8.12.16 בס"ד

אלגוריתמים - הרצאה 4

תכנות דינאמי:

בשיטת הפרד ומשול ישנה בעיה שלעתים נדרש בעץ הרקורסיה לחשב שוב ושוב את אותן תתי בעיות.

שיטת תכנון דינאמי נועדה לפתור בעיה זו ע"י שמירת פתרונות לתתי בעיות שכבר חושבו במבנה נתונים מתאים.

חישוב מספרי פיבונאצ'י:

נזכיר שסדרת פיבונאצ'י מוגדרת כך: $F(0)=0 \; ; \; F(1)=1$

$F(n)=F(n-1)+F(n-2)$

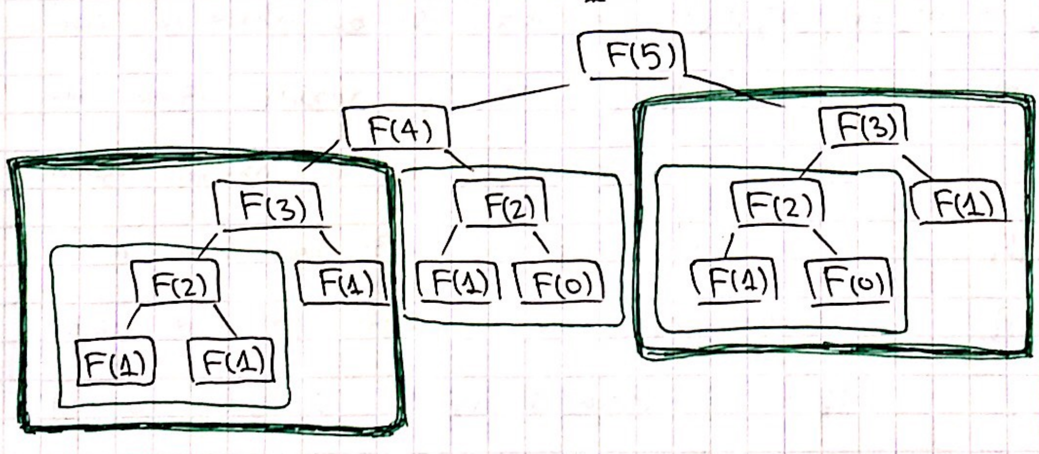
אלגוריתם רקורסיבי לחישוב $F(n)$ הינו:

```
F(n){
    if (n==0){ return 0;}
    if (n==1){return 1;}
    return F(n-1)+F(n-2); }
```

ניתוח זמן הריצה: $T(n)=T(n-1)+T(n-2)+O(1)$

$T(0)=O(1)$

כפי שראינו: $T(n)=\Omega(2^{\frac{n}{2}})=\Omega(\Phi^n)$



נסתכל על עץ החישוב של $F(5)$ הנתון למעלה.

את $F(2)$ חישבנו 3 פעמים, את $F(3)$ חישבנו פעמיים ← בזבוז!

בשיטת תכנות דינאמי נחשב את ערכי הפונק' מלמטה למעלה.

כלומר, נתחיל מחישוב הערכים של $F(1), F(0)$ בהתאם לתנאי העצירה, ואז נחשב את $F(2)$ כסכום שלהם, כך נמשיך הלאה לחישוב $F(3)$, $F(4)$ וכן, עד שנחשב $F(n)$.

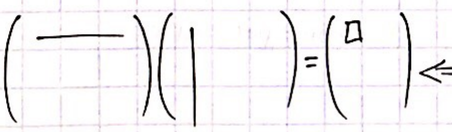
Scanned by CamScanner

בס"ד.

ערכים שכבר חישבנו נשמור במערך באורך n וכך לא נצטרך לחשב אותם מהתחלה בכל פעם. זמן הריצה של הפתרון לחישוב F(n) יהיה O(n) והמקום הדרוש הוא O(n), ניתן להסתפק רק ב-O(1) מקום ע"י שמירת שני הערכים האחרונים של F בלבד, (רק הם נחוצים לחישוב הבא)

```
F(n){
    if (n==0){return 0;}
    if (n==1){return 1;}
    x ← 0;  y ← 1;
    for (i=2 to n)
        temp ← y;
        y ← x+y;
        x ← temp;
    return y;
```

מכפלת שרשרת מטריצות:

מטריצות A ו-B מקיימות $\underset{p\times q}{A}\circ\underset{q\times r}{B}=\underset{p\times r}{C}$ ⇐ 

עלות כפל מטריצות A ו-B באופן נאיבי היא $p\cdot q\cdot r$.

בכפל של מטריצות יש אסוציאטיביות: $(m_1\cdot m_2)\cdot m_3 = m_1\cdot(m_2\cdot m_3)$

אם מימדי המטריצה m_1 הם $P_0\times P_1$ ומימדי המטריצה m_2 הם $P_1\times P_2$ ומימדי המטריצה m_3 הם $P_2\times P_3$, אזי:

החישוב מאגף שמאל ידרוש את מספר הפעולות הבא: $P_0P_1P_2 + P_0P_2P_3$

החישוב מאגף ימין ידרוש את מס' הפעולות הבא: $P_1P_2P_3 + P_0P_1P_3$

נניח ש- $P_0=100$; $P_1=3$; $P_2=100$; $P_3=3$

⇐ החישוב משמאל ידרוש: $30{,}000 + 30{,}000 = 60{,}000$

החישוב מימין ידרוש: $900 + 900 = 1{,}800$

לכן, יש משמעות רבה למיקום הסוגריים.

Scanned by CamScanner

8.12.16

בעיה 8: מכפלת שרשרת המטריצות.

קלט: מימדי סדרת מטריצות $m_1,m_2,\dots,m_n$ כאשר מימדיהם $(P_0,P_1),(P_1,P_2),\dots,(P_{n-1},P_n)$

פלט: סידור סוגריים אופטימלי, כלומר, סידור סוגריים אסוציאטיביות רק שמספר הפעולות הנדרשות לחישוב מכפלת המטריצות $m_1,\dots,m_n$ יהיה המינימלי מבין כל הסידורים האפשריים

← בשלב ראשון נתמקד במציאת מס' הפעולות המינימלי הדרוש לחישוב המכפלה.

הצעות אלגוריתמיות:

1) הפתרון הנאיבי -

(*) עבור על כל סידורי הסוגריים האפשריים.

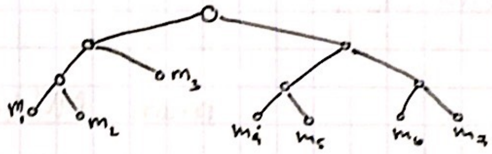
(*) עבור כל סידור נחשב את מס' הפעולות הנדרשות לחישוב המכפלה.

(*) נחזיר את הערך המינימלי המתקבל עבור סידור מבין כל הסידורים.

כמה סידורי סוגריים אפשריים?

$((m_1\cdot m_2)m_3)((m_4 m_5)(m_6 m_7))$

נראה שכל סידור סוגריים אפשרי ניתן לתאר בעזרת עץ בינארי מלא.



כמה עצים בינאריים בעלי n עלים (מלאים) קיימים? מספר זה נקרא מספר קטלן.

המספר קטלן ה-n הוא $C_n=\frac{1}{n+1}\binom{2n}{n}\sim\frac{4^n}{n^{3/2}\cdot\sqrt{\pi}}$

לכן זמן הריצה של הפתרון הנאיבי הוא: $T(n)=O\left(\frac{n\cdot 4^n}{n^{3/2}\cdot\sqrt{\pi}}\right)=O\left(\frac{4^n}{\sqrt{n}}\right)$

2) פתרון רקורסיה (חכם)

בשלב האחרון של החישוב מחשבים את המכפלה: $(m_1,\dots,m_i)(m_{i+1},\dots,m_n)$

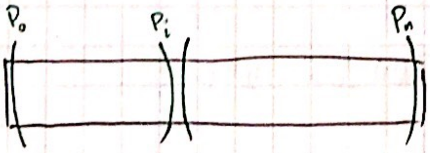
מכיוון שאיננו יודעים איפה המקום ה-i הכי טוב, נצטרך לעבור על כל האפשרויות.

נגדיר פונק' M, כאשר $M(1,n)$ יהיה מספר הפעולות המינימלי הדרוש לחישוב המכפלה $m_1,\dots,m_n$.

באופן כללי, נגדיר $M(i,j)$ להיות מספר הפעולות המינימלי הדרוש לחישוב המכפלה $m_i,m_{i+1},\dots,m_j$.

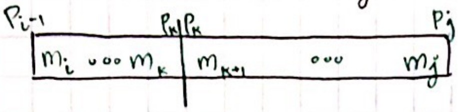
מתקיים - $\cancel{M(1,n)=\min(M(1,i),M(i+1,n))}$

$$M(1,n)=\min_{1\le i\le n}\left(M(1,i)+M(i+1,n)+P_0\cdot P_i\cdot P_n\right)$$



Scanned by CamScanner

באופן כללי :
$$m(i,j) = \min_{1 \le k \le j-1} \left(M(i,k) + M(k+1,j) + P_{i-1} P_k P_j \right)$$



תנאי עצירה : $i=j \Rightarrow M(i,j)=0$

אלגוריתם למימוש נוסחה זו :

```
MVL(i,j)
   if(i==j)  return 0;
   else
     min ← ∞
     for (k=i to j-1)
        cur ← MVL(i,k) + MVL(k+1,j) + P[i-1]·P[k]·P[j]
        if (cur<min) {min ← cur;}
   return min;
```

כאשר הקריאה החיצונית תהיה - `return MVL(1,n);`

ניתוח זמן ריצה : נגדיר את $T(n)$ להיות הזמן הדרוש לחישוב הפתרון האופטימלי למכפלת n מטריצות.

$$T(n) = \begin{cases} 1 & , n=1 \\ \sum_{k=1}^{n-1} \left(T(k) + T(n-k) + \underbrace{c}_{\text{קבוע}} \right) & , n>1 \end{cases}$$

איך מחשבים את $T(n)$?!

$$T(n) = \sum_{k=1}^{n-1} \left(T(k) + T(n-k) + c \right) = 2 \sum_{k=1}^{n-1} T(k) + (n-1)c$$

$$\begin{array}{ccc} T(1) & T(n-1) & c \\ T(2) & T(n-2) & c \\ \vdots & \vdots & \vdots \\ T(n-1) & T(1) & c \end{array} \Big\} \text{ פעמים } n-1$$

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

$$T(n) - T(n-1) = 2T(n-1) + c$$

$$T(n) = 3T(n-1) + c = 9T(n-2) + nc \ge 9T(n-2) = 3^i T(n-i) = 3^{n-1}$$

$\Leftarrow$ קיבלנו $T(n) = \Omega(3^n)$, שזה זמן ריצה אקספוננציאלי!

Scanned by CamScanner

8.12.16

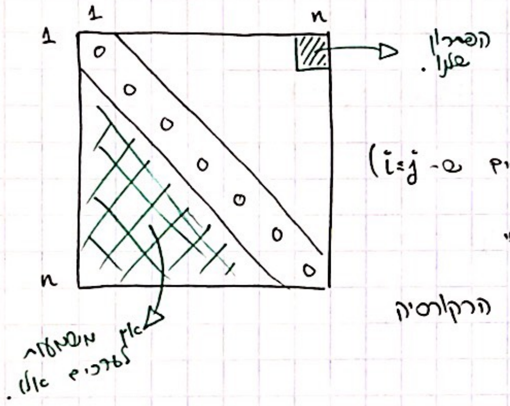
3) פתרון בתכנון דינמי -

נפתור את הבעיה של החישובים החוזרים ע"י תכנות דינמי.

לשם כך, ניצור טבלה בגודל nxn שתשמור את הערכים של M של הפונקציה שכבר חישבנו.

באלכסון נשמור את האפסים.

למה? כי בעצם תנאי העצירה שלנו, מצב בו $i=j$ אזי $\Leftarrow m=0$.



נשים ♡:

אין משמעות לערכים מתחת לאלכסון הראשי.
(כיוון שתמיד הולה $i \geq j$, ואנו מחפשים רק מקרים ש- $i \leq j$)

כעת נמלא את הטבלה אלכסון - אלכסון מהאלכסון הראשי

ועד לפינה הימנית - העליונה של הטבלה בהתאם לנוסחת הרקורסיה

והערכים שכבר חושבו קודם ונשמרו בטבלה.

(חשוב לשים לב שלא מתבצעות קריאות רקורסיביות אלא נעזרים בערכים מהטבלה בלבד)

הפתרון הסופי לבעיה נמצא בפינה הימנית עליונה בתא $M[1,n]$

נשים ♡:

קיימת צורה של עבודה, נסתכל על השלבים:

1) ננתח את הפתרון הנאיבי
2) ננתח את הפתרון הרקורסיבי
3) נבנה את הפתרון בתכנות דינמי.
← נחליט על איזה מבנה נתונים אנו מסתכלים.
← נפרט את אופן המילוי שלו.
← היכן נמצא בו הפתרון הסופי של הבעיה?

פסאודו קוד:

```
MVL(P,n)
    for(i=1 to n)
        m[i,i] ← 0;          ← נמלא את האלכסון של טבלה m באפסים.
    for(diff=1 to n-1)
        for(i=1 to n-diff)
            j ← i+diff ; min ← ∞
            for(k=i to j-1)
                cur ← M[i,k] + M[k+1,j] + P[i-1]·P[k]·P[j];
                if (cur < min)
                    min ← cur
            m[i,j] ← min
    return m[1,n];
```

בסוף החזרנו את התא בפינה הימנית העליונה.

Scanned by CamScanner

בס"ד.

זמן ריצה: $O(n^3)$ (תשובה עם אמורה להגיע כי יש 3 לולאות שרצות
כסדר גודל של n אחת בתוך השניה בתוך השלישית)

סיבוכיות מקום: $O(n^2)$ ← שמרנו בתוך טבלה m בגודל nxn.

דוגמת ריצה:

$m_1 \rightarrow 15 \times 5$

$m_2 \rightarrow 5 \times 10$

$m_3 \rightarrow 10 \times 20$

$m_4 \rightarrow 20 \times 25$

הקלט שלנו יהיה: $\Longrightarrow$

P [15|5|10|20|25] , $m_1 m_2 m_3 m_4$

טבלת m שלנו תהיה: 4x4:

	1	2	3	4
1	0	750	2,500	5,375
2		0	1,000	3,500
3			0	5,000
4				0

$M[1,2] = m_1 \cdot m_2 = 15 \cdot 5 \cdot 10 = 750$

$M[2,3] = m_2 \cdot m_3 = 5 \cdot 10 \cdot 20 = 1{,}000$

$M[3,4] = m_3 \cdot m_4 = 10 \cdot 20 \cdot 25 = 5{,}000$

$M[1,3] = \text{I} \quad m_1(m_2 \cdot m_3) = 0 + 1{,}000 + 1{,}500 = 2{,}500$

$\text{II} \quad (m_1 \cdot m_2) m_3 = 0 + 750 + 3{,}000 = 3{,}750$

$M[2,4] = \text{I} \quad m_2(m_3 m_4) = 0 + 5{,}000 + 1{,}250 = 6{,}250$

$\text{II} \quad (m_2 m_3) \cdot m_4 = 1{,}000 + 0 + 2{,}500 = 3{,}500$

$M[1,4] = \text{I} \quad (m_1 \cdot (m_2 m_3 m_4)) = 3{,}500 + 1{,}875 = 5{,}375$

$\text{II} \quad ((m_1 \cdot m_2)(m_3 \cdot m_4)) = 5{,}750 + 3{,}750 = 9{,}500$

$\text{III} \quad ((m_1 m_2 m_3) m_4) = 2{,}500 + 7{,}500 = 10{,}000$

Scanned by CamScanner

8.12.16

עכשיו אחרי שמצאנו את הסכום של הפעולות המינימלי,

איך נדע איזה סדר של מכפלה ייתן לנו אותו?

→ בכל פעם נשמור את ה-k שמצאנו, כלומר איפה שמנו סוגריים.

	1	2	3	4
1	0	1	1	1
2		0	2	3
3			0	3
4				0

וכך הגענו לפתרון - $m_1((m_2m_3)m_4)$

הסבר קצר:

בכל פעם שמרנו קודם את ה-k, כלומר את האינדקס של m שאחריו הכנסנו סוגריים.

לאחר מכן בטבלה יש לנו את המיקומים הנ"ל.

ניגש לתא $M[1,4]$ ← יש בו 1 לכן אחרי m_1 יש סוגריים כך:

$(m_1)(m_2m_3m_4)$

← פיצלנו את המערך לשני חלקים $M[1,1], M[2,4]$ ניגש לתאים האלה בטבלה.

ניגש לתא $M[1,1]$ ← יש בו 0 ולכן m_1 מוכפל לבד וקיבלנו $m_1(m_2m_3m_4)$

ניגש לתא $M[2,4]$ ← יש בו 3 ולכן יש סוגריים אחרי m_3 וקיבלנו

$m_1((m_2m_3)m_4)$

← פיצלנו את המערך לשני חלקים שוב $M[2,3]$ ו- $M[4,4]$

ניגש לתא $M[4,4]$ ← יש בו 0 ולכן m_4 מוכפל עם עצמו.

ניגש לתא $M[2,3]$ ← יש בו 2 ולכן יש סוגר אחרי m_2, אך בגלל

שיש בו רק 2 תאים ניתן להתעלם מזה כי $m_2 \cdot m_3 = (m_2)(m_3)$

וכך קיבלנו $m_1((m_2m_3)m_4)$

Scanned by CamScanner