שיעור 9
מבוא לסיבוכיות זמן

9.1 הגדרה של סיבוכיות זמן

עד כה כל הבעיות החישוביות שעסקנו בהן הניחו שהמשאבים שעומדים לרשות מכונת הטיורינג שפותרת אותן הם **בלתי מוגבלים**. כעת נעבור לעסוק בשאלה מה קורה כאשר אנחנו מגבילים חלק ממשאבים אלו. יש סוגים רבים של משאבים שניתן לעסוק בהם, אבל שני הנפוצים ביותר בתיאוריה של מדעי המחשב הם

- **זמן החישוב**,
- **הזיכרון שנדרש לצורך החישוב**.

אחת מהבעיות שבהן נתקלים:
כשמעוניינים למדוד את צריכת המשאבים הללו של אלגוריתם מסויים היא שלא ברור כיצד למדוד אותם.
האם זמן חישוב נמדד בשניות?
אם כן, כיצד ניתן לחשב את זמן החישוב עבור אלגוריתם נתון?
האם עלינו לקודד ולהריץ אותו על מחשב מסוים?

אבל במחשבים שונים האלגוריתם ירוץ זמנים שונים בשל

- יעילות המעבד,
- אופטימיזציות שמתבצעות ברמת המעבד,
- אופטימיזציות בזמן הקומפליצה,

וכיוצא בהן.

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

הערה 9.1

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

הגדרה 9.1 זמן הריצה של מכונת טיורינג

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

הגדרה 9.2 סיבוכיות זמן של בעיה/שפה

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

דוגמה 9.1 סיבוכיות זמן של השפה של מחרוזות האוניריות

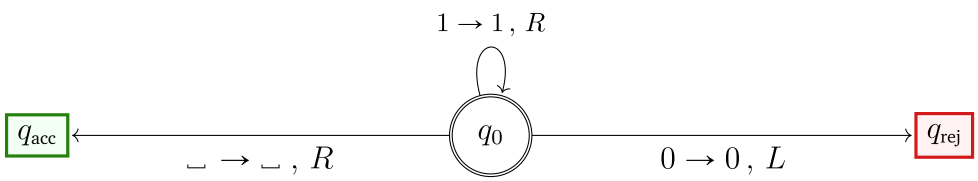
נתבונן על השפה של מחרוזות האוניריות הבאה:

$$L = \left\{ 1^n \;\middle|\; n \geqslant 1 \right\} \ .$$

נבנה מכונת טיורינג הבאה שמכריעה אותה:

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

כאשר $\Sigma = \{0,1\}$ ו- $\Gamma = \{0,1,\text{␣}\}$. המצבים והמעברים מתוארים בהתרשים מצבים שלמטה:



בכדי לחשב את הסיבוכיות זמן של L נתאר את המכונה בפסאודוקוד.
$M =$ על כל קלט w:

(1
- אם המילה היא מילת הריקה תדחה.
- אחרת ממשיכה לשלב 2).

(2
- אם התו הנקרא 0 תדחה.
- אחרת אם התו הנקרא הוא ␣ תקבל.
- אחרת חוזרת לשלב 2).

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

$\Leftarrow$ חסם העליון של מספר צעדי חישוב של M על קלט w הוא n כאשר $n = |w|$.

$\Leftarrow$ M עוצרת בזמן $O(n)$

$\Leftarrow$ L כריעה בזמן $O(n)$.

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

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

דוגמה 9.2

בדוגמה הזו נבנה מכונת טיורינג M עם סרט יחיד שמכריעה את השפה $L = \left\{ a^n b^n \;\middle|\; n \geqslant 0 \right\}$ ואז נחשב את הסיבוכיות זמן שלה.

<u>**בניית המכונת טיורינג**</u>

$M =$ "על קלט w:

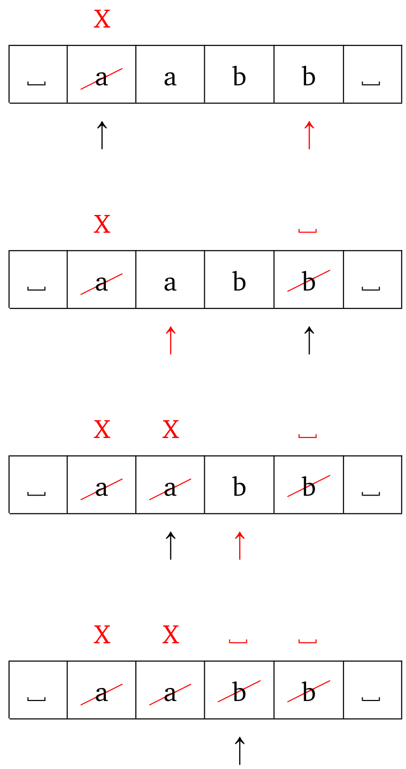
(1) אם התו הנקרא הוא ␣ $\Leftarrow$ M תקבלת.

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

(3) אחרת מוחקת את התו הנקרא ע"י X.

(4) מזיזה את הראש ימינה עד התו הראשון משמאל ל- ␣ בקצה הימין של הקלט.

- אם התו הוא a או $X \Leftarrow M$ דוחה.
- אחרת מוחקת את התו הנקרא ע"י ␣, מזיזה את הראש שמאלה עד התו הראשון מימין ל- 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

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

9.3 דוגמה

כעת נבנה מכונת טיורינג מרובת סרטים M' שמכריעה את השפה $L = \left\{a^n b^n \mid n \geqslant 0\right\}$ ונראה שהסיבוכיות זמן שונה מזה של המכונת טיורינג עם סרט יחיד שראינו בדוגמה הקודמת.

בניית המכונה

$M' =$ "על קלט w:

(1) מעתיקה את ה- b-ים לסרט 2 (ותוך כדי בודקת האם w מהצורה a^*b^*). $O(n)$.

(2) מזיזה את הראשים לתחילת הסרטים. $O(n)$.

(3) אם שני הראשען מצביעים על $\sqcup$ $\Leftarrow$ מקבלת.

(4) אם אחד הראשים מצביע על $\sqcup$ והשני לא $\Leftarrow$ תדחה.

(5) מזיזה את שני הראשים ימינה וחוזרת לשלב (3)." שלבים (3-5): $O(n)$.

סרט (1)

$\sqcup$	a	a	b	b	$\sqcup$

סרט (2)

$\sqcup$	b	b			$\sqcup$

סיבוכיות זמן

נסמן את אורך הקלט ב- $n = |w|$.

הזמן הריצה של M' הוא $O(n)$.

הגדרה 9.3 $TIME(f(n))$

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

$$TIME(f(n)) = \{L \mid \text{קיימת מ"ט דטרמיניסטית המכריעה } L \text{ בזמן } O(f(n))\} \ .$$

דוגמה 9.4

עבור השפה בדוגמה 9.2:

$$L \in TIME(n^2) \ .$$

דוגמה 9.5

עבור השפה בדוגמה 9.3:

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

דוגמה 9.6

הדוגמה הזו ראינו בדוגמה 1.5. נבנה מכונת טיורינג M עם סרט יחיד שמכריעה את השפה $L = \left\{a^{2^n} \mid n \geqslant 0\right\}$ ואז נחשב את הסיבוכיות זמן שלה.

הרעיון

המכונה תסרוק את הסרט שלה משמאל לימין ובכל איטרציה תמחק חצי מה- a-ים שנשארו.

בניית המכונה

$M=$ " על קלט w:

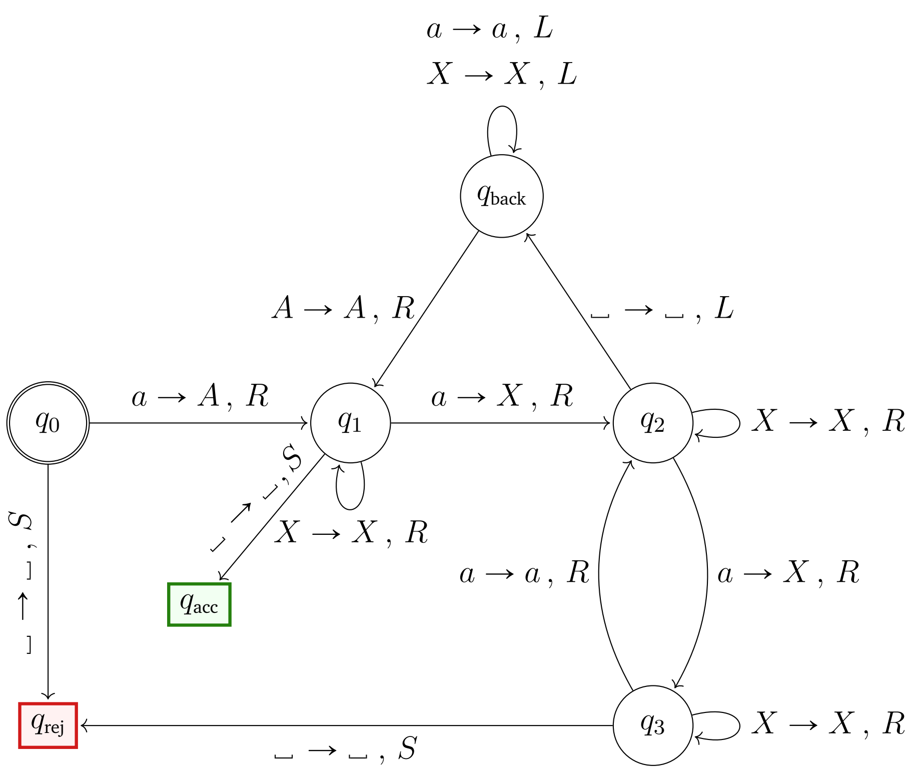
(1) אם התו הנקרא הוא $\sqcup$ $\Leftarrow$ דוחה.

(2) אחרת מחליפה את ה- a הראשון ב- A (כדי לסמן את תחילת הסרט).

(3) סורקת את הסרט משמאל לימין ומוחקת כל a שני:

- אם הסרט מכיל a יחיד $\Leftarrow$ מקבלת.
- אם הסרט מכיל מספר אי-זוגי של a -ים $\Leftarrow$ דוחה.
- מזיזה את הראש שמאלה לתחילת הסרט וחוזרת לשלב (3)."

התרשים מצבים של המכונה מתואר באיור למטה:



<u>**סיבוכיות זמן**</u>

- שלב (3) דורש n צעדים לכל היותר.
- המכונה חוזרת על שלב (3) לכל היותר n פעמים.
- לפיכך הסיבוכיות זמן של השפה L היא $O\left(n^2\right)$. כלומר:

$$L \in TIME\left(n^2\right)\ .$$

דוגמה 9.7 חיסור בינארי

בדוגמה זו נבנה מכונת טיורינג מ- 3 סרטים שמקבלת כקלט שני מספרים שלמים x,y $(x>y)$ בבסיס בינארי ומחשבת את החיסור $x-y$.

<u>**בניית המכונה**</u>

$SUBTRACT =$ " על קלט $\langle x, y \rangle$ כאשר x, y שלמים בבסיס בינארי ו- $x > y$:

1) רושמת את x בסרט 1 ואת y בסרט 2 משמאל לימין כך שהתאים עם הקצוות הימניות של x ו-y מיושרים זה מתחת לזה. בתחילה סרט 3 ריק.

2) מעמידה את ראשי סרטים $1, 2, 3$ על התאים המיושרים של הקצוות הימניים של הקלטים. בפרט, ראשי סרטים 1 ו-2 נמצאים על הספרות הפחות משמעותיות של x ו-y, וראש סרט 3 נמצא בתא שמתחתם.

3) שלב זה מבצע חיסור ללא חוב.

יהי $\sigma_1 \in \{0, 1, \sqcup\}$ תו הנקרא בסרט 1 ויהי $\sigma_2 \in \{0, 1, \sqcup\}$ תו הנקרא בסרט 2.

- אם $(\sigma_1, \sigma_2) = (\sqcup, \sqcup)$ אז M מקבלת.
- אם $(\sigma_1, \sigma_2) = (0, 0)$, כותבת $(0, 0, 0)$, מזיזה את הראשים (L, L, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (1, 0)$, כותבת $(1, 0, 1)$, מזיזה את הראשים (L, L, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (1, 1)$, כותבת $(1, 1, 0)$, מזיזה את הראשים (L, L, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (\sqcup, 0)$, כותבת $(\sqcup, 0, 0)$, מזיזה את הראשים (S, L, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (\sqcup, 1)$, כותבת $(\sqcup, 1, 1)$, מזיזה את הראשים (S, L, L) ועוברת לשלב הבא.
- אם $(\sigma_1, \sigma_2) = (0, \sqcup)$, כותבת $(0, \sqcup, 0)$, מזיזה את הראשים (L, S, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (1, \sqcup)$, כותבת $(1, \sqcup, 1)$, מזיזה את הראשים (L, S, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (0, 1)$, כותבת $(0, 1, 1)$, מזיזה את הראשים (L, L, L) ועוברת לשלב הבא.

4) שלב זה מבצע חיסור כאשר קיים חוב.

- אם $(\sigma_1, \sigma_2) = (1, 0)$, כותבת $(1, 0, 0)$, מזיזה את הראשים (L, L, L) ועוברת לשלב הקודם.
- אם $(\sigma_1, \sigma_2) = (0, 1)$, כותבת $(0, 1, 1)$, מזיזה את הראשים (L, L, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (1, 1)$, כותבת $(1, 1, 1)$, מזיזה את הראשים (L, L, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (0, 0)$, כותבת $(0, 0, 1)$, מזיזה את הראשים (L, L, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (\sqcup, 0)$, כותבת $(\sqcup, 0, 1)$, מזיזה את הראשים (S, L, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (\sqcup, 1)$, כותבת $(\sqcup, 1, 0)$, מזיזה את הראשים (S, L, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (0, \sqcup)$, כותבת $(0, \sqcup, 1)$, מזיזה את הראשים (L, S, L) וחוזרת על שלב זה.
- אם $(\sigma_1, \sigma_2) = (1, \sqcup)$, כותבת $(1, \sqcup, 1)$, מזיזה את הראשים (L, S, L) ועוברת לשלב הקודם."

סיבוכיות זמן

יהי $n = \max(|x|, |y|)$. המכונה $SUBTRACT$ סורקת את השני סרטים שבהם כתובים המספרים x ו- y במקביל, ותוך כדי רושמת את הפלט על סרט 3. לכן $SUBTRACT$ מבצעת $(O(n))$ צעדים.

לכן הסיבוכיות זמן של $SUBTRACT$ היא $O(n)$ (ליניארי).

דוגמה 9.8 האלגוריתם החילוק של אוקלידס

המשפט החילוק של אוקלידס אומר שבהינתן שני מספרים שלמים x,y, אזי קיימים שלמים q,r כך ש

$$x=qy+r\ ,\qquad 0\leqslant r<|y|\ .$$

במקרה הפרטי ש- $0<y<x$ אז:

$$q=\left\lfloor\frac{x}{y}\right\rfloor\ ,\qquad r=x \bmod y\ .$$

קיים אלגוריתם שמקבל כקלט x,y ונותן כפלט q ו- r. האלגוריתם עובד לכל שלמים x,y (בלי קשר לסימן שלהם). אנחנו נסתכל על האלגוריתם במקרה הפרטי ש- $0<y<x$, כמתואר למטה.

<u>בניית האלגוריתם</u>

$DIVISION=$ " על קלט $\langle x,y\rangle$ כאשר x,y שלמים בביס בינארי ו- $0<y<x$:

1) מאתחל $\left.\begin{array}{l} q\leftarrow 0\\ r\leftarrow x\end{array}\right\}$

2) כל עוד ש $r\geqslant y$:

3) $r\leftarrow r-y$

4) $q\leftarrow q+1$

5) פלט: q,r."

<u>סיבוכיות זמן</u>

נסמן $n=|\langle x,y\rangle|$ אורך הקלט.

- $DIVISION$ מבצעת r איטרציות לכל היותר.
- $r<y$ לכן y הוא חסם עליון של המספר האיטרציות המקסימלי של $DIVISION$.
- לכן $DIVISION$ מבצע $O(n)$ איטרציות לכל היותר.
- בכל איטרציה $DIVISION$ מבצע חיסור בינארי עם $SUBTRACT$ אשר (כפי שראינו בדוגמה 9.7) הוא $O(n)$.
- לכן הסיבוכיות זמן של $DIVISION$ היא

$$O\left(n^2\right)\ .$$

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

משפט 9.1

לכל מכונת טיורינג מרובת סרטים M הרצה בזמן $t(n)$ קיימת מכונת טיורינג סרט יחיד M' השקולה ל- M ורצה בזמן $O\left(t^2(n)\right)$.

הוכחה:

תהי M מכונת טיורינג כלשהי עם k סרטים הרצה בזמן $O\left(t(n)\right)$.

נבנה מכונת טיורינג S עם סרט אחד שרצה בזמן $O\left(t^2(n)\right)$.

- ראשית נרשום את התוכן של ה- k סרטים של M על הסרט היחיד של S.
- התכנים של הסרטים מופרדים על ידי תו # על הסרט היחיד.
- המיקומים של הראשים של כל הסרטים של M מסומנים על הסרט היחיד על ידי חצים.

להלן יש דוגמה לכך של מכונת טיורינג עם 3 סרטים:

<u>המכונת טיורינג M</u>

0	1	0	1	0	␣	␣
			↑			

a	a	a	␣	␣	␣	␣
		↑				

a	b	b	␣	␣	␣	␣
	↑					

<u>המכונת טיורינג S</u>

#	0	1	0	1	0	#	a	a	a	#	a	b	b
				↑					↑			↑	

אפשר לסמלץ צעד חישוב אחד של M במכונת טיורינג S באופן הבא:

<u>**בניית המכונה S**</u>

1) תחילה S מאתחלת את הסרט שלה בלכתוב את התכנים של ה- k סרטים על הסרט היחיד שלה, עם תו # להפריד בין סרטים שונים של M.

2) בנוסף S רושמת תו # בקצה השמאלי כדי לסמן את התחילת הקלט ותו # בקצה הימין כדי לסמן את סוף הקלט.

3) בכדי לסמלץ צעד חישוב אחד של M בהמכונה S, המכונה S סורקת את הסרט מ- # הראשון ל- # האחרון. בסריקה זו S זוכרת את המיקומים של ה- k ראשים על פי התאים שמסומנים עם חצים, באמצעות k תאי זכרון.

4) אחר כך S מבצעת סריקה שנייה של הסרט. בסריקה זו, לפי הפונקצית המעברים של M, המכונה S מבצעת, לכל $1 \leqslant i \leqslant k$:

- כתיבה של הסימן החדש בסרט ה-i במיקום הנוכחי של הראש של סרט ה-i,
- תזוזה של הראש של סרט ה- i.

5) במצב שהראש של אחד הסרטים של M זז ימינה לתו רווח מצד ימין של סוף הקלט, אז S תוסיף תא אחד עם תו רווח מצד שמאל של ה- # המפריד בין סרטים, ואחר כך היא תזיז את כל התאים שבצד ימין של התא המוסף מקום אחד ימינה.

סיבוכיות זמן של המכונה S

יהי n האורך של התוכן הכי ארוך מתוך כל התכנים של ה- k סרטים של M.

- האורך של התוכן של S שווה לסכום הארכים של התכנים של ה- k סרטים של M.
- נתון שהזמן הריצה של M הוא $O(t(n))$.

$\Leftarrow$ M עושה שימוש של $O(t(n))$ תאים.

$\Leftarrow$ אורך הקלט של S חסום מלמעלה ע"י $O(t(n))$.

$\Leftarrow$ סריקה שלמה של הקלט של S מתבצעת ב- $O(t(n))$ צעדים.

$\Leftarrow$ מכיוון שכל סריקה דורשת זמן $O(t(n))$ וכדי לדמות צעד אחד של M, המכונה S מבצעת שתי סריקות, אז S לוקחת זמן $O(t(n))$ לבצע צעד חישוב אחד של M.

$\Leftarrow$ הסיבוכיות זמן של S היא:

$$O(t(n)) \ \ O(t(n)) = O\left(t^2(n)\right)$$

■

9.3 יחס בין הסיבוכיות של מכונת טיורינג דטרמיניסטית ומכונת טיורינג אי דטרמיניסטית

משפט 9.2

לכל מכונת טיורינג אי-דטרמיניסטית N הרצה בזמן $f(n)$, קיימת מכונת טיורינג דטרמיניסטית D השקולה ל- N ורצה בזמן $2^{(f(n))}$.

הוכחה:

בהינתן מכונת טיורינג אי-דטרמיניסטית N הרצה בזמן $f(n)$ מכונת טיורינג דטרמיניסטית D באותו אופן כמו בהוכחת השקילות במשפט 4.1.

כלומר, בהינתן קלט w, D תסרוק את עץ החישוב של N ו- w לרוחב ותקבל כל אחד החישובים של N המסתיים ב- q_{acc}.

בהינתן קלט w באורך n:

- כל מסלול בעץ החישוב של N על w חסום ע"י $f(n)$.
- מספר 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

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

ולכן זמן הריצה של D חסום ע"י

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

נתייחס כאן לשני החסמים הבאים:

■

(1 חסם פולינומיאלי הוא חסם מהצורה n^c עבור $c > 0$ כלשהו.

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

9.4 המחלקה P

הגדרה 9.4 בעיית הכרעה

בעיית הכרעה מוגדרת באופן הבא:

"בהינתן קלט כלשהו, האם הקלט מקיים תנאי מסוים"

דוגמה 9.9

בהינתן מספר n, האם n ראשוני?

כל בעיית הכרעה ניתן לתאר כשפה שקולה:

$$L_{\text{prime}} = \left\{ \langle n \rangle \;\middle|\; \text{ראשוני } n \right\} \ .$$

משפט 9.3

שפה $\equiv$ בעיית הכרעה .

הגדרה 9.5 אלגוריתם זמן פולינומיאלי

אומרים כי אלגוריתם A מכריעה בעייה בזמן פולינומיאלי אם קיים קבוע $c > 0$ כך שזמן הריצה של A על כל קלט w חסום ע"י $O\left(|w|^c\right)$.

משפט 9.4 התזה של צירץ' (Church Thesis)

אם קיים אלגוריתם המכריע בעייה בזמן פולינומיאלי, אז קיימת מכונת טיורינג דטרמיניסטית המכריעה את השפה השקולה לבעייה זו בזמן פולינומיאלי.

$$\text{מכונת טיורינג} \equiv \text{אלגוריתם מכריעה} .$$

הגדרה 9.6 המחלקה P

המחלקה P היא אוסף כל הבעיות (השפות) שקיים עבורן אלגוריתם (מכונת טיורינג דטרמיניסטית) המכריע אותן בזמן פולינומיאלי.

דוגמה 9.10

בדוגמה 9.2 הוכחנו כי קיימת מכונת טיורינג דטרמייסטית שמכריעה עת השפה

$$L = \left\{a^n b^n \;\middle|\; n \geqslant 0\right\} \in P$$

בזמן $O\left(n^2\right)$. לכן $L \in P$.

דוגמה 9.11

תהי L_{prime} השפה הבאה:

$$L_{\text{prime}} = \left\{\langle n\rangle \;\middle|\; \text{ראשוני } n\right\} .$$

הוכיחו: $L_{\text{prime}} \in P$

פתרון:

כדי להוכיח כי $L_{\text{prime}} \in P$, נבנה אלגוריתם שמכריע את L_{prime} ואחר כך נראה כי האלגוריתם רץ בזמן פולינומיאלי.

האלגוריתם עצמו עושה שימוש של המשפט הבא:

<u>משפט: למספר פריק יש מחלק הקטן מ- או שווה לשורשו</u>

יהי n מספר שלם. אם n פריק אז קיים שלם $\sqrt{n}\geqslant a$ שמחלק את n.

<u>הוכחה:</u>

- נניח בשלילה כי n פריק ולא קיים מחלק d של n כך ש- $d \leqslant \sqrt{n}$.
- n פריק אז קיימים שלמים ab כך ש- $n = ab$, כאשר $1 < a, b < n$.
- נניח כי $1 < a \leqslant b < n$.
- על פי ההנחה שלנו, אם a מחלק n אז $a > \sqrt{n}$. בנוסף: $b \geqslant a > \sqrt{n}$.
- לכן

$$n = ab > \left(\sqrt{n}\right)\left(\sqrt{n}\right) = n ,$$

ז"א $n > n$ וזו סתירה!

■

על סמך המשפט הזה נבנה אלגוריתם $PRIME$ המכריע את L_{prime} בזמן פולינומיאלי. הרעיון של האלגוריתם הוא, בהינתן קלט n, לבדוק אם קיים מחלק של n אשר קטן מ- או שווה ל- $\sqrt{n}$. אם כן אז n פריק ואם לא אז n ראשוני.

<u>בניית האלגוריתם</u>

$PRIME =$ על קלט x:

1) בודק אם $x = \langle n \rangle$ ו- n מספר שלם חיובי.

- אם לא $\Leftarrow$ דוחה.

2) אם $n = 1$ $\Leftarrow$ דוחה.

3) אם $n = 2$ $\Leftarrow$ מקבל.

4) אחרת לכל $3 \leqslant d \leqslant \lfloor \sqrt{n} \rfloor$:

- מחשב $n \bmod d$.
- $n \bmod d = 0$ $\Leftarrow$ מקבל.

5) מקבל.

<u>סיבוכיות זמן</u>

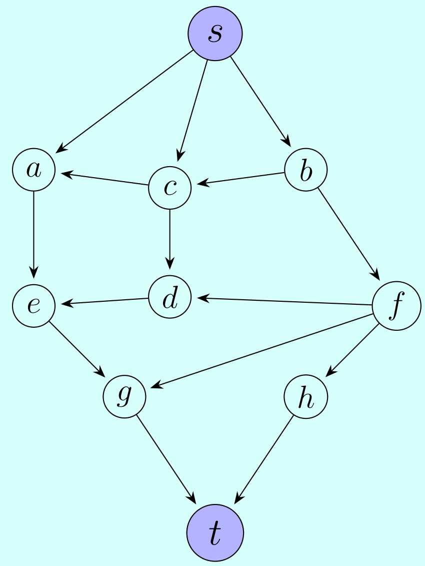
- $PRIME$ מבצע $O(n)$ איטרציות.
- בכל איטרציה $PRIME$ מחשב $n \bmod d$ באמצעות האלגוריתם $DIVISION$, אשר הוכחנו בדוגמה 9.8 שהוא רץ בזמן $O(n^2)$.
- לכן הסיבוכיות זמן של $PRIME$ היא $O(n^3)$

לפיכך:

$$L_{\text{prime}} \in P \ .$$

9.5 בעיית PATH

הגדרה 9.7 בעיית המסלול בגרף מכוון



<u>קלט:</u> גרף מכוון $G=(V,E)$ ושני קודקודים $s,t\in V$.

<u>פלט:</u> 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 הבעייה $PATH$.

$A=$ על קלט $\langle G,s,t\rangle$:

(1 צובע את s.

(2 מבצע $|V|-1$ פעמים:

- לכל צלע $(u,\mathrm{v})\in E$:
  - אם u צבוע $\Leftarrow$ צבע את v.

(3
- אם t צבוע $\Leftarrow$ החזיר "כן".
- אחרת $\Leftarrow$ החזיר "לא".

זמן ריצה של האלגוריתם הוא $O\left(|V|\cdot|E|\right)$ פולינומיאלי במספר הקודקודים $|V|$.

האם זה פולינומיאלי בגודל הקלט $|\langle G\rangle|$?

איך נקודד את G?

- נניח כי $|V|=n$ ו- $V=\{1,2,3,\quad\cdots\quad,n\}$.

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

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

- נניך כי מספרים מקודדים בבסיס ביניארי.
- אזי גודל הקידוד של G שווה $n^2 + n \log_2 n$, כלומר

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

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

ולכן A רץ בזמן פולינומיאלי בגודל הקלט. ■

9.6 הבעיית $RELPRIME$

הגדרה 9.8 מספרים זרים (Relatively prime)

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

הגדרה 9.9 בעיית $RELPRIME$

<u>קלט:</u> שני מספרים x ו- y.

<u>פלט:</u> האם x ו- y זרים?

$$RELPRIME = \left\{ \langle x, y \rangle \;\middle|\; \gcd(x,y) = 1 \right\} .$$

אנחנו נוכיח כי ניתן להכריע את $RELPRIME$ בזמן פולינומיאלי, כלומר נוכיח $RELPRIME \in P$ במשפט 9.8 למטה. לפניכן נסביר את האלגוריתם של אוקלידס למציאת ה- gcd של שני שלמים, ומתוך זה נוכל לחשב את הסיבוכיות זמן של $RELPRIME$. ראשית נזכיר משפט שלמדנו בקורסים קודמים:

משפט 9.6 השמפט 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: ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטים שימושיים" בדף 110. ■

האלגוריתם של אוקלידס הוא אלגוריתם, שמקבל כקלט שני מספרים x, y ופולט את $\gcd(x,y)$. הוא מתבוסס על המשפט 9.6. האלגוריתם עצמו הוא כדלקמן:

$EUCLID =$ על קלט x, y:

(1) כל עוד $y \neq 0$:

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

(3) $\text{swap}(x, y)$

(כלומר מחליפים בין x ו- y).

(4) מחזירים את x.

לדוגמה:

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

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

משפט 9.7 (משפט עזר)

אם y $x > y$ אז $x \bmod y < \frac{x}{2}$.

הוכחה: ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטים שימושיים" בדף 111. ■

משפט 9.8

$$RELPRIME \in P \ .$$

הוכחה:
נבנה אלגוריתם A המכריע את $RELPRIME$ בזמן פולינומיאלי. $RELPRIME$ היא השפה של הבעיה, שמקבלת כקלט שני מספרים שלמים x, y ומחזירה תשובה לשאלה, האם x, y זרים. כלומר:

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

לכן A משתמש בהאלגוריתם של אוקלידס $EUCLID(x, y)$ כדי לחשב $\gcd(x, y)$.

<u>**בניית האלגוריתם A המכריע $RELPRIME$:**</u>

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

- אם $EUCLID(x, y)$ מחזיר $\gcd(x, y) = 1$ אז A מקבל.
- אחרת A דוחה."

<u>**הוכחת הנכונות**</u>

הנכונות של A מנובעת ישר מהנכונות של האלגוריתם האוקלדס, $EUCLID$.

<u>**סיבוכיות זמן**</u>

נראה כי A רץ בזמן פולינומיאלי בגודל הקלט $\langle x, y\rangle$. נסמן את אורך הקלט $n = |\langle x, y\rangle|$.

- לפי משפט עזר 9.7: $x \bmod y < \frac{x}{2}$.
- בכל איטצרציה, בשלב (2) המשתנה x מקבל את הערך החדש $x \leftarrow x \bmod y$.
ניתן לחשב את $x \bmod y$ בעזרת האלגוריתם החילוק של אוקלידס $DIVISION$ שרץ בזמן $O(n^2)$ (ראו דוגמה 9.8).
- לכן בכל איצטרציה הערך החדש של x קטן ממש מחצי של הערך הקודם של x.

- לכן אחרי כל איטרציה, x קטן בלפחות חצי.
- בשלב (3), A מחליף בין x ו- y, אז אחרי כל 2 איטרציות, גם x קטן בלפחות חצי וגם y קטן בלפחות חצי.
- לכן המספר הפעמים המקסימלי שאפשר לבצע שלבים (2) ו- (3) היא $\min\left(\ 2\lfloor\log_2 x\rfloor\ ,\ 2\lfloor\log_2 y\rfloor\ \right) \triangleq m$.
- לכן m הוא חסם עליון של מספר האיטרציות ש $EUCLID$ מבצע.
- $n \leqslant m$ כאשר n הוא האורך של הקלט $\langle x,y\rangle$ לכן $EUCLID$ מבצע $O(n)$ איטרציות.
- כל איטרציה מבצעת $DIVISION$ כדי לחשב $x \bmod y$ אשר רץ בזמן $O\left(n^2\right)$.

$\Leftarrow$ A רץ בזמן $O\left(n^3\right)$. כלומר:

$$RELPRIME \in TIME\left(n^3\right)\ .$$

לכן:

$$RELPRIME \in P\ .$$

■

9.7 *הוכחות של משפטים שימושיים

משפט 9.9 השמפט 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 הזו היא רק להוסיף הבנה להוכחה של משפט 9.8 לסיבוכיות זמן של $RELPRIME$ למטה.
היא לא הוכחה שאתם תיבחנו עליה ואפשר לדלג עליה.
נתחיל אם משפט החילוק של אוקלידס, שאומר שאם x,y שלמים אז קיימים שלמים q ו- $0\leqslant r<y$ כך ש:

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

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

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

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

$$d\mid\gcd\left(y,x \bmod y\right)\ . \qquad (2*)$$

כעת נגדיר $\bar{d}\triangleq\gcd(y,y \bmod x)$.
מכיוון ש- $\bar{d}$ הוא מחלק משותף של y ו- $y \bmod x$ אז $\bar{d}\mid y$ וגם $\bar{d}\mid x \bmod y$. לכן בזכות משוואה (1*):

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

ז"א $y \mid \bar{d}$ וגם $x \mid \bar{d}$ אז בהכרח:

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

לסיכום, לפי משוואות (2*) ו- (3*):

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

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

משפט 9.10 (משפט עזר)

אם y $x >$ אז $x \bmod y < \frac{x}{2}$.

הוכחה: יש שני מקרים:

1) $y \leqslant \frac{x}{2}$

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

נוכיח את הטענה עבור שני המקרים.

מקרה 1: $y \leqslant \frac{x}{2}$.

לפי משפט החילוק של אוקלידס אם x,y שלמים עבור $x > y$ אז קיימים $q = \left\lfloor \frac{x}{y} \right\rfloor$ ו- $r = x \bmod y$ כך ש $0 \leqslant r < y$ ו-

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

בפרט $r < y$ וגם $y \leqslant \frac{x}{2}$ לפיכך $x \bmod y < y \leqslant \frac{x}{2}$.

מקרה 2: $y > \frac{x}{2}$.

לפי משפט החילוק של אוקלידס אם x,y שלמים עבור $x > y$ אז קיימים שלם $q = \left\lfloor \frac{x}{y} \right\rfloor$ ושלם $r = x \bmod y$ כך ש $0 \leqslant r < y$ ו-

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

בפרט אם $y > \frac{x}{2}$ אז $x < 2y$. אז בהכרח $q < 2$. מכיוון ש- $x > y$ ו- $q = \left\lfloor \frac{x}{y} \right\rfloor$ אז הערך המינימלי של q הוא $q = 1$. לכן אם $q < 2$ בהכרח $q = 1$. לכן יש לנו

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

מכאן

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

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

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

■