

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**.

אתה מהבעיות שבהן 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

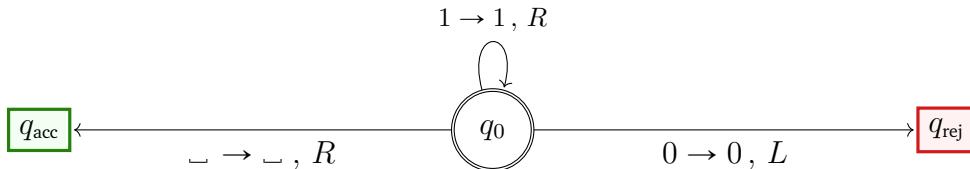
נתבונן על השפה של מחרוזות האוניריות הבאה:

$$L = \{1^n \mid n \geq 0\}.$$

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 :

- (1) • אם המילה היא מילת הריקה מקבל.
- אחרת ממשיכה לשלב (2).
- (2) • אם התו הנזכר 0 תדחה.
- אחרת אם התו הנזכר 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 לפחות 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 פחותה מ- $|w|$ צעדי חישוב, המשמעות היא המכונה כלול לא קראה את כל הקלט, וזה אינו נקרה נפוץ במיוחד (אם כי הוא בח初恋 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 $L = \{a^n b^n \mid n \geq 0\}$ ואז נחשב את הסיבוכיות זמן שלה.

בנייה המכונת טיורינג

"על קלט w :

- (1) אם התו הנזכר הוא 1 $\Leftarrow M$ מקבל.

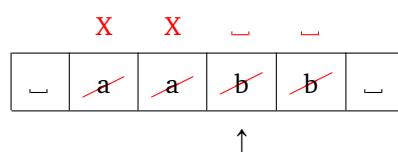
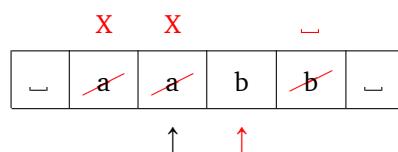
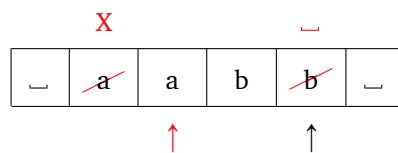
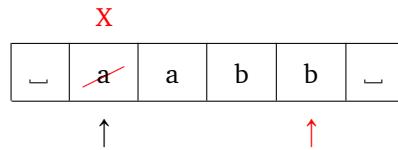
(2) אחרת אם הtau הנקרא הוא $M \Leftarrow b$ דוחה.

(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 על קלט: M' $O(n)$ (1) מעתיקה את ה- b -ים לסדרת 2 (ותוך כדי לבדוק האם w מהצורה a^*b^*).(2) מוחק את ה- b -ים מסרט 1. $O(n)$

(3) מזיהה את הראשים לתחילת הסרטיים.

(4) אם שני הראשים מצביעים על $_ \leftarrow$ מקבלת.(5) אם אחד הראשים מצביע על $_ \leftarrow$ והשני לא \leftarrow תדחה. $O(n)$ שלבים (3-5):

(6) מזיהה את שני הראשים ימינה וחזרת לשלב (3).

סדרת (1)



סדרת (2)

סיבוכיות זמןנסמן את אורך הקלט ב- $n = |w|$ זמן הריצה של M' הוא $O(n)$ **הגדרה 9.3** $TIME(f(n))$

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

$$TIME(f(n)) = \{L \mid O(f(n)) \text{ זמן } L \text{ בזמן } f(n)\}.$$

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 זו ראיינו בדוגמה 1.5. נבנה מכונת טיריניג M עם סרט ייחיד שמכריעה את השפה $L = \{a^{2^n} \mid n \geq 0\}$ ואז נחשב את הסיבוכיות זמן שלה.

הרעיווןהמכונה תסרק את הסרט שלה משמאל לימין ובכל איטרציה תמחק חצי מה- a -ים שנשארו.

בנייה המכונה

" על קלט $w = M$

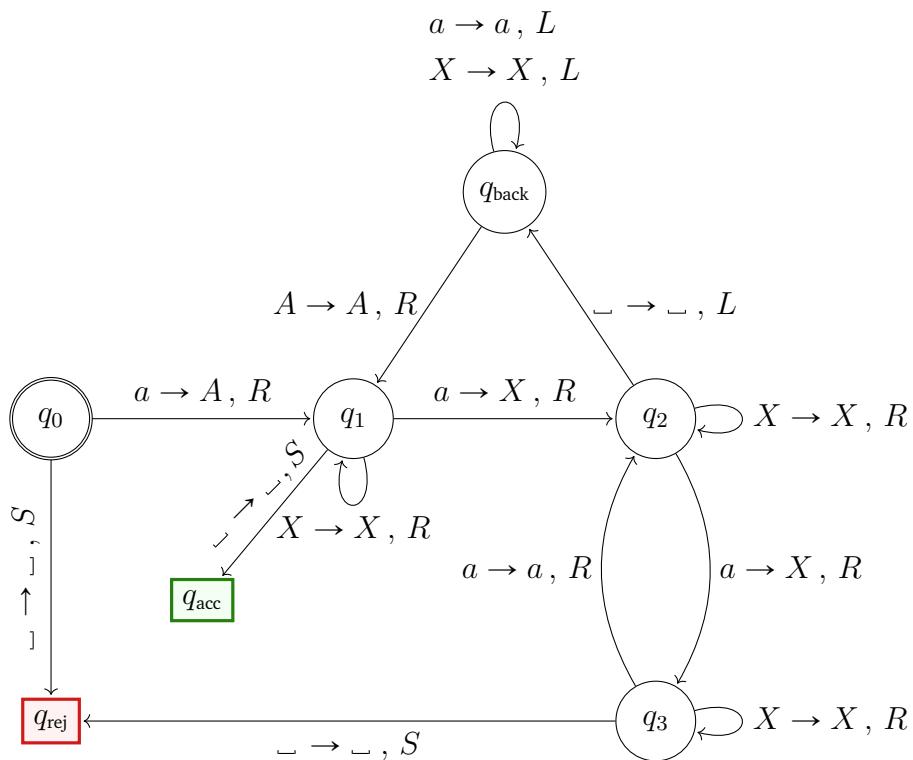
(1) אם הטו הנקרה הוא $_ \Leftarrow$ דוחה.

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

(3) סורקת את הסרט משמאלה לימין ומוחקמת כל a שני:

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

התרשים מציבים של המבנה מתואר באירור למטה:

סיבוכיות זמן

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

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

דוגמה 9.7 חישור ביןארי

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

בנייה המכונה

$\text{SUBTRACT} = \langle \text{על קלט } y, x \rangle$ כאשר y, x שלמים בבסיס ביןארי ו- $y > x$:

1) רושמת את x בסרט 1 ואת y בסרט 2 משמאלי לימין כך שהתאים עם הקצוות הימניות של x ו- y מיושרים זה מתחת זה. בתחילת הסרט 3 ריק.

2) מעמידה את ראש הסרטים 3, 2, 1 על התאים המיושרים של הקצוות הימניים של הקלטים. בפרט, ראש הסרטים 1 ו-2 נמצאים על הספרות הפחות משמעותית של x ו- y , וראש הסרט 3 נמצא בתא שמתוחתם.

3) שלב זה מבצע חישור ללא חוב.

יהי $\sigma_1, \sigma_2 \in \{0, 1\}$ תו הנקרה הסרט 1 והוא $\sigma_2 \in \{0, 1\}$ תו הנקרה הסרט 2.

- אם $(\sigma_1, \sigma_2) = (_, _)$ אז M מקבלת.
- אם $(0, 0, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (0, 0)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(1, 0, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (1, 0)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(1, 1, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (1, 1)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(_, 0, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (_, 0)$ מזיהה את הראשים (S, L, L) וחזרת על שלב זה.
- אם $(_, 1, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (_, 1)$ מזיהה את הראשים (S, L, L) וועברת לשלב הבא.
- אם $(0, _, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (0, _)$ מזיהה את הראשים (L, S, L) וחזרת על שלב זה.
- אם $(1, _, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (1, _)$ מזיהה את הראשים (L, S, L) וחזרת על שלב זה.
- אם $(0, 1, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (0, 1)$ מזיהה את הראשים (L, L, L) וועברת לשלב הבא.

4) שלב זה מבצע חישור כאשר קיים חוב.

- אם $(1, 0, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (1, 0)$ מזיהה את הראשים (L, L, L) וועברת לשלב הקודם.
- אם $(0, 1, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (0, 1)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(1, 1, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (1, 1)$ מזיהה את הראשים (L, L, L) וחזרת על שלב זה.
- אם $(0, 0, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (0, 0)$ מזיהה את הראשים (S, L, L) וחזרת על שלב זה.
- אם $(_, 0, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (_, 0)$ מזיהה את הראשים (S, L, L) וחזרת על שלב זה.
- אם $(_, 1, 0), \text{כותבת}, (\sigma_1, \sigma_2) = (_, 1)$ מזיהה את הראשים (S, L, L) וחזרת על שלב זה.
- אם $(0, _, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (0, _)$ מזיהה את הראשים (L, S, L) וחזרת על שלב זה.
- אם $(1, _, 1), \text{כותבת}, (\sigma_1, \sigma_2) = (1, _)$ מזיהה את הראשים (L, S, L) וועברת לשלב הקודם.

סיבוכיות זמן

יהי $|x|, |y| = n$. המכונה SUBTRACT את השני הסרטים 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 $O(n)$ (lienari).

דוגמה 9.8 האלגוריתם החילוק של אוקלידס

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

$$x = qy + r , \quad 0 \leq r < |y| .$$

במקרה הפרטי ש- $x < y < 0$ אז:

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

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

בנייה האלגוריתם

: $0 < y < x$ כאשר y, x שלמים בבסיס בינארי ו- $x = DIVISION$

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

(2) כל עוד ש- $r \geq y$

$$r \leftarrow r - y \quad (3)$$

$$q \leftarrow q + 1 \quad (4)$$

(5) פלט: q, r .

סיבוכיות זמן

נסמן $| \langle x, y \rangle |$ אורץ הקלט.

• מבצע r איטרציות לכל היותר.

• $y < r$ 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

$$O(n^2) .$$

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

משפט 9.1

לכל מכונת טיורינג מרובה סרטים M הרצה בזמן $t(n)$ קיימת מכונת טיורינג סרט יחיד M' השköלה ל- M ורצה בזמן $O(t^2(n))$.

הוכחה:

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

نبנה מכונת טיורינג S עם סרט אחד שרצה בזמן $O(t^2(n))$.

- ראשית נרשום את התוכן של k הסרטים 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:

המכונת טיורינג M

0	1	0	1	0	-	-
↑						

a	a	a	-	-	-	-
↑						

a	b	a	b	-	-	-	-
↑							

המכונת טיורינג S

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

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

בנויות המכונה S

- 1) תחילת S מתחילה את הסרט שלה בלבוט את התכנים של k הסרטים על הסרט היחיד שלה, עם תו '#' להפריד בין סרטים שונים של M .
- 2) בנוסף S רושמת תו '#' בקצה השמאלי כדי לסמנו את התחלת הקלט ותו '#' בקצת הימין כדי לסמנו את סוף הקלט.

3) ב כדי לסייע צעד חישוב אחד של M בהמוניה S , המכונה S سورקת את הشرط מ- # הראשון ל- # האחרון. בສירקה זו S זכרת את המיקומים של ה- k ראשים על פי התאים שמסומנים עם חצים, באמצעות k תא זכרון.

4) אחר כך S מבצעת סירקה שנייה של הشرط. בסירקה זו, לפי הפונקציית המעברים של M , המכונה S מבצעת, לכל $1 \leq i \leq k$

- כתיבה של הסימן החדש בסרטה ה- i במקומות הנוכחי של הראש של סרט ה- i .
- תזוזה של הראש של סרט ה- i .

5) במצב שהראש של אחד הסרטים של M זו ימינה לתו רוח מצד ימין של סוף הקלט, אז S תוסיף תא אחד עםתו רוח מצד שמאל של ה- # המפריד בין סרטים, ולאחר כך היא תזוז את כל התאים שבצד ימין של התא המוסף מקום אחד ימינה.

סיבוכיות זמן של המכונה

יהי n האורך של התוכן הכי ארוך מthan 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 $O(t(n))$.

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

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

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

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

\Leftarrow הסיבוכיות זמן של S היא:

$$O(t(n)) O(t(n)) = O(t^2(n))$$

9.3 יחס בין הסיבוכיות של מכונת טיריניג דטרמיניסטיבית ומכונת טיריניג אי דטרמיניסטיבית

משפט 9.2

לכל מכונת טיריניג אי-דטרמיניסטיבית N הרצה בזמן $f(n)$, קיימת מכונת טיריניג דטרמיניסטיבית D השකולה לה- N ורצה בזמן $2^{f(n)}$.

הוכחה:
בhinתן מכונת טיריניג אי-דטרמיניסטיבית N הרצה בזמן $f(n)$ מכונת טיריניג דטרמיניסטיבית D באותו אופן כמו בהוכחת השקילות במשפט 4.1.

כלומר, בהינתן קלט w , D תסrox את עץ החישוב של N ו- w לרוחב 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 $f(n)$.
- 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

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

ולכן זמן הריצה של D חסום ע"י

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

נתיחס כאן לשני החסמים הבאים:



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

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

9.4 המחלקה P

הגדרה 9.4 בעיית הכרעה

בעיית הכרעה מוגדרת באופן הבא:

"בhinintן קלט כלשהו, האם הקלט מקיים תנאי מסוים"

דוגמה 9.9

בhinintן מספר n , האם n ראשוני?

כל בעיית הכרעה ניתנת לתאר כשפה שקולה:

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

משפט 9.3

. שפה \equiv בעיית הכרעה .

הגדרה 9.5 אלגוריתם זמן פולינומיAli

אומרים כי אלגוריתם A מכריעה בעיה בזמן פולינומיAli אם קיימים קבוע $0 < c$ 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 המכריע בעיה בזמן פולינומי, אז קיימת מכונת טיורינג דטרמיניסטיבית המכריע את השפה השקולה לבעיה זו בזמן פולינומי.

. מכונת טיורינג ≡ אלגוריתם מכריע .

הגדרה 9.6 המחלקה P

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

דוגמה 9.10

בדוגמה 9.2 הוכחנו כי קיימת מכונת טיורינג דטרמיניסטיבית שמכריעת עת השפה

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

בזמן $O(n^2)$. לכן $L \in P$.

דוגמה 9.11

תהי L_{prime} השפה הבאה:

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

הוכיחו: $L_{\text{prime}} \in P$

פתרונות:

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

האלגוריתם עצמו עוזה שימוש של המשפט הבא:

משפט: למספר פריך יש מחלק הקטן מ- או שווה לשורשו

יהי n מספרשלם. אם n פריך אז קיים שלם $a \leq \sqrt{n}$ שמחלק את n .

הוכחה:

- נניח בsvilleה כי n פריך ולא קיים מחלק d של n כך ש- $\sqrt{n} < d \leq n$.
- n פריך אז קיימים שלמים a, b כך ש- $n = ab$, כאשר $1 < a, b < n$.
- נניח כי $n < a \leq b$.

על פי ההנחה שלנו, אם a מחלק n אז $\sqrt{n} > a$. בנוסף:

• לכן

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

ז"א $n > n$ וזה סתירה!

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

בנייה האלגוריתם

: $x = PRIME$ על קלט

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

- אם לא \Leftarrow דוחה.

2) אם $n = 1 \Leftarrow$ דוחה.

3) אם $n = 2 \Leftarrow$ מקבל.

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

- מחשב $n \bmod d$

$\Leftarrow n \bmod d = 0$ • מקבל.

5) מקבל.

סיבוכיות זמן

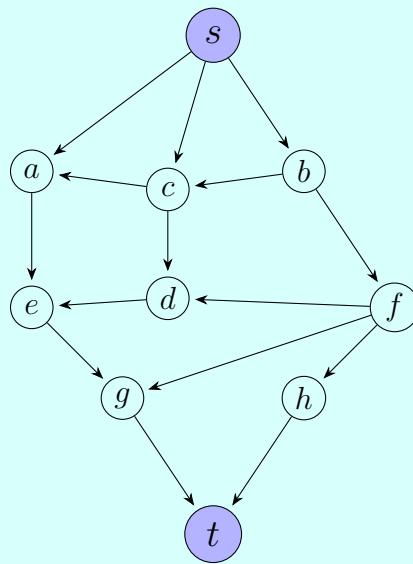
- $PRIME$ מבצע $O(n)$ איטרציות.
- בכל איטרציה $PRIME$ מחשב $n \bmod d$ באמצעות האלגוריתם $DIVISION$, אשר הוכחנו בדוגמה 9.8 שהוא רץ בזמן $O(n^2)$.
- לכן הסיבוכיות זמן של $PRIME$ היא $O(n^3)$.

לפיכך:

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

9.5 בעיית PATH

הגדרה 9.7 בעיית המסלול בגרף מכוון



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

פלט: האם קיים מסלול ב- G מ- s ל- t ?

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

решת

$$PATH \in P.$$

הוכחה: נבנה אלגוריתם A שמכריע את $PATH$ בזמן פולינומיAli.

בנייה האלגוריתם

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

(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 מקבל.
- אחרת \Leftarrow דוחה.

הוכחת הנכונות

הוכחה של הכוון \Leftarrow

: $x \in PATH$ אם

. t גרף מכוון וגם $s, t \in V$ כך קיימים מסלול מ- s ל- t .

\Leftarrow קיימים קבוצת צלעות

$$C = \{u_1u_2, u_2u_3, \dots, u_{k-2}u_{k-1}, u_{k-1}u_k\} \subseteq E,$$

כasher $.k \leq |V| - 1$ $u_1 = s$, $u_k = t$

\Leftarrow לכל $1 \leq i \leq k - 1$, באיטרציה i הקודקוד u_{i+1} יצובע.

\Leftarrow מכיוון ש- $|V| \leq k$ איזי עד סוף הלולאה הקודקוד $u_k = t$ יצובע.

\Leftarrow מכיוון שקודקוד t צובע אז A מקבל.

הוכחה של הטעון \Rightarrow

: $x \notin PATH$ \Leftarrow 2 מקרים:

מקרה 1 $x \neq \langle G, s, t \rangle$ ואז דוחה.

מקרה 2 אחרית $x = \langle G, s, t \rangle$ כאשר $G = (V, E)$ גרף מכוון וגם $s, t \in V$ כך שלא קיימים מסלול מ- s ל- t .

לאחר האיטרציה ה- i כל קודקוד שצובע הוא קודקוד שנייתן להגעה אליו מ- s ע"י מסלול מכוון של אורך i לכל היותר.

לא ניתן להגעה לקודקוד ע"י אף מסלול בגרף G .

\Leftarrow לא קיימים סדרת קודקודיים $t \rightarrow \dots \rightarrow u_2 \rightarrow u_3 \rightarrow \dots \rightarrow s$ כך שבסוף הלולאה t יהיה צובע. דוחה. $A \Leftarrow$

סיבוכיות זמן

- נסמן אורך הקלט $.N = \langle G, s, t \rangle$
- נסמן $m = |E|$ ו- $n = |V|$
- מבצע לולאה לכל $1 \leq i \leq |V| - 1$ שדורש $O(|V|)$ צעדים.
- בתוך הלולא מבצע לולאה מעל צלעות עברון הנקודה ההתחלה צובעת. \Leftarrow שלב זה דורש $O(|E|)$ צעדים.
- סבב"כ הסיבוכיות זמן של A היא $O(|E| \cdot |V|) = O(nm)$
- בנסוף: $n \leq N$ וגם $m \leq N$ לכן A דורש N^2 צעדי חישוב לכל היותר.
- לכן A רץ בזמן $O(N^2)$

■ $.PATH \in P \Leftarrow PATH \in TIME(N^2) \Leftarrow O(N^2)$ בזמן $PATH$

9.6 הבעייה 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

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 פולינומי, כלומר נוכיח $\text{RELPRIME} \in P$ 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 שלמדו בקורסים קודמים:

משפט 9.6 המשפט של האלגוריתם של אוקלידס

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

אם x, y שלמים אז

■ **הוכחה:** ההוכחה היא לא חלק של הקורס ומופיע בסעיף האחרון "הוכחות של משפטי שימושיים" בדף 110. המשפט של אוקלידס הוא אלגוריתם, שמקבל כקלט שני מספרים y, x וpollut את $\gcd(x, y)$. הוא מתבסס על המשפט 9.6. האלגוריתם עצמו הוא כדלקמן:

$x, y = \text{על קלט}$

(1) כל עוד $y \neq 0$:

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

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

(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 $\text{RELPRIME} \in P$ נctrיך למשפט עזר הבא:

משפט 9.7 (משפט עזר)

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

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

משפט 9.8

$$RELPRIME \in P .$$

הוכחה:

نبנה אלגוריתם A המכירע את $RELPRIME$ בזמן פולינומיאי. $RELPRIME$ היא השפה של הבעיה, שמקבלת כקלט שני מספרים שלמים x, y ומחייבת תשובה לשאלת, האם y זרימ. ככלומר:

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

לכן A משתמש באלגוריתם של אוקלידס ($EUCLID(x, y)$) כדי לחשב בנייה האלגוריתם A המכירע:

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

- אם $\gcd(x, y) = 1$ אז A מקבל.

- אחרת A דוחה."

הוכחת הנכונות

הנכונות של A מנובעת ישיר מהנכונות של האלגוריתם האוקלידס, $EUCLID$.

סיבוכיות זמן

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

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

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

ניתן לחשב את $y \bmod x$ בעורת האלגוריתם החילוק של אוקלידס $DIVISION$ שרצה בזמן $O(n^2)$ (ראו דוגמה 9.8).

לכן בכל איטרציה הערך החדש של x קטן ממחצית הערך הקודם של x .

לכן אחרי כל איטרציה, x קטן לפחות חצי.

בשלב (3), A מחלייף בין x ו- y , או אחריו כל 2 איטרציות, גם x קטן לפחות חצי וגם y קטן לפחות חצי.

לכן המספר הפעמים המקסימלי שאפשר לבצע שלבים (2) ו- (3) היא $m \triangleq \min(2 \lfloor \log_2 x \rfloor, 2 \lfloor \log_2 y \rfloor)$.

לכן m הוא חסם עליון של מספר האיטרציות ש- $EUCLID$ מבצע.

אם $n \leq m$ הוא האורך של הקלט $\langle x, y \rangle$ לנכון $EUCLID$ מבצע $O(n)$ איטרציות.

כל איטרציה מבצעת $DIVISION$ כדי לחשב $x \bmod y$ אשר רץ בזמן $O(n^2)$.

רץ בזמן $O(n^3)$. ככלומר:

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

לכן:

$$RELPRIME \in P .$$



9.7 *הוכחות של משפטי שימושיים

משפט 9.9 המשפט של האלגוריתם של אוקלידסאם 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* למטרה. היא לא הוכחה שאותם תיבחנו עלייה ואפשר לדלג עליה.

נתחיל אם משפט החלוק של אוקלידס, שאומר שאם x, y שלמים אז קיימים שלמים q ו- $r \leq 0$ 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- $d \mid y$ וגם $d \mid (x \bmod y)$. לכן בaczות מסוואה (1*):

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

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 בaczות מסוואה (1*):

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

וז"א וגם $\bar{d} \mid x$ אז בהכרח:

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

לסיכום, לפי משוואות (2*) ו- (3*):
 $d \mid \bar{d} \wedge \bar{d} \mid d$.

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

משפט 9.10 (משפט עזר)אם $x \bmod y < \frac{x}{2}$ אז $x > y$.**הוכחה:** יש שני מקרים:

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

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

נווכיח את הטענה עבור שני המקרים.

מקרה 1: $y \leq \frac{x}{2}$

לפי משפט החלוק של אוקלידס אם x, y שלמים Überom $x > y$ אז קיימים $q = x \text{ mod } y$ ו- $r = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \text{ mod } y)$.

$$\text{בפרט } r < y \text{ ו- } \frac{x}{2} \leq y \text{ לפיכך } x \mod y < y \text{ ו- } 0 \leq r < y.$$

מקרה 2: $y > \frac{x}{2}$

לפי משפט החלוק של אוקלידס אם x, y שלמים Überom $x > y$ אז קיימים $q = x \text{ mod } y$ ו- $r = qy + r = \left\lfloor \frac{x}{y} \right\rfloor y + (x \text{ mod } y)$.

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

$$x = qy + r = (1)(y) + ry + (x \text{ mod } y).$$

מכאן

$$x - y = x \text{ mod } y.$$

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

$$x \text{ mod } y < \frac{x}{2}.$$

