$A \times B \times C \times D$$

$A \times (B \times C \times D)$ | $(A \times B) \times (C \times D)$ | $(A \times B \times C) \times D$

- 30 | 48 | 18

	A	B	C	D
A	0	12	10	18
B		0	6	14
C			0	12
D				0

	A	B	C	D
A	0	0	0	2
B		1	1	2
C			2	2
D				3

TRACE FOR THE MULTIPLICATION PATH

	A	B	C	D
A	0	0	0	2
B		1	1	2
C			2	2
D				3

Best partition for ABCD is after position 2

$$(A \times B \times C) \times D$$

Then look-up for ABC.

Best partition for ABC is after position 0

$$A \times (B \times C)$$

Then look-up for BC

Best partition for BC is after position 1

$$B \times C$$

TIME COMPLEXITY

- There are $\frac{n(n+1)}{2}$ cells in the table to fill
- In each cell, there are at most n cases to compare
- Time complexity = $O(n^3)$

ASSIGNMENT 5

- Generate 2 similar sequences with at least 6 elements each. Solve for the longest common subsequence algorithm
  - Fill the optimal matching table
  - Trace for the LCS.