הרצאה 10: משפט האב (Master Theorem) אלגוריתם Strassen למכפלת מטריצות ריבועיות

להרצאה זו, שני חלקים:

1. נציג ונוכיח את משפט האב המאפשר חישוב סיבוכיות של אלגוריתמים רקורסיביים רבים.
2. נציג את אלגוריתם Strassen להכפלת מטריצות ריבועיות ונחשב את הסיבוכיות שלו, בעזרת משפט האב.

משפט האב

קיימים אלגוריתמים רקורסיביים רבים שאנו מתקשים בחישוב הסיבוכיות שלהם. משפט האב מספק לנו מכשיר רב עוצמה לטיפול ברוב האלגוריתמים הרקורסיביים. אנו נלמד גרסה מסויימת של משפט האב. גרסה אחרת נמצאת בספר הקורס.

התנאים להפעלת משפט האב

משפט האב מאפשר לנו לדון באלגוריתמים רקורסיביים המקיימים את התנאים הבאים:
לכל בעיה בגודל n, אשר אינה מקרה קצה, האלגוריתם מבצע a קריאות רקורסיביות לפתרון a תת בעיות בגודל n/b.

שימו לב

1. a ו b הם קבועים התלויים בבעיה אך לא ב n.
2. a תת הבעיות אינן בהכרח חלוקה של הקלט ל a חלקים זרים.

דוגמא

מיון ע״י מיזוג:

1. **חלק** את מערך הקלט ל 2 מערכים בגודל $n/2$. ($a = b = 2$).
2. **מיין** כל תת-מערך באופן על ידי קריאה רקורסיבית למיון-מיזוג.
3. **מזג** את שתי המחציות הממוינות.

לעומת דוגמא זו, **מיון מהיר** אינו עונה לדרישות אלה משום שבמיון מהיר מחלקים את מערך הקלט לשני קבצים שהיחס בין גדליהם אינו קבוע.

הצגת סיבוכיות מיון-מיזוג על ידי מערכת משוואות רקורסיביות

נסמן ב $T(n)$ את הזמן הדרוש למיון-מיזוג מערך בן n איברים.
הזמן הנדרש לכל קריאה רקורסיבית הוא כמובן $T\left(\frac{n}{2}\right)$ ומאחר שמיזוג של שני מערכים באורך $n/2$ כל אחד אורך זמן לינארי ומיון קובץ באורך 1 אורך זמן קבוע כלשהו נקבל כי :

$$T(n) = 2T\left(\frac{n}{2}\right) + dn$$

$$T(1) = c$$

עבור c ו d קבועים כלשהם.

הכללת מערכת המשוואות

נתבונן באלגוריתם רקורסיבי A לפתרון בעיה כלשהי P. נניח כי אם מריצים את A על קלט בגודל n אזי :

1. נדרשות a קריאות רקורסיביות עם קלט בגודל n/b.
2. מספר הפעולות בשגרה הקוראת (top level), למעט הקריאות הרקורסיביות, הוא $df(n)$ כאשר $f(n)$ היא פונקציה כלשהי התלויה בגודל הקלט.
3. פתרון מקרה הקצה של הרקורסיה, עבור קלט בגודל 1, דורש c פעולות.
4. הקבועים a, b, c, d והפונקציה f תלויים בבאלגוריתם A.

הכללת מערכת המשוואות (המשך)

נסמן ב$T(n)$ את הזמן הנדרש להרצת האלגוריתם A על קלט בגודל n.
אזי מתקיים :

$$T(n) = aT\left(\frac{n}{b}\right) + df(n)$$

$$T(1) = c$$

להלן נציג את משפט האב (Master Theorem) הפותר את מערכת המשוואות הזו באופן כללי.

משפט הMaster

תהי $T(n)$ פונקציה המוגדרת על ידי :

$$T(n) = aT\left(\frac{n}{b}\right) + df(n)$$

$$T(1) = c$$

אזי אם f היא פונקציה כפלית (תוגדר להלן) התנהגות הפונקציה $T(n)$ תלויה ביחס בין a לבין $f(b)$ כמתואר להלן :

1. אם $a > f(b)$,
 $T(n) = \Theta\left(n^{\log_b a}\right)$.
2. אם $f(b) = a$,
 $T(n) = \Theta\left(n^{\log_b a} \log n\right)$.
3. אם $a < f(b)$, $T(n) = \Theta(f(n))$.

הערות למשפט הMaster

1. **במקרה 1 -** רוב צעדי A מתבצעים על ידי הקריאות הרקורסיביות.
   במקרה 2 - רוב צעדי A מתבצעים מחוץ לקריאות הרקורסיביות.
   במקרה 3 - שני הגורמים הקודמים תורמים לסיבוכיות.
2. לערכי c ו-d אין שום השפעה על $T(n)$.
3. בספר מוצגת גרסה קצת שונה של המשפט, אשר אינה מניחה כי הפונקציה f היא כפלית. במקרה זה, הטיפול המתמטי בבעיה שונה, אולם ההשלכות המעשיות דומות.

הוכחת משפט הMaster

הוכחת משפט ה-Master מתקבלת על ידי פתרון המשוואות הרקורסיביות.

אנו נטפל במשוואה פשוטה יותר:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

נציב בנוסחה את $T\left(\frac{n}{b}\right)$ ונקבל:

$$T(n) = a\left[aT\left(\frac{n}{b^2}\right) + f\left(\frac{n}{b}\right)\right] + f(n)$$

נפשט ונקבל:

$$T(n) = a^2T\left(\frac{n}{b^2}\right) + af\left(\frac{n}{b}\right) + f(n)$$

פתרון המשוואות הרקורסיביות

נמשיך להציב את $T\left(\frac{n}{b^2}\right)$ ונקבל:

$$T(n) = a^2\left[aT\left(\frac{n}{b^3}\right) + f\left(\frac{n}{b^2}\right)\right] + af\left(\frac{n}{b}\right) + f(n))$$

נפשט ונקבל:

$$T(n) = a^3T\left(\frac{n}{b^3}\right) + a^2f\left(\frac{n}{b^2}\right) + af\left(\frac{n}{b}\right) + f(n))$$

$$= a^3T\left(\frac{n}{b^3}\right) + \sum_{i=0}^{2} a^i f\left(\frac{n}{b^i}\right)$$

פתרון המשוואות הרקורסיביות

כדי שהרקורסיה תיגמר יש להגיע עד למקרה הקצה כלומר $T(1)$. כדי להשיג זאת, על הקריאה הרקורסיבית להתבצע לעומק i, המקיים $n = b^i$ כלומר $i = \log_b n$.

המשך ההצבה $\log_b n$ פעמים מניב:

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

אם נציב $T(1) = c$ ונזכור כי $n = b^{\log_b n}$ נקבל:

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

פתרון המשוואות הרקורסיביות

כעת נשים לב כי:

$$a^{\log_b n} = \left(b^{\log_b a}\right)^{\log_b n} =$$
$$= \left(b^{\log_b n}\right)^{\log_b a} = n^{\log_b a}$$

אם נציב שני ביטויים אלה ונקבל כי

$$T(n) = \underbrace{cn^{\log_b a}}_{*} + \underbrace{\sum_{i=0}^{(\log_b n)-1} a^i f\left(b^{\log_b n-i}\right)}_{**}$$

המחובר הראשון, $(*)$, מתנהג כמו $\Theta\left(n^{\log_b a}\right)$ וזהו פתרון הבעיה כאשר $f(n)=0$, אולם אין זה ברור כלל וכלל איך אפשר לטפל במחובר השני $(**)$.

פונקציות כפליות

כדי לטפל במחובר השני נניח כי f היא **פונקציה כפלית**.

פונקציה $f(x)$ היא **פונקציה כפלית** אם לכל x ו-y f מקיימת:

$$f(xy) = f(x)f(y).$$

דוגמא: הפונקציה הפולינומית $f(x) = x^d$ (d קבוע) היא כפלית שכן

$$f(xy) = (xy)^d = x^d y^d = f(x)f(y)$$

שימו לב: באופן מעשי, רוב הפונקציות המתקבלות בשימוש במשפט האב הן כפליות.

תכונות פונקציות כפליות

תהי f פונקציה כפלית כלשהי. אזי $f(1)$ מקיים:

$$f(1) = f(1\cdot 1) = f(1)\cdot f(1) = 1$$

נחשב כעת את ערך $f\left(\frac{a}{b}\right)$ עבור מנה כלשהי $\frac{a}{b}$:

$$f(1) = 1 = f\left(b\cdot\frac{1}{b}\right) = f(b)\cdot f\left(\frac{1}{b}\right) \Rightarrow$$

נחלק את שני האגפים ב $f(b)$ ונקבל:

$$f\left(\frac{1}{b}\right) = \frac{f(1)}{f(b)}$$

$$f\left(\frac{a}{b}\right) = f(a)\cdot f\left(\frac{1}{b}\right) = \frac{f(a)}{f(b)}$$

פתרון המשוואות הרקורסיביות

עבור פונקציה כפלית f

נניח כעת כי הפונקציה f היא **כפלית**.

במקרה כזה נפתח את $(**)$ ונקבל:

$$(**) = \sum_{i=0}^{(\log_b n)-1} a^i f\left(\frac{b^{\log_b n}}{b^i}\right) =$$

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

$$= f(b^{\log_b n}) \sum_{i=0}^{(\log_b n)-1} \frac{a^i}{f(b)^i} =$$

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

וזהו טור גיאומטרי עם מנה $q = \frac{a}{f(b)}$

פתרון המשוואות הרקורסיביות

עבור פונקציה כפלית *f* (המשך)

אם נניח כי $a \neq f(b)$ ונשתמש בנוסחת הטור הגיאומטרי עבור $q = \frac{a}{f(b)}$ נקבל:

$$(**) = \left(f(b)\right)^{\log_b n} \frac{\left(\frac{a}{f(b)}\right)^{\log_b n} - 1}{\frac{a}{f(b)} - 1} =$$

ולאחר פישוט נוסף נגיע ל

$$(**) = \frac{a^{\log_b n} - \left(f(b)\right)^{\log_b n}}{\frac{a}{f(b)} - 1}$$

פתרון המשוואות הרקורסיביות

עבור פונקציה כפלית *f* (המשך):

נתבונן ב $(**)$

$$(**) = \frac{a^{\log_b n} - \left(f(b)\right)^{\log_b n}}{\frac{a}{f(b)} - 1}$$

בביטוי זה, המכנה הוא קבוע ולכן ערכו של הביטוי תלוי ביחס בין a לבין $f(b)$ כמתואר בניתוח הבא:

מקרה 1

$a > f(b)$ - במקרה זה, ערך $(**)$ המונה מתנהג כמו $\Theta\left(a^{\log_b n}\right)$ והמכנה הוא קבוע ולכן במקרה זה נקבל:

$$(**) = \Theta\left(a^{\log_b n}\right) = \Theta\left(n^{\log_b a}\right)$$

מאחר שערך $(*)$ גם הוא $\Theta\left(n^{\log_b a}\right)$ נקבל כי במקרה זה מתקיים:

$T(n) = \Theta\left(n^{\log_b a}\right)$.

מקרה 2

$a < f(b)$ - במקרה זה, ערך המונה הוא $\Theta\left(f\left(b^{\log_b n}\right)\right) = \Theta(f(n))$ וסימנו שלילי. מאחר שגם סימן המכנה הוא שלילי נקבל:

$$(**) = \Theta\left(f\left(b^{\log_b n}\right)\right) = \Theta(f(n))$$

מאחר שערך $(*)$ הוא $\Theta\left(a^{\log_b n}\right)$ ו$a < f(b)$, במקרה זה, ערך $(*)$ קטן מערך $(**)$ ומתקיים כי

$T(n) = \Theta(f(n))$.

מקרה 3

$$a = f(b)$$

במקרה זה, נוסחת הטור הגיאומטרי אינה שמישה, שכן ערך המכנה בנוסחה הוא 0. אולם, במקרה זה מתקיים:

$$(**) = \sum_{i=0}^{(\log_b n)-1} a^i \frac{f\left(b^{\log_b n}\right)}{f\left(b^i\right)} =$$

$$= \sum_{i=0}^{\log_b n-1} f(n) = \log_b n \cdot f(n)$$

ומאחר שמתקיים:

$$f(n) = f\left(b^{\log_b n}\right) = (f(b))^{\log_b n} =$$

$$= a^{\log_b n} = n^{\log_b a}$$

מקבלים כי במקרה זה:

$$T(n) = \Theta\left(n^{\log_b a}\log_b n\right)$$

פתרון המשוואות הרקורסיביות עבור פונקציה כפלית f (סיכום):

לכל פונקציה כפלית f, ערך הפונקציה T המוגדרת על ידי נוסחת הנסיגה

$$T(n) = aT\left(\frac{n}{b}\right) + df(n)$$

$$T(1) = c$$

הוא

1. עבור $a > f(b)$ $\Theta\left(n^{\log_b a}\right)$.
2. עבור $a < f(b)$ $\Theta(f(n))$.
3. עבור $a = f(b)$ $\Theta\left(n^{\log_b a}\log n\right)$.

תרגיל: אפשר (וצריך) לבדוק כי הוספת הקבוע d אינה משנה את הפיתוח.

דרכים לשיפור אלגוריתם הפרד ופתור

יהי A אלגוריתם הפרד ופתור שסיבוכיותו מקיימת את מערכת המשוואות המופיעה במשפט האב. מהן הדרכים האפשריות לשיפור הסיבוכיות של A?

1. אם ש$a > f(b)$, נסה להקטין את $\log_b a$ על ידי הקטנת a או הגדלת b.
2. אם $f(b) \ge a$ נסה להקטין את סיבוכיות הפעולות שאינן כלולות בקריאות הרקורסיביות.
3. הקבועים c וd אינם מתבטאים בפתרון ולכן הקטנתם לא תשנה את הסיבוכיות.

דוגמא - סיבוכיות מיון-מיזוג

כזכור, מיון מיזוג מקיים:

$$T(n) = 2T\left(\frac{n}{2}\right) + dn$$

$$T(1) = c$$

מאחר ש $a = b = 2$ ו $f(n) = n$, אנו מקבלים כי

$$a = 2 = f(2) = f(b)$$

ומכאן, סיבוכיות מיון-מיזוג היא

$$T(n) = \Theta\left(n^{\log_2 2}\log_2 n\right) = \Theta(n\log n)$$

כעת, נציג את אלגוריתם Strassen למכפלת מטריצות מרובעות. סיבוכיות האלגוריתם תקבע תוך שימוש במשפט האב.

אלגוריתם Strassen

מכפלת מטריצות ריבועיות:

קלט: שתי מטריצות **ריבועיות** A ו B מסדר $n \times n$.

פלט: מכפלת הטריצות $C = A \times B$.

שימו לב: גודל הקלט הוא $O(n^2)$.

נפתח את ההרצאה בחזרה על האלגוריתם התקני ולאחר מכן נפתח את אלגוריתם Strassen. נסיים את ההרצאה בחישוב הסיבוכיות של האלגוריתם, שהיא $\Theta(n^{2.81\ldots})$, תוך שימוש במשפט האב.

מכפלת מטריצות

מכפלת המטריצות A ו B היא מטריצה C, המקיימת

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

ומסמנים $C = A \times B$.

מהי סיבוכיות בעיית חישוב מכפלת שתי מטריצות?
ברור כי $\Omega(n^2)$ היא חסם תחתון לסיבוכיות הבעיה (למה?).

אלגוריתם ישיר

האלגוריתם הישיר לחישוב C הוא:

```
for i = 1 to n do
  for j = 1 to n do
    c_ij ← 0
    for k = 1 to n do
      c_ij ← c_ij + a_ik · b_kj
    end
  end
end
```

זמן הריצה: $\Theta(n^3)$.

בהמשך ההרצאה נפתח אלגוריתם יעיל יותר המשתמש בשיטת הפרד ופתור ומשיג זמן ריצה $\Theta(n^{2.81\ldots})$.

תצוגה מטריצית של כפל מטריצות

נפתח כעת אלגוריתם אחר, רקורסיבי, למכפלת מטריצות. נפתח על ידי הצגת מטריצה בעזרת 4 תת-מטריצות:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

תחת תצוגה זו, המכפלה $C = A \times B$ מתקבלת כמכפלת שתי מטריצות מסדר 2×2 שאבריהן הן תת מטריצות של מטריצות הקלט.

תצוגה מטריצית של כפל מטריצות

$$C=\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}\times\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}=$$

$$\begin{pmatrix} A_{11}B_{11}+A_{12}B_{21} & A_{11}B_{12}+A_{12}B_{22} \\ A_{21}B_{11}+A_{22}B_{21} & A_{21}B_{12}+A_{22}B_{22} \end{pmatrix}=$$

$$=\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

הערות

1. כל אחת מן המטריצות A_{ij}, B_{ij} ו-C_{ij}, $i,j=1,2$ היא מסדר $\frac{n}{2}\times\frac{n}{2}$ ובכל מטריצה כזו יש $\frac{n^2}{4}$ איברים.
2. במקרה זה, פעולות החיבור והכפל הן **פעולות של מטריצות**. פעולות הכפל מחושבות על ידי קריאות רקורסיביות לאלגוריתם הכפל. הסימון × הושמט כדי לחסוך במקום. פעולות חיבור המטריצות משתמשות באלגוריתם הבא:

סכום מטריצות

תהיינה A ו-B מטריצות ריבועיות מסדר n (בגודל $n\times n$).

סכום המטריצות $D=A+B$ הוא מטריצה D המקיימת

$$d_{ij}=a_{ij}+b_{ij}\quad i,j=1,2,...,n$$

נתבונן בבעיית חישוב סכום מטריצות:

קלט: שתי מטריצות מסדר n.

פלט: סכום המטריצות.

טענה: סיבוכיות בעיית חישוב סכום מטריצות היא $\Theta\left(n^2\right)$.

הוכחה: כל אלגוריתם חייב לייצר את מטריצת הסכום שגודלה הוא n^2.

אלגוריתם הפרד ופתור (רקורסיבי) לחישוב מכפלת מטריצות בגודל $n\times n$

1. "רפד" את שתי המטריצות באפסים עד שמימדן יגיע לחזקה של 2 הגדולה מ-n והקרובה אליו ביותר.
2. **אם** $n=1$ הכפל A ב-B (כפל שלמים)
3. **אחרת**
   - 3.1 חלק כל מטריצה לארבע.
   - 3.2 חשב את שמונה המכפלות הנדרשות (בעזרת קריאות רקורסיביות).
   - 3.3 בצע ארבע פעולות חיבור.
   - 3.4 מזג התוצאות למטריצה הנדרשת.

אנליזה של זמן החישוב

נסמן ב $T(n)$ את מספר הפעולות האריתמטיות, חיבור וכפל, הנדרשות להכפלת שתי מטריצות שגודלן $n \times n$.
מן האלגוריתם הרקורסיבי שהצגנו נובעות **משוואות הרקורסיה** הבאות:

$$T(n) = 8T\left(\frac{n}{2}\right) + 4\left(\frac{n}{2}\right)^2 = 8T\left(\frac{n}{2}\right) + n^2$$

ידוע גם כי $T(1) = 1$ כי מטריצה מסדר 1×1 היא סקלר ומכפלת שתי מטריצות כאלה דורשת פעולת כפל יחידה.

הצבה במשוואות הרקורסיה

נשתמש במשוואות הרקורסיה כדי לחשב:

$$T(2) = 8T(1) + 4 \cdot 1^2 = 12$$

$$T(4) = 8T(2) + 4^2 = 96 + 16 = 112$$

בדרך זו נוכל לחשב את מספר הצעדים הדרושים עבור זוג מטריצות מסדר n, לכל $n > 0$ אך לא נוכל לחשב את מספר הצעדים הזה **כפונקציה של** n.
כדי לחשב את סיבוכיות האלגוריתם, נשתמש במשפט האב ונקבל:

$$c=1,\ d=1,\ a=8,\ b=2,\ f(x) = x^2$$

במקרה זה, $f(2) = 4 < a$ ומקבלים

$$T(n) = \Theta\left(n^{\log_2 8}\right) = \Theta(n^3)$$

דיון

ברור כי $2n^3 - n^2 = O(n^3)$, כלומר סיבוכיות האלגוריתם החדש שווה לסיבוכיות האלגוריתם הישיר. מה הרווחנו?

תשובה:
הרווחנו דרך חדשה להסתכלות על הבעיה. כעת אנו יכולים לחפש אלגוריתם רקורסיבי **יעיל יותר** כדי לצמצם את זמן החישוב הנדרש כדי לחשב את תת המטריצות C_{ij}, $.i, j = 1,2$

האלגוריתם של Strassen

מדען המחשבים השויצרי, Strassen, גילה שיטה חלופית להכפלת שתי מטריצות מגודל 2×2 .
בשיטת Strassen מתבצעות שבע פעולות כפל מטריצות במקום שמונה פעולות כפל מטריצות המתבצעות בשיטה המקובלת. בתמורה עולה מספר פעולות החיבור.
הפעלת שיטת Strassen מניבה אלגוריתם להכפלת מטריצות שהסיבוכיות שלו היא:

$$\Theta\left(n^{\log_2 7}\right) = \Theta\left(n^{2.81\ldots}\right).$$

האלגוריתם של Strassen (המשך)

האינטואיציה: המקור לאיבר n^3 הן המכפלות הרקורסיביות. סיבוכיות החיבור היא n^2 בלבד. כדי להקטין את הסיבוכיות נוריד את מספר המכפלות ונעלה את מספר הסכומים.

הרעיון: כמו באלגוריתם הרקורסיבי המקורי נחשב את ארבע המטריצות C_{11}, C_{12}, C_{21} ו C_{22}.
כדי לחשב מטריצות אלה נחשב תחילה שבע מטריצות ביניים:

שבע מטריצות הביניים

כדי לחשב את המטריצות C_{11}, C_{12}, C_{21} ו C_{22}, נחשב תחילה, כשלב בינים, את שבע המטריצות הבאות:

$$M_1=(A_{12}-A_{22})\times(B_{21}+B_{22})$$
$$M_2=(A_{11}+A_{22})\times(B_{11}+B_{22})$$
$$M_3=(A_{11}-A_{21})\times(B_{11}+B_{22})$$
$$M_4=(A_{11}+A_{12})\times B_{22}$$
$$M_5=A_{11}\times(B_{12}-B_{22})$$
$$M_6=A_{22}\times(B_{21}-B_{11})$$
$$M_7=(A_{21}+A_{22})\times B_{11}$$

חוק הפילוג עבור מטריצות

לפני שנמשיך בהצגת האלגוריתם יש לציין כי צריך (וגם קל) לוודא כי **חוק הפילוג (דיסטריבוטיביות)** של הכפל מעל החיבור מתקיים גם עבור כפל וחיבור מטריצות כלומר: לכל שלוש מטריצות מסדר $n\times n$ A, B ו C מתקיים:

$$A\times(B+C)=A\times B+A\times C$$
$$(B+C)\times A=B\times A+C\times A$$

מדוע יש צורך בשני חוקי פילוג?

תרגיל:

שכנעו את עצמכם כי חוק הפילוג עבור כפל וחיבור מטריצות אכן מתקיים.

האלגוריתם של Strassen (המשך):

לאחר חישוב שבע מטריצות הביניים נשתמש בהן כדי לחשב את ארבעת המטריצות הסופיות כדלהלן:

$$C_{11}=M_1+M_2-M_4+M_6$$
$$C_{12}=M_4+M_5$$
$$C_{21}=M_6+M_7$$
$$C_{22}=M_2-M_3+M_5-M_7$$

נבדוק את הנוסחה הראשונה:

$$\begin{array}{ll} A_{12}B_{21}-A_{22}B_{21}+A_{12}B_{22}-A_{22}B_{22} & (=M_1) \\ +A_{11}B_{11}+A_{11}B_{22}+A_{22}B_{11}+A_{22}B_{22} & (=M_2) \\ -A_{11}B_{22}-A_{12}B_{22} & (=M_4) \\ +A_{22}B_{21}-A_{22}B_{11} & (=M_6) \\ =C_{11} & \end{array}$$

תרגיל: בדוק נכונות שאר הנוסחאות.

סיבוכיות אלגוריתם Strassen:

סיבוכיות הזמן נתונה על ידי המשוואה הרקורסיבית הבאה:

$$T(n) = 7T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2$$

$$T(1) = c$$

נשתמש במשפט האב ונקבל:

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

פרמטרים אלה מקיימים:

$f(2) = 4 < a$

ולכן

$$T(n) = \Theta\left(n^{\log_2 7}\right) = O(n^{2,81...})$$

סיכום

בהרצאה זו למדנו את **משפט האב (Master Theorem)**, המאפשר לנו לקבוע סיבוכיות של מגוון רחב של אלגוריתמים רקורסיביים.

פתחנו את ההרצאה בהצגה פרמטרית של מערכת משוואות רקורסיבית כך שתשקף את סיבוכיות הזמן של אלגוריתמים רקורסיביים המקיימים דרישה מסוימת.

לאחר מכן הצגנו את משפט האב המאפשר חישוב הסיבוכיות של אלגוריתמים כאלה.

סיכום (המשך)

כדי להדגים את השיטה, טיפלנו בבעית כפל המטריצות הריבועיות מסדר n.

בתחילה הצגנו את האלגוריתם הישיר שסיבוכיותו היא $\Theta\left(n^3\right)$ (יש לזכור כי גודל הקלט הוא: $\Theta\left(n^2\right)$).

המשכנו בהצגת אלגוריתם הפרד ופתור פשוט פותר את הבעיה על ידי **חלוקת מטריצות הקלט ל4**. כתוצאה, הבעיה המקורית מחולקת ל**8 תת בעיות הנפתרות באופן רקורסיבי.**

סיכום **(המשך)**

האלגוריתם הזה, לא שיפר את סיבוכיות האלגוריתם המקורי אך פתח את הדלת ל**אלגוריתם** Strassen אשר שיפר את הסיבוכיות על ידי מציאת דרך מתוחכמת ל**חישוב 8 תוצאות המכפלות הנדרשות** על ידי ביצוע בפועל של **7 מכפלות בלבד.**

מספר הפעמים בהן חישבנו **סכום של מטריצות עלה מ 4 ל 18**. ניתוח סיבוכיות הזמן של אלגוריתם Strassen, התקבל כתוצאה מיידית של משפט האב (Master Theorem).

סיבוכיות אלגוריתם Strassen היא $\Theta\left(n^{2.81...}\right)$.

סיכום (המשך)

אלגוריתם Strassen מראה את הדרך לשיפורים נוספים על ידי הורדת מספר המכפלות הנדרש והעלאת מספר הסכומים.

היום משתמשים בשיטות דומות אך מסובכות יותר, מחלקים כל בעיה למספר רב יותר של חלקים, במקום 4 בלבד, ומגיעים לאלגוריתמים למכפלת מטריצות בסיבוכיות זמן $O\left(n^{2.3...}\right)$.

שימו לב: אלה אלגוריתמים בעלי **ערך תיאורטי בלבד**, שכן **קבוע הפרופורציה** עולה לערכים בלתי אפשריים.

סיכום (המשך)

מה צריך לזכור מהרצאה זו?

1. פיתוח מערכת המשוואות הרקורסיביות.
2. משפט האב ושימושיו.
3. להבין את הוכחת המשפט.
4. הגדרת כפל מטריצות והאלגוריתם הישיר.
5. האלגוריתם הרקורסיבי הפשוט.
6. רעיון אלגוריתם Strassen (אין צורך לזכור את מטריצות הביניים ואת הפיתוח בעל פה.

נספח: חישוב מדויק של סיבוכיות האלגוריתם הרקורסיבי הראשון

מציאת נוסחה סגורה

כדי לחשב את סיבוכיות האלגוריתם, עלינו למצוא **נוסחה סגורה** לחישוב $T(n)$.

טענה: נתון

$$T(n) = 8T\left(\frac{n}{2}\right) + n^2 \ , \ T(1) = 1$$

אזי (ניחוש): לכל $n = 2^i$

$$T(n) = 2n^3 - n^2$$

הוכחה

עבור $n = 2^i$ ההוכחה באינדוקציה על i. כלומר, יש להניח נכונות לגבי n ולהוכיח לגבי $2n$. נשתמש בנוסחה הרקורסיבית לחישוב $T(2n)$ בעזרת $T(n)$:

$$T(2n) = 8T(n) + (2n)^2 \ =$$

בתוך נוסחה זו נציב את פתרוננו עבור $T(n)$:

$$T(2n) = 8 \cdot (2n^3 - n^2) + 4n^2$$

ונפשט כדי לקבל:

$$= 16n^3 - 4n^2 =$$

$$= 2(2n)^3 - (2n)^2$$

מש"ל

