

שיעור 9

מבוא לסיבוכיות זמן

9.1 הגדרה של סיבוכיות זמן

עד כה כל הבעיות החישוביות שעשכנו בהן הניחו שהמשאים שעומדים לרשות מכונת הטיורינג שפותרת אותן הם **בלתי מוגבלים**. כתע נ עבור לעסוק בשאלת מה קורה כאשר אנחנו מוגבלים חלק ממשאים אלו. יש סוגים רבים של ממשאים שנייתן לעסוק בהם, אבל שני הנפוצים ביותר בתיאוריה של מדעי המחשב הם

- **זמן חישוב**,

- **הזיכרון שנדרש לצורך החישוב**.

את מה בעיות שבוחן נתקלים:
 כשמעוניינים למדוד את צרכית המשאים הללו של אלגוריתם מסוים היא שלא ברור כיצד למדוד אותם.
 האם זמן חישוב נמדד בשניות?
 אם כן, כיצד ניתן לחשב את זמן החישוב עבור אלגוריתם נתון?
 האם علينا לקודד ולהריץ אותו על מחשב מסוים?

אבל במחשבים שונים האלגוריתם ירוץ זמן זנים שונים בשל

- **יעילות המעבד**,

- **אופטימיזציות שמתבצעות ברמת המעבד**,

- **אופטימיזציות בזמן הקומפליציה**,

וכיווץ בהן.

אפיו תנאים חיצוניים כמו החום בסביבת המעבד עשויים להשפיע על זמן הריצה. מכאן הרצון למצוא הגדרה **תיאורטיבית** של זמן ריצה, שאינה תלולה בחומרה זו או אחרת.

הערה 9.1

זמן הריצה של מ"ט M על קלט w , נמדד ביחס לגודל הקלט w , כלומר $(|w|) \cdot f$.

הגדרה 9.1 זמן הריצה של מכונת טיורינג

זמן הריצה של מ"ט M על קלט w היא פונקציה $f(|w|)$ השווה במספר הצעדים הנדרש בחישוב של M על w .

הגדרה 9.2 סיבוכיות זמן של בעיה/שפה

בהתנן קלט w באורך $|w| = n$. אומרים כי ניתן להכריעה שפה L בזמן $f(n)$ אם קיימת מ"ט M המכrijעה את L כך שלכל $w \in \Sigma^*$, זמן הריצה של M על w חסום ע"י $(|w|) \cdot f$.

דוגמה 9.1 סיבוכיות זמן של השפה של מחרוזות האוניריות

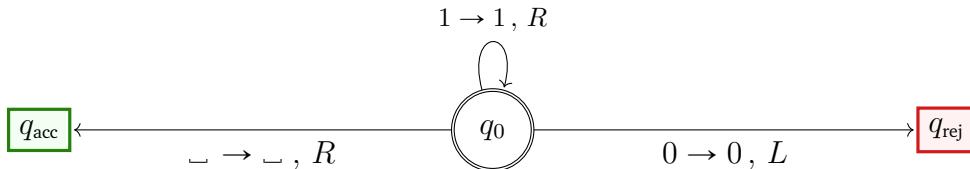
נתבונן על השפה של מחרוזות האוניריות הבאה:

$$L = \{1^n \mid n \geq 1\}.$$

נבנה מכונת טיורינג הבאה שמכריעה אותה:

$$M = (Q, q_0, \Sigma, \Gamma, \delta, q_{acc}, q_{rej})$$

כאשר $\Sigma = \{0, 1\}$ ו- $\Gamma = \{0, 1\}$. המ מצבים והמעברים מתוארים בהתרשים מצבים של מטפה:



בכדי לחשב את הסיבוכיות זמן של L בהתאם למוגנה בפסאודוקוד.

על כל קלט w : $M = M$

(1) • אם המילה היא מילת הריקה תדחה.

• אחרת ממשיכה לשלב (2).

(2) • אם התו הנקרא 0 תדחה.

• אחרת אם התו הנקרא הוא 1 מקבל.

• אחרת חוזרת לשלב (2).

כל שהקלט ארוך יותר, כך M תבצע צעדי חישוב נוספים יותר. בפרט המספר הצעדים המקסימלי הוא במקרה שבקלט w יש רק תווי 1 ולכן המ"ט תבצע $|w| = n$ צעדי חישוב. בכל מקרה אחר היא תבצע פחות מ- n צעדים.

$$\begin{aligned} &\Leftarrow \text{חסם העליון של מספר צעדי חישוב של } M \text{ על קלט } w \text{ הוא } n \text{ כאשר } |w| = n. \\ &\Leftarrow M \text{ עוצרת בזמן } O(n). \\ &\Leftarrow L \text{ כריעה בזמן } O(n). \\ &\Leftarrow L \in TIME(n). \end{aligned}$$

באופן כללי, אם מכונה על קלט w מבצעת פחות מ- $|w|$ צעדי חישוב, המשמעות היא המכונה כלל לא קראה את כל הקלט, וזה אינו נדיר במיוחד (אם כי הוא בחילוט קיים). אם כן ברור שمدידות זמן חריצת היא תמיד **ביחס לאורך הקלט**.

דוגמה 9.2

בדוגמה זו נבנה מכונת טיורינג M עם סרט ייחיד שמכריע את השפה $L = L$ ואז נחשב את הסיבוכיות זמן שלה.

בנייה המכונת טיורינג

"על קלט w : M

(1) אם התו הנקרא הוא 1 מקבל.

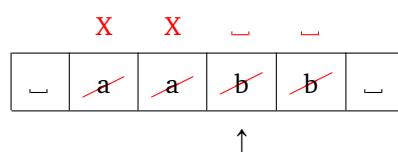
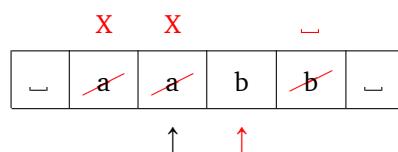
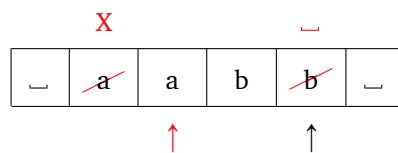
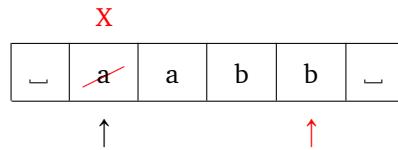
(2) אחרת אם הtau הנקרא הוא $M \Leftarrow b$ דוחה.

(3) אחרת מוחקמת את הtau הנקרא ע"י X .

(4) מזיהה את הראש ימינה עד הtau הראשון משמאלו ל- \sqsubset בקצבה הימין של הקלט.

- אם הtau הוא a או $M \Leftarrow X$ דוחה.

• אחרת מוחקמת את הtau הנקרא ע"י \sqsubset , מזיהה את הראש שמאלה עד הtau הראשון מימין ל- $\sqsubset X$ וחוזרת לשלב (1).



זמן הריצה

נסמן את אורך הקלט ב- $n = |w|$.

• M מבצעת $\left\lfloor \frac{n}{2} \right\rfloor$ איטרציות.

• בכל איטרציה M סורקת את הסרט פעמיים וזה עולה ($O(n)$ צעדים).

• לכן סה"כ הזמן הריצה של M חסום ע"י

$$\frac{n}{2} \cdot O(n) = O(n^2).$$

דוגמה 9.3

כעת נבנה מכונת טיורינג מרובת סרטים' M' שמכריעת השפה $L = \{a^n b^n \mid n \geq 0\}$ ונראה שהסיבוכיות זמן שונה מזו של המכונת טיורינג עם סרט יחיד שראינו בדוגמה הקודמת.

בנייה המכונה

:w קלט ע"ל = M'

- .*O(n)* (1) מעתיקת את ה- b -ים לסרטט 2 (ותוך כדי לבדוק האם w מהצורה a^*b^*).

.*O(n)* (2) מזיהה את הראשים לתחילת הסרטים.

(3) אם שני הראשים מצביעים על $_ \Rightarrow$ מקבלת.

(4) אם אחד הראשים מצביע על $_ \text{ והשני לא} \Rightarrow$ לא.

(5) מזיהה את שהע הראשים ימינה וחוזרת לשלב (3).".

שלבים (3-5): *O(n)*

. $O(n)$: (3-5)

(1) סרט _____ a a b b _____

(2) סרט _____ b b _____

סיבות זמן

. $n = |w|$ נסמן את אורך הקלט ב-

זמן הריצה של M' הוא $O(n)$

הגדלה $TIME(f(n))$ 9.3

נגדיר הקבוצת השפות $\text{TIME}(f(n))$ כך שלכל שפה $L \in \text{TIME}(f(n))$ קיימת מכונת טיריניג דטרמיניסטית שמכריעה את L בזמן לכל היותר $f(n)$, כאשר n הוא האורך של הקלט:

$$TIME(f(n)) = \{L \mid O(f(n)) \text{ זמני } L \text{ בזמנו } f(n)\}.$$

9.4 דוגמה

عبر השפה בדוגמה 9.2:

$$L \in \text{TIME}(n^2) .$$

9.5 דוגמה

عبر השפה בדוגמה 9.3:

$$L \in TIME(n) \enspace .$$

9.2 יחס בין הסיבוכיות של מכונת טיורינג עם סרט יחיד ומכונת טיורינג מרובת סרטים

משפט 9.1

לכל מ"ט מרובה סרטים M הרצה בזמן $f(n)$ קיימת מ"ט סרט יחיד M' השකולה לו - M ורצה בזמן $O(f^2(n))$.

הוכחה:

בhinתן מ"ט מרובה סרטים M , הרצה בזמן $f(n)$, נבנה מ"ט עם סרט יחיד M' באותו אופן כמו בהוכחת השקילות במשפט 3.1.

כלומר, M' שומרת את התוכן של k סרטים של M על הסרט היחיד שלו (עם הפרדה ע"י $\#$), ובכל צעד חישוב, M' סורקת את הסרט שלו כדי לאזות שת האותיות שמתחאת לראשים (שמסומנות ב- $\hat{\alpha}$) ואחרי זה, משתמש בפונקציות המעברים של M , וسورקת את הסרט פעם נוספת כדי לעדכן את התוכן בכל אחד מהסרטים ואת מיקום הראש בכל אחד מהסרטים.

1)

2)

⋮
⋮
⋮

k)

#	$\hat{\alpha}_1$	#	$\hat{\alpha}_2$	#	$\hat{\alpha}_3$	#	
---	------------------	---	------------------	---	------------------	---	--

כמה לוקח לו M' לסרוק את הסרט שלו? מכיוון שהסרט של M' מכיל את התוכן של k הסרטים של M , והגודל של כל אחד מהסרטים של M חסום ע"י $f(n)$, גודל הסרט של M' חסום ע"י

$$k \cdot f(n) = O(f(n)) .$$

העלות של הסקירה של M' הסרט שלו היא $O(f(n))$ וזה עלות של צעד חישוב בריצה של M' על הקלט.

מכיוון ש- M רצהה בזמן $(n)^f$, זמן היצרה של M חסום ע"י

$$f(n) \cdot O(f(n)) = O(f^2(n)) .$$



9.3 יחס בין הסיבוכיות של מ"ט דטרמיניסטי ומ"ט א"ד

משפט 9.2

לכל מ"ט א"ד N הרצהה בזמן $(n)^{f(n)}$, קיימת מ"ט דטרמיניסטי D השකולה לו N ורצתה בזמן $2^{(f(n))}$.

הוכחה:

בhinתן מ"ט א"ד N הרצתה בזמן $f(n)$ מ"ט דטרמיניסטי D באותו אופן כמו בהוכחת השקילות המשפט 4.1. כולם, בהינתן קלט w , D תסרו את עץ החישוב של N ו- w לרווח ותקבל כל אחד החישובים של N המסתויים ב- q_{acc} .

בhinתן קלט w באורך n :

- כל מסלול בעץ החישוב של N על w חסום ע"י $f(n)$.
- מסטת החישובים ש- D מבצעת חסום ע"י מספר הקודקודים בעץ החישוב של N ו- w .
- מכיוון שמספר הבנים של כל קודקוד בעץ החישוב חסום ע"י

$$C = 3|Q| \cdot |\Gamma|$$

מספר הקודקודים בעץ החישוב חסום ע"י

$$C^0 + C^2 + \dots + C^{f(n)} \leq C^{f(n)+1} = C \cdot C^{f(n)} .$$

ולכן זמן הריצה של D חסום ע"י

$$f(n) \cdot C \cdot C^{f(n)} \leq C^{f(n)} \cdot C^{f(n)} = C^{2f(n)} = (C^2)^{f(n)} = 2^{C' \cdot f(n)} = 2^{O(f(n))} .$$

נתיחה לכך לשני החסמים הבאים:



1) חסם פולינומיAli הוא חסם מהצורה n^c עבור $c > 0$ כלשהו.

2) חסם אקספוננציאלי הוא חסם מהצורה 2^{n^c} עבור $c > 0$ כלשהו.

9.4 החלוקת P

הגדרה 9.4 בעיית הכרעה

בעיית הכרעה מוגדרת באופן הבא:

"בhinתן קלט כלשהו, האם הקלט מקיים תנאי מסוים "

דוגמה 9.6

בhinתן מספר n , האם n ראשוני?

כל בעיית הכרעה ניתן לתאר כפשה שකולה:

$$L_{\text{prime}} = \{\langle n \rangle \mid n \text{ ראשוני}\}.$$

משפט 9.3

. שפה \equiv בעיית הכרעה

הגדרה 9.5 אלגוריתם זמן פולינומיAli

אומרים כי אלגוריתם A מכרייעת בעיה בזמן פולינומיAli אם קיים קבוע $c > 0$ כך שזמן הריצה של A על כל קלט w חסום ע"י $O(|w|^c)$.

משפט 9.4 התזה של צירץ' (Church Thesis)

אם קיים אלגוריתם המכרייע בעיה בזמן פולינומיAli, אז קיימת מכונת טיורינג דטרמיניסטיבית המכרייע את השפה השקולה לבעיה זו בזמן פולינומיAli.

. מכונת טיורינג \equiv אלגוריתם מכרייע

הגדרה 9.6 המחלקה P

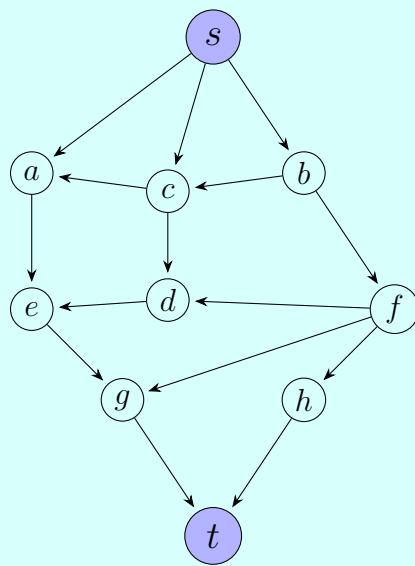
המחלקה P היא אוסף כל הבעיות (השפות) שקיימים עבורן אלגוריתם (מכונת טיורינג דטרמיניסטיבית) המכרייע אותן בזמן פולינומיAli.

דוגמה 9.7

$$L = \{a^n b^n \mid n \geq 0\} \in P.$$

9.5 בעיית PATH

הגדלה 9.7 בעיית המסלול בגרף מכוון



קלט: גראף מכוון $G = (V, E)$ ושני קודקודים $s, t \in V$

פלט: האם קיים מסלול ב- G מ- s ל- t ?

$$PATH = \{ \langle G, s, t \rangle \mid \text{קיים מסלול ב- } G \text{ מ- } s \text{ ל- } t \}$$

משפט 9.5

$$PATH \in P .$$

הוכחה: נבנה אלגוריתם A עבור הבעיה $.PATH$

$$:\langle G, s, t \rangle = A$$

1) צובע את s .

2) מבצע $|V| - 1$ פעמים:

- לכל צלע $(u, v) \in E$:
- * אם u צבוע \Leftarrow צבע את v .
- אם t צבוע \Leftarrow החזיר "כן".
- אחרת \Leftarrow החזיר "לא".

זמן ריצה של האלגוריתם הוא $O(|V| \cdot |E|)$ פולינומיAli, במספר הקודקודים $|V|$.

אם זה פולינומיAli בגודל הקלט ?

איך נקודד את G ?

- נניח כי $n = |V|$

- נניח כי הצלעות נתונות ע"י מטריצה M בגודל $n \times n$ כך ש-

$$M_{ij} = \begin{cases} 1 & (i,j) \in E \\ 0 & (i,j) \notin E \end{cases}.$$

- נניח כי מספרים מקודדים בסיס ביניארי.

- אזי גודל הקידוד של G שווה $n \log_2 n + n^2$, כלומר

$$|\langle G \rangle| = \Omega(|V|^2) \Rightarrow |V| = O(|\langle G \rangle|).$$

ולכן כל אלגוריתם הרץ בזמן פולינומיائي במספר הקודוקודים $|V|$ ירוץ בזמן פולינומיائي בגודל הקידוד $|\langle G \rangle|$.

■ ולכן A רץ בזמן פולינומיائي בגודל הקלט.

RELPRIME 9.6 הביעית

הגדרה 9.8 מספרים זרים (Relatively prime)

אומרים כי שני מספרים שלמים y, x הם זרים אם המחלק המשותף הגדול ביותר, מסומן $\gcd(x, y)$, שווה 1.

הגדרה 9.9 ביעית RELPRIME

קלט: שני מספרים x ו- y .

פלט: האם x ו- y זרים?

$$\text{RELPRIME} = \{\langle x, y \rangle \mid \gcd(x, y) = 1\}.$$

אנחנו נוכיח כי ניתן להכrie את RELPRIME בזמן פולינומיائي, כלומר נוכיח $\text{RELPRIME} \in P$ במשפט 9.8 למטה. לפניכן נסביר את האלגוריתם של אוקליזד למציאת ה- \gcd של שני שלמים, ומכיון זה נוכל לחשב את הסיבוכיות זמן של RELPRIME . ראשית נזכיר משפט שלמדו בקורסים קודמים:

משפט 9.6 המשפט של האלגוריתם של אוקליזד

אם x, y שלמים אז

$$\gcd(x, y) = \begin{cases} x & y = 0 \\ \gcd(y, x \bmod y) & y \neq 0 \end{cases}.$$

■ הוכחה: ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטי שימושיים" בדף 105.

האלגוריתם של אוקליזד הוא אלגוריתם, שמקבל קלט שני מספרים y, x וpollט את $\gcd(x, y)$. הוא מתבסס על המשפט 9.6. האלגוריתם עצמו הוא כדלקמן:

: x, y על קלט = EUCLID

(1) כל עוד $0 \neq y$:

$x \leftarrow x \bmod y$ (2)

swap(x, y) (3)

(כלומר מחליפים בין x ו- y).
.

(4) מחזירים את x .

לדוגמא:

$$\gcd(18, 32) = \gcd(32, 18) = \gcd(18, 14) = \gcd(14, 4) = \gcd(4, 2) = \gcd(2, 0) = 2 .$$

כדי להוכיח כי $\text{RELPRIME} \in P$ נצטרך למשפט עזר הבא:

משפט 9.7 (משפט עזר)

$$x \bmod y < \frac{x}{2} \text{ או } x > y$$

■ הוכחה: ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטיים שימושיים" בדף 105.

משפט 9.8

$$\text{RELPRIME} \in P .$$

הוכחה:

בננה אלגוריתם A המכريع את RELPRIME בזמן פולינומיAli. RELPRIME היא השפה של הבעיה, שמקבלת כקלט שני מספרים שלמים y, x ומהזירה תשובה לשאלת, האם y, x זרים. כלומר:

$$\langle x, y \rangle \in \text{RELPRIME} \iff \gcd(x, y) = 1 .$$

לכן A משתמש באלגוריתם של אוקלידס ($\text{EUCLID}(x, y)$) כדי לחשב

בנייה האלגוריתם A המכريع

"על קלט $\langle x, y \rangle$: מרים את EUCLID על x ו- y .
.

- אם $\gcd(x, y) = 1$ מוחזר $\text{EUCLID}(x, y)$ אז A מקבל.

- אחרת A דוחה.

הוכחת הנכונות

הנכונות של A מובעת ישירות מהנכונות של האלגוריתם האוקלידס, EUCLID .

סיבוכיות זמן

נראה כי A רץ בזמן פולינומיAli בגודל הקלט $\langle x, y \rangle$.

- לפי משפט עזר 9.7: $x \bmod y < \frac{x}{2}$

- בכל איטרציה, בשלב (2) המשתנה x מקבל את הערך החדש $x \bmod y$.

- לכן בכל איטרציה הערך החדש של x קטן ממש מחצי של הערך הקודם של x .

- לכן אחרי כל איטרציה, x קטן לפחות חצי.

- בשלב (3), A מחליף בין x ו- y , אז אחרי כל 2 איטרציות, גם x קטן לפחות חצי וגם y קטן לפחות חצי.
- לכן המספר המקסימלי של פעמים שאפשר לבצע שלבי (2) ו- (3) היא $\min(2\lfloor \log_2 x \rfloor, 2\lfloor \log_2 y \rfloor)$.
- לכן המספר המקסימלי של האיטרציות ש $EUCLID$ מבצע הוא $\lfloor y \rfloor$.
- וזה בדיק זמן הריצה של האלגוריתם A .

ולכן A רץ בזמן פולינומיAli בגודל הקלט.

ולכן

$$RELPRIME \in P.$$



9.7 *הוכחות של משפטי שימושיים

משפט 9.9 השפט של האלגוריתם של אוקלידס

אם x, y שלמים אז

$$\gcd(x, y) = \begin{cases} x & y = 0 \\ \gcd(y, x \bmod y) & y \neq 0 \end{cases}.$$

הוכחה: (להעשרה בלבד)

המטרה של ההוכחה זו היא רק להוסיף הבנה להוכחה של משפט 9.8 לסיבוכיות זמן של $RELPRIME$ למטה. היא לא הוכחה שאתם תיבחנו עלייה ואפשר לדלג עליה.

נתחיל אם משפט החילוק של אוקלידס, שאומר שאם x, y שלמים אז קיימים שלמים q ו- r כך ש:

$$x = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \bmod y). \quad (1*)$$

נגיד $d \triangleq \gcd(x, y)$. מכיוון ש- d הוא מחלק משותף של x ו- y אז $d \mid x$ ו- $d \mid y$. לכן בזכות משווהה (1*):

$$(d \mid x) \wedge (d \mid y) \xrightarrow{\text{משווהה (1*)}} d \mid (x \bmod y)$$

וז"א וגם $d \mid (x \bmod y)$ אז בהכרח:

$$d \mid \gcd(y, x \bmod y). \quad (2*)$$

כעת נגיד $\bar{d} \triangleq \gcd(y, y \bmod x)$.

מכיוון ש- \bar{d} הוא מחלק משותף של y ו- $y \bmod x$ אז $\bar{d} \mid y$ ו- $\bar{d} \mid y \bmod x$. לכן בזכות משווהה (1*):

$$(\bar{d} \mid y) \wedge (\bar{d} \mid x \bmod y) \xrightarrow{\text{משווהה (1*)}} \bar{d} \mid x$$

וז"א וגם $\bar{d} \mid x$ אז בהכרח:

$$\bar{d} \mid \gcd(x, y). \quad (3*)$$

לסיכום, לפי משווהות (2*) ו- (3*):

$$d \mid \bar{d} \wedge \bar{d} \mid d.$$

מכיוון ש- $\gcd(x, y) = \gcd(y, x \bmod y)$ אז בהכרח $d = \bar{d}$, $\bar{d} > 0$.



משפט 9.10 (משפט עזר)

$$\text{אם } x \bmod y < \frac{x}{2} \text{ או } x > y$$

הוכחה: יש שני מקרים:

$$y \leq \frac{x}{2} \quad (1)$$

$$y > \frac{x}{2} \quad (2)$$

נוכיח את הטענה עבור שני המקרים.

$$\text{מקרה 1: } y \leq \frac{x}{2}$$

לפי משפט החלוק של אוקלידס אם x, y שלמים Überom $y > x$ אז קיימים כך $r = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y$

$$x = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \bmod y) .$$

$$x \bmod y < y \leq \frac{x}{2} \text{ לפיכך } r < y \text{ ובפרט } y < r < y$$

$$\text{מקרה 2: } y > \frac{x}{2}$$

לפי משפט החלוק של אוקלידס אם x, y שלמים Überom $y > x$ אז קיימים שלם $r = x \bmod y$ ושלם $q = \left\lfloor \frac{x}{y} \right\rfloor$ כך $0 \leq r < y$

$$x = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \bmod y) .$$

בפרט אם $y > x$ אז $x < 2y$. מכיוון ש- $q < 2$. הוא $q = 1$. לכן אם $q = 1$ אז $x < 2y$.

$$x = qy + r = (1)(y) + ry + (x \bmod y) .$$

מכאן

$$x - y = x \bmod y .$$

כעת נציב את ההנחה ההתחלהית $x - y < \frac{x}{2} \Leftarrow y > \frac{x}{2}$ ונקבל

$$x \bmod y < \frac{x}{2} .$$

