מבני נתונים ומבוא לאלגוריתמים
מפגש הנחיה מס' 3

מדעי המחשב, קורס מס' 20407

סמסטר 2016ב

מנחה: ג'ון מרברג

מה ראינו במפגש הקודם?

- תרגילים לפרק 2
- פרק 3 בספר – גידול של פונקציות
  - המוטיבציה לשימוש בסימונים אסימפטוטיים
  - הגדרת הסימונים
  - חסמים שימושיים ותכונות שימושיות
  - חישוב סיבוכיות הזמן של אלגוריתם
  - תרגילים לפרק 3

מפגש שלישי

- נושאי השיעור
  - סימונים אסימפטוטים – תרגילים נוספים
  - פרק 4 בספר – נוסחאות נסיגה
    - מבוא
    - שיטת ההצבה
    - שיטת האיטרציה
    - שיטת האב
    - תרגילים

מבוסס על מצגת של ברוך חייקין ואיציק בייז

נוסחאות נסיגה - מבוא

- הנוסחה מבטאת את עלות האלגוריתם (בדרך כלל עבור מס' הפעולות המבוצעות)
- מייצגת עלות של אלג' רקורסיבי ולפיכך מכילה מופע של עצמה
- דוגמאות של נוסחת נסיגה $T(n)$ עבור אלגוריתם רקורסיבי עם קלט בגודל n
  - $T(n) = T(n-1) + O(n)$
  - $T(n) = 2T(n/2) + O(n\log n)$
  - $T(n) = T(n/3) + T(2n/3) + O(\log n)$
  - $T(n) = T(n/2) + O(1)$
- זמן זה מייצג את המקרה הכללי. בנוסף יש לספק עלות למקרי קצה (בד"כ עלות קבועה על קלט קטן, נניח $n=1$)

נוסחאות נסיגה - מבוא

- מבנה טיפוסי של נוסחת נסיגה $T(n)$ עבור אלגוריתם רקורסיבי עם קלט בגודל n
  - אם $n<=n_0$ *(n מספיק קטן)*, אז $T(n) = \Theta(1)$ (זהו תנאי השפה)
  - אחרת, $T(n) = \sum_{i=1}^{k} T(n_i) + O(g(n))$

  כאשר
    - $k>=1$ *הוא מספר התת-בעיות*
    - $g(n)$ הוא חסם עליון על זמן ההפרדה והצירוף
    - $n_i < n$ הוא גודל הקלט עבור תת-בעיה ה-i

דוגמאות לנוסחאות נסיגה

אלגוריתם רקורסיבי	תנאי השפה	נוסחת נסיגה	פתרון
חיפוש בינארי	$T(1) = \Theta(1)$	$T(n) = T(n/2) + \Theta(1)$	$T(n) = \Theta(\log n)$
מיון-מיזוג	$T(1) = \Theta(1)$	$T(n) = 2T(n/2) + \Theta(n)$	$T(n) = \Theta(n \log n)$
מגדלי האנוי	$T(1) = \Theta(1)$	$T(n) = 2T(n-1) + \Theta(1)$	$T(n) = \Theta(2^n)$

שיטת ההצבה

- השיטה
  - מנחשים פתרון מהסוג $T(n) = O(f(n))$,
    כלומר $T(n) \le cf(n)$ לכל $n \ge n_0$
  - מוכיחים את נכונות הפתרון באינדוקציה על n:
    - מציבים את הניחוש בנוסחת הנסיגה
    - מוצאים קבועים c ו- n_0 עבורם תנאי השפה מתקיימים

שיטת ההצבה – דוגמה

- חסם עליון על חיפוש בינארי

נוסחת הנסיגה: $T(n) = T(n/2) + 1$, תנאי השפה: $T(1) = 1$

- ננחש את הפתרון $T(n) = O(\log n)$, כלומר $T(n) \le c\log_2 n$ לכל $n \ge n_0$
 וננסה למצוא קבועים c, n_0 עבורם האי-שוויון מתקיים
- צעד האינדוקציה

$$\begin{aligned} T(n) &= T(n/2) + 1 \\ &\le c\log_2(n/2) + 1 && \blacktriangleright \text{ by the induction hypothesis} \\ &= c\log_2 n - c\log_2 2 + 1 = c\log n - c + 1 \\ &\le c\log_2 n && \blacktriangleright \text{ inequality holds for } c \ge 1 \end{aligned}$$

- בסיס האינדוקציה
אי אפשר להוכיח את בסיס האינדוקציה עבור $n = 1$, לכן נתחיל מ- $n = 2$:

$$\mathrm{T}(2) = T(1) + 1 = 1 + 1 = 2 \le c\log_2 2 = c$$

- הטענה אכן נכונה עבור הקבועים $c = 2, n_0 = 2$

שיטת האיטרציה

- השיטה
  - מתחילים בנוסחה $T(n) = \sum_{i=1}^{k} T(n_i) + O(g(n))$
  - בשלב הבא, במקום כל $T(n_i)$ מפתחים את נוסחת הנסיגה שוב, וכך הלאה...
  - כשמגיעים לתנאי השפה, מציבים את הביטוי המתאים במקום $T(n_0)$
- אם וכאשר מסתמן דפוס מסוים המאפשר ניחוש פתרון, אפשר גם לעבור לשיטת ההצבה

שיטת האיטרציה – דוגמה 1

- חיפוש בינרי

נוסחת הנסיגה: $T(n) = T(n/2) + 1$, מקרה קצה: $T(1) = 1$

- נפתח (לשם הפשטות נניח ש- $n = 2^m$, כלומר $m = \log_2 n$):

$$\begin{aligned}
T(n) &= T(n/2) + 1 \\
&= (T(n/4) + 1) + 1 = T(n/4) + 2 \\
&= (T(n/8) + 1) + 2 = T(n/8) + 3 \\
&= \ldots = T(n/2^i) + i = \ldots \\
&= T(n/2^m) + m \\
&= T(1) + m \\
&= 1 + \log_2 n
\end{aligned}$$

- מתקבל הפתרון $T(n) = \Theta(\log n)$

האוניברסיטה הפתוחה

שיטת האיטרציה – דוגמה 2

- מיון-מיזוג

נוסחת הנסיגה: $T(n) = 2T(n/2) + cn$, מקרה קצה: $T(1) = 1$

- נפתח (לשם הפשטות נניח ש- $n = 2^m$, כלומר $m = \log_2 n$):

$$
\begin{aligned}
T(n) &= 2T(n/2) + cn \\
&= 2(2T(n/4) + cn/2) + cn = 4T(n/4) + 2cn \\
&= 4(2T(n/8) + cn/4) + 2cn = 8T(n/8) + 3cn \\
&= \ldots = 2^i T(n/2^i) + icn = \ldots \\
&= 2^m T(n/2^m) + mcn \\
&= 2^m T(1) + mcn \\
&= 2^m + mcn \\
&= \mathrm{n} + \log_2 n \cdot cn
\end{aligned}
$$

- מתקבל הפתרון $T(n) = \Theta(n\log n)$

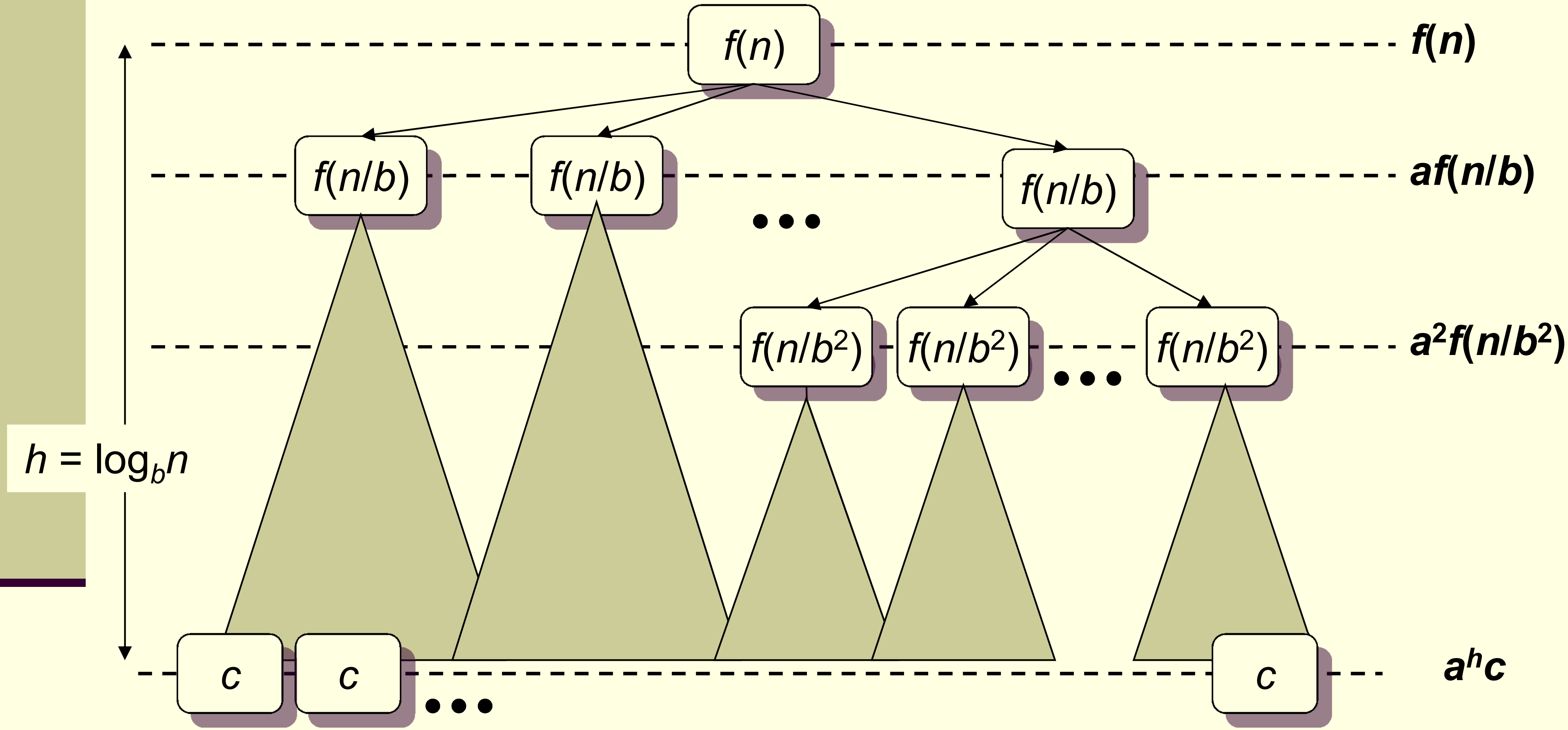
שיטת האב (Master Method)

האוניברסיטה הפתוחה

- הרעיון: טיפול אחיד בקבוצה גדולה של נוסחאות נסיגה
  - צורת הנוסחה: $T(n) = aT(n/b) + f(n)$ כאשר $f(n)$ פונקציה חיובית
  - הבעיה מתחלקת ל-a תת-בעיות בגודל n/b כל אחת,
    כאשר הפרמטרים $a \geq 1$, $b > 1$ הם קבועים
    - למשל: חיפוש בינארי ומיון-מיזוג, אבל לא מגדלי האנוי
- ניתן לראות את הריצה כעץ קריאות – בכל צומת פנימי מתבצעים הפרדה וצירוף בזמן $f()$, ובכל עלה מטופל מקרה בסיס בזמן קבוע
  - גובה העץ הוא $\log_b n$, ברמה ה- j בעץ יש a^j צמתים, וגודל הבעיה ברמה ה-j הוא n/b^j
  - לכן סה"כ עבודת ההפרדה והצירוף המתבצעת בעץ היא
    $\Sigma a^j f(n/b^j)$, כאשר $0 \leq j \leq (\log_b n) - 1$
  - מספר העלים בעץ הוא $a^{\log_b n} = n^{\log_b a}$ ובכל אחד מהם מתבצעת עבודה בגודל קבוע c
- לכן נוסחת הנסיגה מתכנסת ל:

$$\boxed{T(n) = cn^{\log_b a} + \sum_{j=0}^{(\log_b n)-1} a^j f\left(\frac{n}{b^j}\right)}$$

עץ הקריאות



$$T(n) = cn^{\log_b a} + \sum_{j=0}^{(\log_b n)-1} a^j f\left(\frac{n}{b^j}\right)$$

שיטת האב (המשך)

- כאמור, שיטת האב מטפלת בנוסחת הנסיגה
  $$T(n) = aT(n/b) + f(n)$$
- הגדרה לא פורמאלית של שיטת האב:
  - משווים בין שני גורמים:
    - מספר העלים בעץ הקריאות: $n^{\log_b a}$
    - *עלות הפירוק והצירוף בשורש:* $f(n)$
  - אם $n^{\log_b a}$ "גדול יותר"* אזי $T(n) = \Theta(n^{\log_b a})$
  - אם $f(n)$ "גדול יותר"* אזי $T(n) = \Theta(f(n))$
  - אם שני הערכים מאותו סדר גודל, אזי $T(n) = \Theta(f(n)\log n)$

* "גדול יותר" מתייחס להגדרה פורמאלית שתובא בהמשך

שיטת האב (המשך)

האוניברסיטה הפתוחה

- אם נצמצם את תחום הדיון לפונקציות f מסוג פולינום או פולילוגריתם בלבד (פונק' נפוצות בקורס שלנו) אזי שיטת האב אומרת כדלקמן:

  - צורת הנוסחה : $T(n) = aT(n/b) + f(n)$ כאשר $a\geq 1$ ו $b>1$ הם קבועים כלשהם (למשל a=4, b=2)
    - $n^{\log_b a}$ הוא פולינום ב- n (למשל n^2)
    - $f(n)$ הוא כאמור פולינום או פולילוגריתם או שילוב (למשל $n^{1.5}\log^2 n$)

  - בכדי לקבוע מי "גדול יותר", $n^{\log_b a}$ או $f(n)$, נשווה בין החזקות <u>הפולינומיות</u> של שני הגורמים (כי כל פולילוגריתם קטן אסימפטוטית מכל פולינום)
    - הביטוי עם החזקה הפולינומית הגדולה יותר הוא ה"גדול יותר" ולכן הוא קובע את התוצאה
      - למשל בין n^2 ל- $n^{1.5}\log^2 n$ ה"גדול יותר" הוא n^2 ולכן $T(n) = \Theta(n^2)$
    - אם החזקות הפולינומיות שוות, שיטת האב מתאימה רק כאשר שני הגורמים הם מאותו סדר גודל, כלומר אם $f(n) = \Theta(n^{\log_b a})$, ואז $T(n) = \Theta(f(n)\log n)$

  - בהמשך נציג הרחבה של מקרה השוויון גם עבור $f(n)$ פולילוגריתמי

שיטת האב – הגדרה פורמלית

- שיטת האב מתייחסת לנוסחת הנסיגה: $T(n) = aT(\frac{n}{b}) + f(n)$ כאשר $a \geq 1$, $b>1$
- השיטה מבחינה בין שלשה מקרים של עלות העבודה בעץ הקריאות:

1. **הגורם הדומיננטי הוא עלות העבודה המתבצעת בעלים (= מספר העלים)**

   אם קיים קבוע $\varepsilon > 0$ כך ש- $f(n) = O(\frac{n^{\log_b a}}{n^{\varepsilon}})$ אז $T(n) = \Theta(n^{\log_b a})$

2. **העבודה מתחלקת באופן שווה בין כל הרמות בעץ**

   אם $f(n) = \Theta(n^{\log_b a})$ אז $T(n) = \Theta(n^{\log_b a} \log n)$

3. **הגורם הדומיננטי הוא עלות העבודה בשורש**

   אם קיים קבוע $\varepsilon > 0$ כך ש- $f(n) = \Omega(n^{\log_b a} n^{\varepsilon})$ אז $T(n) = \Theta(f(n))$

   במקרה 3 חייב להתקיים גם תנאי הרגולריות על *f(n)* כדלקמן:

   קיימים קבועים $c<1$ ו- n_0 כך ש- $af(\frac{n}{b}) \leq cf(n)$ לכל $n \geq n_0$

(*) הערה לגבי היחס בין הגדלים במקרים 1, 3:

נשים לב שהגורם הדומיננטי חייב להיות גדול פולינומיאלית מהגורם השני, בפקטור של n^{ε}

האוניברסיטה הפתוחה

שיטת האב – דוגמה למקרה 1

- בעיה 4-1ה (עמ' 73 בספר)

נוסחת הנסיגה: $T(n) = 7T(n/2) + n^2$

$a = 7,\ b = 2,\ f(n) = n^2$

- אנו מבחינים כי: $n^{\log_b a} = n^{\log_2 7} > n^2 = f(n)$
- לכן אנו מנסים לחסום את $f(n)$ מלמעלה
- בדיקת התנאי: נבחר $\varepsilon > 0$ מתאים כך שיתקיים

$$f(n) = n^2 = O(n^{\log_b a}/n^{\varepsilon}))$$

נשתמש בהגדרות הסימון האסימפטוטי O:

$$n^2 \leq cn^{\log_b a}/n^{\varepsilon}$$

$$n^{\varepsilon} \leq cn^{\log_b a}/n^2 = cn^{\log_2 7}/n^2) = cn^{\log_2 (7/4)}$$

$$\blacktriangleright\ \varepsilon = \log_2(7/4) > 0 \qquad c=1$$

- הראינו שנוסחת הנסיגה שייכת למקרה 1
- מסקנה: $T(n) = \Theta(n^{\log_2 7}) \approx \Theta(n^{2.8})$

שיטת האב – דוגמה למקרה 2

- מיון-מיזוג

נוסחת הנסיגה: $T(n) = 2T(n/2) + n$

$a=2,\ b=2,\ f(n)=n$

- אנו מבחינים כי במקרה זה: $n^{\log_b a} = n^{\log_2 2} = n = f(n)$
- לכן: $f(n) = n = \Theta(n^{\log_b a})$
- הראינו שנוסחת הנסיגה שייכת למקרה 2
- מסקנה: $\mathrm{T}(n)=\Theta(n\log n)$

האוניברסיטה הפתוחה

שיטת האב – עוד דוגמה למקרה 2

- בעיה 1-4ג (עמ' 73 בספר)

נוסחת הנסיגה: $T(n) = 16T(n/4) + n^2$

$a = 16,\ b = 4,\ f(n) = n^2$

- אנו מבחינים כי במקרה זה: $n^{\log_b a} = n^{\log_4 16} = n^2 = f(n)$
- לכן $f(n) = n^2 = \Theta(n^{\log_b a})$
- הראינו שנוסחת הנסיגה שייכת למקרה 2
- מסקנה: $T(n) = \Theta(n^2 \log n)$

שיטת האב – דוגמה למקרה 3

- בעיה 1-4ד (עמ' 73 בספר)

  נוסחת הנסיגה: $T(n) = 7T(n/3) + n^2$

  $a = 7,\ b = 3,\ f(n) = n^2$

- אנו מבחינים כי: $n^{\log_b a} = n^{\log_3 7} < \mathrm{n}^{\log_3 9} = n^2 = f(n)$
- לכן אנו מנסים לחסום את $f(n)$ מלמטה
- בדיקת התנאי: נבחר $\varepsilon > 0$ כך שיתקיים

$$f(n) = n^2 = \Omega(n^{\log_b a} \cdot n^{\varepsilon})$$

נשתמש בהגדרת הסימון האסימפטוטי Ω:

$$n^2 \geq cn^{\log_b a}n^{\varepsilon}$$

$$cn^{\varepsilon} \leq n^2/n^{\log_b a} = n^2/n^{\log_3 7} = n^{(\log_3 9 - \log_3 7)} = n^{\log_3 (9/7)}$$

$$\blacktriangleright\ \varepsilon = \log_3(9/7) > 0 \qquad c = 1$$

- בדיקת תנאי הרגולריות: נבחר $c<1$ כך שיתקיים $af(n/b) \leq cf(n)$

$$af(n/b) = 7(n/3)^2 = (7/9)n^2$$

$$\blacktriangleright\ c = 7/9 < 1$$

- הראינו שנוסחת הנסיגה שייכת למקרה 3
- מסקנה: $T(n) = \Theta(n^2)$

האוניברסיטה הפתוחה

שיטת האב – מקרה 2 המורחב

- נרחיב את הנוסחה של מקרה 2 עבור מצבים בהם $f(n)$ חסום על ידי פונקציה שכוללת כופל פולילוגריתמי

  אם $f(n) = \Theta(n^{\log_b a} \cdot \log^k n)$ עבור $k \geq 0$

  אז פתרון נוסחת הנסיגה $T(n) = aT(n/b) + f(n)$

  הוא: $T(n) = \Theta(n^{\log_b a} \cdot \log^{k+1} n)$

- אינטואיטיבית, בהרחבה זו השוויון בין $f(n)$ לבין מספר העלים הוא עד כדי פקטור של פונקציה פולילוגריתמית
- מופיע כתרגיל 4.4-2 (עמ' 72 בספר)

דוגמה – כפל מטריצות

- דרוש אלגוריתם לכפל מטריצות $C = A \cdot B$
  - A, B, C מטריצות ריבועיות בגודל $n \times n$ ($n = 2^m$)
  - תזכורת: האיבר C_{ij} של מטריצת המכפלה $C = A \cdot B$ מחושב ע"י המכפלה הפנימית $C_{ij} = \Sigma_{k=1}^{n} A_{ik}B_{kj}$
  - אלגוריתם פשוט:

SimpleMatrixMultiply(A, B, C, n)
1. **for** $i \leftarrow 1$ **to** n
2.   **do for** $j \leftarrow 1$ **to** n
3.   **do** $C[i, j] \leftarrow 0$
4.   **for** $k \leftarrow 1$ **to** n
5.   **do** $C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]$
6. **return** C

  - זמן הריצה: $\Theta(n^3)$

האוניברסיטה הפתוחה

כפל מטריצות (המשך)

- מה לגבי אלגוריתם רקורסיבי?
  - הרעיון: הפרד ומשול

$$A \times B = \begin{bmatrix} A1 & A2 \\ A3 & A4 \end{bmatrix} \times \begin{bmatrix} B1 & B2 \\ B3 & B4 \end{bmatrix} = \begin{bmatrix} C1 & C2 \\ C3 & C4 \end{bmatrix} = C$$

  - נחשב רקורסיבית את 8 מכפלות הרבעים, ונחבר

$$C1 = A1 \cdot B1 + A2 \cdot B3 \qquad C3 = A3 \cdot B1 + A4 \cdot B3$$

$$C2 = A1 \cdot B2 + A2 \cdot B4 \qquad C4 = A3 \cdot B2 + A4 \cdot B4$$

  - נוסחת הנסיגה המתקבלת: $T(n) = 8T(n/2) + 4(n/2)^2$
  - פתרון (מקרה 1 של שיטת האב): $T(n) = \Theta(n^3)$
  - אותו זמן ריצה אסימפטוטי, אך צריכת הזיכרון היא $\Theta(\log n)$
  - לא הושג כל שיפור...

כפל מטריצות (המשך)

- האלגוריתם של שטראסן (Strassen, 1969)
  - מספיקות רק 7 מכפלות רבעים!

$$P1 = A1 \cdot (B2 - B4)$$
$$P2 = (A1 + A2) \cdot B4$$
$$P3 = (A3 + A4) \cdot B1$$
$$P4 = A4 \cdot (B3 - B1)$$
$$P5 = (A1 + A4) \cdot (B1 + B4)$$
$$P6 = (A2 - A4) \cdot (B3 + B4)$$
$$P7 = (A1 - A3) \cdot (B1 + B2)$$

$$\mathrm{C}1 = P4 + P5 + P6 - P2$$
$$\mathrm{C}2 = P1 + P2$$
$$\mathrm{C}3 = P3 + P4$$
$$\mathrm{C}4 = P1 + P5 - P3 - P7$$

  - נוסחת הנסיגה המתקבלת: $T(n) = 7T(n/2) + 18(n/2)^2$
  - פתרון (מקרה 1 של שיטת האב): $T(n) = \Theta(n^{\log_2 7}) = \Theta(n^{2.81})$
  - באופן מעשי, עבור $n < 45$ האלגוריתם הנאיבי עדיף.