נניח שיש לנו רצף או שרשור A1, A2, …, A*n*   
של n מטריצות שיש להכפיל.

כלומר, אנו רוצים לחשב את התוצאה A1A2…A*n.*

ישנן דרכים רבות אפשריות (הצבת סוגריים שונות סוגריים) לחשב את התוצאה.

**לדוגמא**, בהינתן רצף המטריצות: A1, A2, A3, A4 של 4 מטריצות-

בואו נחשב את התוצאה של A1A2A3A4:

* יש 5 דרכים אפשריות לבצע את זה-
  + (A1(A2(A3A4)))
  + (A1((A2A3)A4))
  + ((A1A2)(A3A4))
  + ((A1(A2A3))A4)
  + (((A1A2)A3)A4)

**בעיית כפל רצף מטריצות** -

* + בהינתן שרשור של n מטריצות, A1, A2, …, A*n*   
    כאשר לכל מטריצה *i*=1, 2, …, *n*, מימדי המטריצה A*i* הם *pi*-1×*pi*
  + מחפשים את החלוקה של השרשור כך שביצוע הכפל ייתן מספר מינימלי דרוש של מכפלות סקלריות לביצוע המכפלה של A1A2…A*n*

חיפוש של כל האפשרויות ובדיקה מה הכי משתלם לוקח זמן מעריכי, לא רלוונטי.

**מבנה פתרון אופטימלי -**

* + נסמן ב Ai..j את המטריצה הנוצרת מהכפלת המטריצות Ai Ai + 1… Aj
  + הכפלה אופטימלית של המטריצות A1A2…A*n* מחלקת את רצף המטריצות לשני רצפים, בין A*k* לבין A*k*+1 , עבור מספר שלם k כלשהו 1 ≤ *k* < *n :*

**A1A2…A*n=* (A1A2…Ak)(Ak+1….A*n*)**

* + לחישוב התוצאה מחשבים את התוצאה של A1..*k*   
    ואת התוצאה של A*k*+1..*n* ,   
    אחר כך ניתן לחשב את התוצאה הסופית, A1..*n* .

**A*i*..*j* = (A*i* A*i*+1…A*k*)·(A*k*+1A*k*+2…A*j*)= A*i*..*k* · A*k*+1..*j***

עלות חישוב כפל המטריצותA*i*..*j* יהיה:

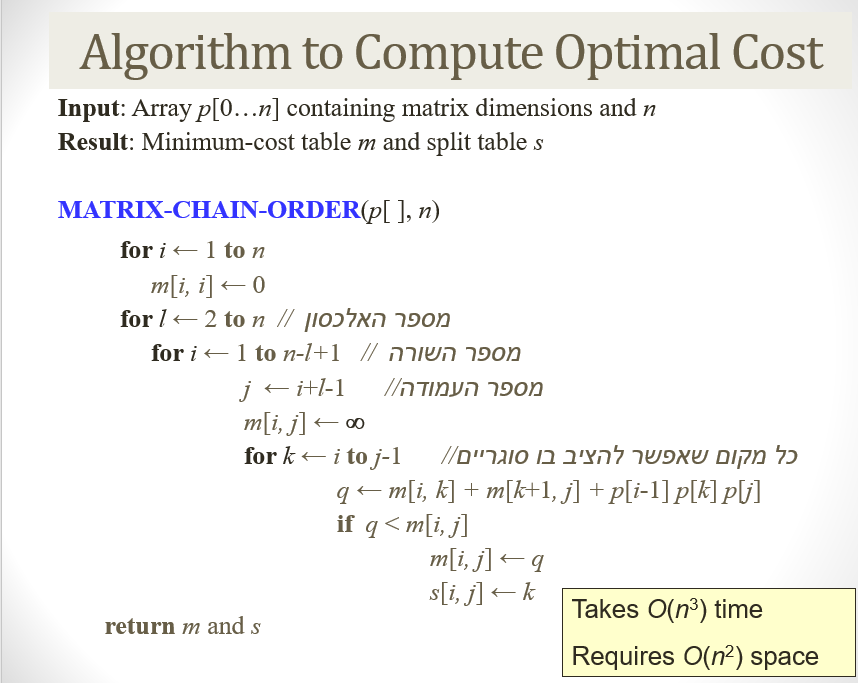
* + - עלות החישוב של A*i*..*k*
    - עלות החישוב של A*k*+1..*j*
    - עלות ההכפלה של המטריצות A*i*..*k* עם A*k*+1..*j*  ששווה ל *pi*-1*pk pj*
  + *m*[*i*, *j* ] = *m*[*i*, *k*] + *m*[*k*+1, *j* ] + *pi*-1*pk pj    
     for i* ≤ *k* < *j*
  + *m*[*i*, *i* ] = 0 for *i*=1,2,…,*n*

אך ההכפלה האופטימלית תתקבל עבור k כלשהו אפשרי בטווח: *i* ≤ *k* < *j,* ***לא ידוע איזה k!!*** ולכך יש לבדוק כל k אפשרי ולבחור את האופטימלי.

**M[i,j] = if i == j : 0 , else if i < j : *min* {*m*[*i*, *k*] + *m*[*k*+1, *j* ] + *pi*-1*pk pj* }** **for i** ≤ **k < j**

כדי לעקוב אחר בניית פתרון אופטימלי, משתמשים בטבלאות.

*s*[*i*, *j* ] יהיה הערך של *k* כך שפיצול הרצף- A*i* A*i*+1 … A*j* אחריו ייתן מכפלה אופטימלית.



אומנם MATRIX-CHAIN-ORDER קובע את המספר האופטימלי של המכפלות, אך הוא אינו מראה ישירות כיצד להכפיל את המטריצות.

לא קשה לבנות פתרון אופטימלי מהמידע המחושב המאוחסן בטבלה s[1..n, 1..n] .

כל תא s [i, j] מתעד את הערך של k כך שהסוגריים האופטימליים של Ai Ai+1 ··· Aj מפצלים את הרצף בין Ak לבין Ak+1.

ניתן לחשב את מספר המכפלות הנדרשות באופן רקורסיבי.

הפונקציה הרקורסיבית שלהלן מדפיס את החלוקה האופטימלית של  
 Ai, Ai+1, ..., Aj,

* + בהינתן בטבלת ה- s המחושבת על ידי  
     MATRIX-CHAIN-ORDER והערכים i ו- j.
  + הקריאה הראשונית ל:PRINT-OPTIMAL-PARENS(s, 1, n) תדפיס את הצבת הסוגריים האופטימלית ל-A1, A2, ..., An.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי