Rubayet Mam

18.10.23

MCM

→ square matrix এর জন্য → mcm algo apply করতে পারব না,

$A_1 \rightarrow p \times q$

$A_2 \rightarrow q \times r$

$A_1\ A_2\ A_3\ A_4$

$((A_1A_2A_3)A_4)$ $((A_1A_2)(A_3A_4))$ $(A_1(A_2A_3A_4))$

$A_1\ A_2\ A_3$

$((A_1A_2)A_3)$ $(A_1(A_2A_3))$

$i \quad j \qquad i \quad j$

$A_1\ A_2\ A_3$

mcm of (1, 2, 3)

$i=1 \quad j=3$

$A_i, A_{i+1}, A_{i+2} \ldots A_j$

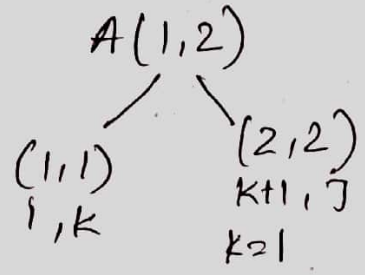
A_i ——— A_j

A_k

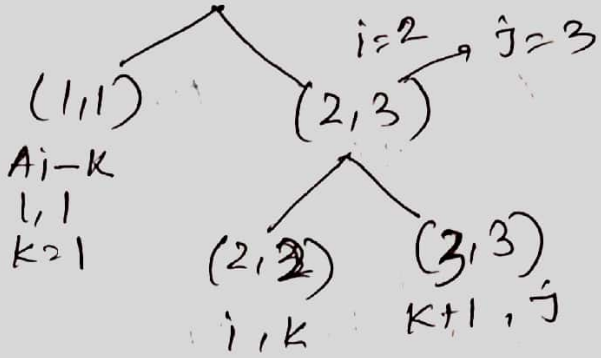
$(A_i \ldots A_{i+1} \ldots A_k)$

$(A_{k+1} \ldots A_{k+2} \ldots A_j)$

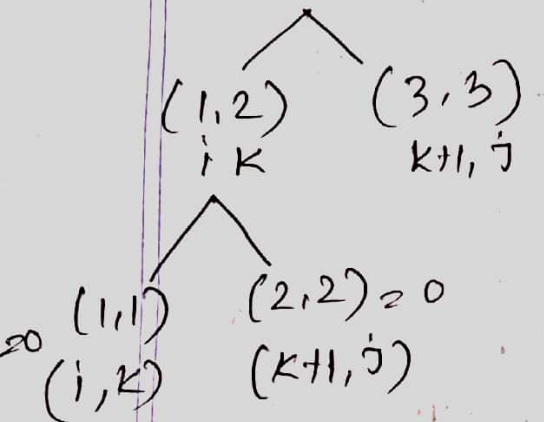
A(1,2) mcm(1,2)



m(1-3)



m(1-3)



A = 4×5
B = 5×6
C = 6×7
~~Pi-1 0~~

k=1

P_{i-1} P_k P_j

P_0 P_1 P_3

= 4 × 5 × 7

k=2

P_0 P_2 P_3

4 × 6 × 7

mcm → optimal + overlap

↓

সহজে এই Property maintain করে তাহলে dynamic Programming.

23.10.23

exponential time of n

$A_i \cdots A_k$ $\quad$ $A_{k+1} \cdots A_j$

$T(K)$ $\quad$ $T(n-k)+1$

$$2\sum_{i=1}^{n-1} T(i) + n$$

$P_{i-1}\ P_k\ P_j$

$T(1) \geq 1 = 2^0$

$$1 = 2^0$$

$$2\sum_{i=1}^{n-1} 2^i + n$$

$$= 2(2^{n-1}) + n$$

$$= n2^n + n \geq 2^{n-1}$$

mcm 2^{n-1}

$k = i$ to $j-1$

cost = mcm (i,k) + mcm $(k+1, j)$ + $P_{i-1}\ P_k\ P_j$

if cost $< m[i,j]$

$m[i,j]$ = cost

memoization এ value store করে রাখা তাই recursive call হবে কিন্তু time complexity কম হবে।

Time complexity : memoization

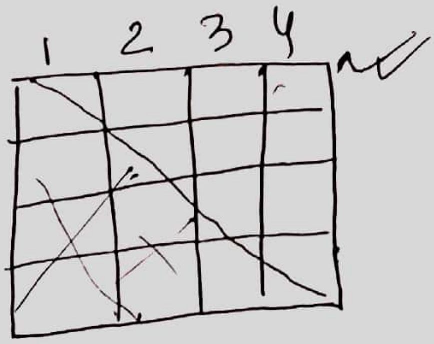
$O(n^2) \cdot n = O(n^3)$

recursive, $T = O(2^n)$

408 Pg → Fig - 15.7

recursive algo use করে time complexity - বেশি
memoization " " " " → কম

→ Illustration



378 Pg 15.2-2, (3,4) → recursive

15.2-5 → বাদ

mcm (1,3) (i, j)

(1,3)
(1,1) (2,3)
A1
(i, s[i,j]) (2,2) (3,3) (s[i,j], +1, j)
A2 A3

or, (1,3)
(1,2) (3,3)

(1×2) 3

i, j সমান না হওয়া পর্যন্ত recursive call

(A1 (A2 A3))

15.3-2 →
merge sort → sub problem property নাই

381 → Page - slide - 402

CS CamScanner

diff → greedy, dp and divide and conquer

কেন আমরা bottom up use করব কেন করব না top down না ?

06/11/23

lab – Sazia maam

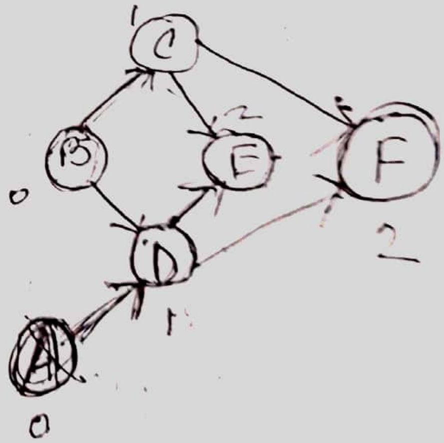
Topological Sort

↳ 1st check → cycle directed আছে কিনা। topological sort
→ cycle থাকলে হবে না।

map = {A=0, B=0, C=0, D=1, E=2, F=2}

যাদের 0 তাদের queue তে নিব।

1. pop
2. neighbour দের 1 কমাবো
3. যাদের 0 তাদের queue তে push



A

ordering: A, B

DFS এর মাধ্যমে
→ directed acyclic graph

① প্রতিটা vertices এর indegree বের করতে হবে। (incoming edge)

② এ " Print যার indegree 0 → A

print: A B C E D F

A এর কোন incoming edge নাই