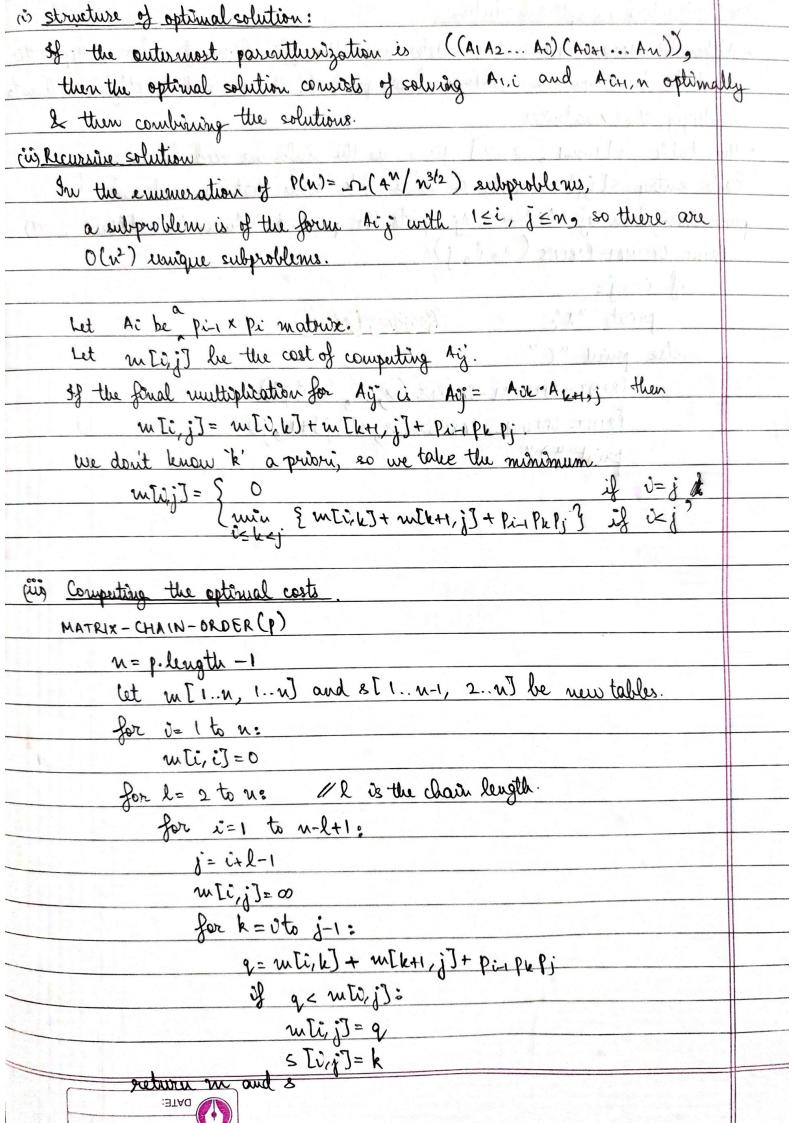
16.04.2020 18. Optimal Matrix chair Multiplication A product of matrices is July paranthesized of it is either · a single matrix, or · a product of 2 July parenthesized matrices, surrounded by pare ritheses. Each parenthesization defines a set of n-1 matrix multiplications. we just need to pick the parenthesization that corresponds to hest ordering There are 20cm ways to parenthense the matrix: Trying out all possibilities is a lad idea, or P(W)= 12 (4 / n3/2) Dynamic Programming steps: 1. characterize the structure of an optimal solution. 2. Recursively define the value of an optimal solution. 3. Compute the value of an optimal solution bottom-up. 4. Construct an optimal solution from the computed also



cir Countructing an aptimal solution.

Although Mareix-Chain-Droek determines the optimal no. of scalar multiplocations needed to compute a matrix-chain product, it does not directly show how to multiply the matrices.

The table s[1...,1, 2...n] gives us the trifo we need to do so.

Each entry s[ij] records a value of k such that an optimal parentherization of A: Ain... A; splits the product between Ak and Akn.

PRINT-OPTIMAL-PARENS (2, i, j)

if i==j:

print "A";

else prind "("

PRINT-OPTIMAL-PARENS (s, i, s[i,j])

PRINT-OPTIMAL-PARENS (s, s[i,j]+1,j)

print ")"