תרגילים 9: סיבוכיות

שאלה 1 תהי MOD השפה שמוגדרת: $MOD = \{\langle a,b,m\rangle \,|a \equiv b \pmod m \ , a,b,m \in \mathbb{N}\}$.
הוכיחו כי $MOD \in P$.

שאלה 2 תהי $MODEXP$ השפה שמוגדרת:

$$MODEXP = \left\{\langle a,b,c,p\rangle \,|a^b \equiv c \pmod p \ , a,b,c,p \in \mathbb{N}\right\} .$$

הוכיחו כי $MODEXP \in P$.

שאלה 3 נתונות שתי בעיות A ו- B מעל אותו אלפיביט Σ, שני אלגוריתמי אימות V_1 ו- V_2 עבור A ו- B (בהתאמה) הרצים בזמן פולינומיאלי.

א) בנו אלגוריתם אימות V עבור הבעיה $A \cup B$. תארו במילים את האלגוריתם והוכיחו את נכונה הבניה.

ב) הוכיחו כי אלגוריתם שבניתם בסעיף א' רץ בזמן פולינומיאלי.

שאלה 4 בעיית $PARTITION$ מוגדרת באופן הבא:
בהינתן קבוצת מספרים $A = \{a_1, a_2, \dots, a_n\}$, האם קיימת חלוקה של A לשתי קבוצות A_1 ו- A_2 כך ש-

- $A_1 \cap A_2 = \emptyset$
- $A_1 \cup A_2 = A$
- $\sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i = \frac{1}{2} \sum_{a_i \in A} a_i$.

בנו מכונת טיורינג אי-דטרמיניסטית המכריעה את $PARTITION$ בזמן פולינומיאלי.

שאלה 5 נתונה בעיה A ונתון אלגוריתם M_A המכריע עת A בזמן פולינומיאלי. נגדיר את הבעיה $B = \left\{ww \;\middle|\; w \in A\right\}$.

א) בנו אלגוריתם M_B המכריע את B. תארו במילים את האלגוריתם והוכיחו את נכונות הבניה.

ב) האם האלגוריתם שבניתם רץ בזמן פולינומיאלי? הסבירו.

שאלה 6 קבעו אם הטענה הבאה נכונה, לא נכונה או שקולה לשאלה פתוחה:
קיים אלגוריתם המקבל כקלט גרף לא מכוון G ומכריע בזמן פולינומיאלי האם G מכיל קבוצה בלתי תלויה בגודל 1000.

שאלה 7 בעיית $2COLOR$ מוגדרת באופן הבא:

בהינתן גרף (לא מכוון) $G=(V,E)$, האם קיימת פונקציית צביעה $c: V \to \{\text{red}, \text{blue}\}$ כך שלכל קשת $(u,w) \in E$ מתקיים $c(u) \neq c(w)$?

נגדיר: $2COLOR = \{\langle G \rangle \mid G \text{ ניתנת לצבייעה חוקית ב- } 2 \text{ צבעים} \}$.

הוכיחו: $2COLOR \in P$.

2

תשובות

שאלה 1

נבנה אלגוריתם אשר מכריע את MOD בזמן פולינומיאלי:

בניית האלגוריתם

$MODEQV =$ "על קלט x:

1) בודק אם $x = \langle a, b, m \rangle$ כאשר a, b, m טבעיים.
   - אם לא $\Leftarrow$ דוחה.
2) מחשב $x = |a - b|$.
3) מאתחל $r \leftarrow x$.
4) כל עוד ש- $r \geq m$:
   - $r \leftarrow r - m$
5)
   - אם $r = 0$ $\Leftarrow$ מקבל.
   - אחרת דוחה."

נכונות

הוכחה לכיוון $\Leftarrow$

אם $a \equiv b \pmod{m}$

$\Leftarrow$ $m \mid a - b$

$\Leftarrow$ $m \mid x$ כשאר $x = |a - b|$

$\Leftarrow$ קיים שלם q כך ש: $x = qm$

$\Leftarrow$ אחרי q איטרציות של הלולאה האלגוריתם יפלוט $r = 0$

$\Leftarrow$ $MODEQV$ יקבל.

הוכחה לכיוון $\Rightarrow$

אם $a \not\equiv b \pmod{m}$

$\Leftarrow$ $m \nmid a - b$

$\Leftarrow$ $m \nmid x$ כשאר $x = |a - b|$

3

$\Leftarrow$ לא קיים שלם q כך ש: $x = qm$

$\Leftarrow$ אחרי $\left\lfloor \frac{x}{m} \right\rfloor$ איטרציות הלולאה תסתיים ותפלוט $r \neq 0$

$\Leftarrow$ $MODEQV$ ידחה.

סיבוכיות זמן

- יהי $n = |\langle a,b,m \rangle|$ האורך שך הקלט.
- בשלב (1) $MODEQV$ סורק את הקלט ודורש $O(n)$ צעדים.
- הסיבוביות זמן של החיסור בשלב (2) הוא $O(n)$.
- הלולאה מתבצעת לכל היותר x פעמים. לכן יהיו $O(n)$ איטרציות.
- בכל איטרציה $MODEQV$ מחשב את החיסור $r-m$ אשר דורש לכל היותר $O(n)$ צעדי חישוב.
- לכן הלולאה מתבצעת בכל היותר $O\left(n^2\right)$ צעדים.

לכן $MODEQV$ עולה $O(n^2)$ צעדי חישוב לכל היותר.
לכן: $MOD \in TIME\left(n^2\right)$.
מצאנו אלגוריתם שמכריע MOD בזמן פולינומיאלי לפיכך

$$MOD \in P \ .$$

<u>שאלה 2</u> נבנה מכונת טיורינג M שמכריעה את $MODEXP$ בזמן פולינומיאלי.

הרעיון של האלגוריתם

ראשית נסביר את השיטה של האלגוריתם שמחשב חזקה מודולרית: $a^b \bmod p$.

נתונה חזקה מודולרית

$$a^b \bmod p$$

החזקה b מתפרקת לצירוף לינארי של חזקות של 2 באופן הבא:

$$b = b_k 2^k + b_{k-1}2^{k-1} + \ldots + b_1 2^1 + b_0 2^0 = b_k\beta^k + b_{k-1}\beta^{k-1} + \ldots + b_1\beta^1 + b_0\beta^0 \ ,$$

כאשר $\beta^i = 2^i$ וכל b_i שווה ל- 0 או 1. למעשה הפירוק הזה הוא הייצוג בינארי של b:

$$b = [b_k b_{k-1} \ldots b_1 b_0]_2$$

כאשר $[\ldots]_2$ מסמן ייצוג של מספר שלם בבסיס ביניארי. מכאן:

$$a^b \bmod p = a^{b_k\beta_k + b_{k-1}\beta_{k-1} + \ldots + b_1\beta_1 + b_0} \bmod p = a^{b_k\beta_k} a^{b_{k-1}\beta_{k-1}} \ldots a^{b_1\beta_1} a^{b_0} \bmod p$$

לפי תכונת הכפל של היחס מודולרי:

$$a^b \bmod p = \left(a^{b_k\beta_k} \bmod p\right)\left(a^{b_{k-1}\beta_{k-1}} \bmod p\right)\ldots\left(a^{b_1\beta_1} \bmod p\right)\left(a^{b_0} \bmod p\right) \bmod p \ .$$

כעת נרשום את הביטוי הזה בצורה הבאה:

$$a^b \bmod p = (x_k \bmod p)(x_{k-1} \bmod p)\ldots(x_1 \bmod p)(x_0 \bmod p) \bmod p \ . \qquad (1*)$$

כאשר
$$\forall 0 \le i \le k \qquad x_i = \begin{cases} a^{2^i} & : b_i = 1 \\ 1 & : b_i = 0 \end{cases} .$$
אפילו יותר מזה, מכיוון ש- $2^i = \left(2^{i-1}\right)^2$ אז אפשר לגזור את היחס רקורסיבי הבא. אם נגדיר
$$z_i = a^{2^i} \bmod p$$
אזי
$$z_i = \left(a^{2^{i-1}}\right)^2 \bmod p = \left(a^{2^{i-1}} \bmod p\right)\left(a^{2^{i-1}} \bmod p\right) \bmod p = z_{i-1}^2 \bmod p \ .$$
שימו לב:
$$x_i = \begin{cases} z_i & : b_i = 1 \ , \\ 1 & : b_i = 0 \ . \end{cases} \qquad (*2)$$
האלגוריתם עצמו שמחשב את $a^b \bmod p$ עושה שימוש של התכונה (2*) משוואה (1*).

<u>**בניית האלגוריתם**</u>

$M=$ "על הקלט $\langle a,b,c,p\rangle$, כאשר a,b,c,p שלמים בייצוג בינארי:

1) מאתחלת:

- $z \leftarrow a \bmod p$
- $x \leftarrow 1$.
- $i \leftarrow 0$.

2) אם הייצוג בינארי של b הוא $b = b_k b_{n-1} \dots b_1 b_0$ אז לכל $0 \le i \le k$:

- אם $b_i = 1$:
  $x \leftarrow x \cdot z \bmod p$.
- $z \leftarrow z^2 \bmod p$.
- $i \leftarrow i+1$.

3)
- אם $x \equiv c \bmod p \Leftarrow$ M מקבלת.
- אחרת M דוחה. "

למטה האלגורים הזה רשום בסימון של אלגוריתמים.

Algorithm 1 MODEXP

Input: Integers a, b, c, p in binary representation.
$z \leftarrow a \bmod p$.
$x \leftarrow 1$
$i \leftarrow 0$
while $i \leq k$ **do**
  if $b_i = 1$ **then**
    $x \leftarrow x \cdot z \bmod p$
  end if
  $z \leftarrow z^2 \bmod p$
  $i \leftarrow i + 1$
end while
if $x \equiv c \bmod p$ **then**
  accept
else
  reject
end if

<u>סיבוכיות זמן</u>

- נסמן אורך הקלט: $n = |\langle a, b, c, p\rangle|$.
- בשלב (1) האתחול $x \leftarrow 1$ עולה $O(1)$ וגם $i \leftarrow 0$ עולה $O(1)$.
- השמה $z \leftarrow a \bmod p$ דורש חישוב של $a \bmod p$ בעזרת האלגוריתם $DIVISION$ שראינו בכיתה ע"י האלגוריתם החילוק של אוקלידס, שעולה $O(n^2)$.
- הלולאה מבצעת k איטרציות, כאשר k הוא המספר ספרות בייצוג בינארי של החזקה b. מכיוון ש- $b \leq n$ אז הלולאה מבצעת $O(n)$ איטרציות לכל היותר.
- בכל איטרציה M מחשבת $z^2 \bmod p$, ולפעמים $xz \bmod p$, בעזרת האלגוריתם החילוק של אוקלידס ע"י האלגורים $DIVISION$ שעולה $O(n^2)$.
- לבסוף M בודקת אם $x \equiv c \pmod{p}$ בעזרת האלגוריתם $MODEQV$ מסעיף א) שעולה $O(n^2)$ צעדי חישוב לכל היותר.
- לפיכך M מבצעת לכל היותר
$$O(1) + O(1) + O(n) + O(n)\, O\left(n^2\right) + O\left(n^2\right) = O\left(n^3\right) \ .$$

מכאן המכונה מכריעה את $MODEXP$ בזמן $O(n^3)$. לכן:
$$MODEXP \in TIME\left(n^3\right) \quad \Rightarrow \quad MODEXP \in P \ .$$

6

שאלה 3

א) הרעיון:

V מקבל בקלט זוג (w,y) ורוצה לבדוק האם y הוא עדות לזה ש- $w \in A \cup B$.

לצורך זה V מריץ את V_1 על הזוג (w,y).
אם V_1 קיבל אזי V מקבל.
אחרת, V מריץ את V_2 על הזוג (w,y) ועונה כמוה.

האלגוריתם

$V =$ על קלט (w,y):

1) מריץ את V_1 על (w,y).

- אם V_1 מקבל $\Leftarrow$ V מקבל.
- אם V_1 דוחה $\Leftarrow$ V מריץ את V_2 על (w,y) ועונה כמוה.

נכונות

אם $w \in A \cup B$

$\Leftarrow$ $w \in A$ או $w \in B$

$\Leftarrow$ קיימת עדות y כך ש- V_1 מקבל את הזוג (w,y) או V_2 מקבל את הזוג (w,y).

$\Leftarrow$ קיימת עדות y כך ש- V מקבל את הזוג (w,y).

אם $w \notin A \cup B$

$\Leftarrow$ $w \notin A$ וגם $w \notin B$

$\Leftarrow$ לכל עדות y, V_1 דוחה את הזוג (w,y) וגם V_2 דוחה את הזוג (w,y).

$\Leftarrow$ לכל עדות y, V דוחה את הזוג (w,y).

ב) נסמן p_1 הפולינום של V_1.
נסמן p_2 הפולינום של V_2.

אזי זמן הריצה של V חסום על ידי $O\left(p_1\left(|w|\right) + p_2\left(|w|\right)\right)$ ולכן V פולינומיאלי בגודל $|w|$.

שאלה 4

נבנה מ"ט א"ד M המכריעה את $PARTITION$ בזמן פולינומיאלי.

$M =$ על קלט $\langle A \rangle$:

7

1) בוחרת באופן א"ד תת-קבוצות A_1 של A.

2) בודקת האם סכום האיברים של A_1 שווה חצי מסכום האיברים של A.

- אם כן $\Leftarrow$ מקבלת.
- אם לא $\Leftarrow$ דוחה.

<u>נכונות הבנייה</u>

אם $\langle A\rangle \in PARTITION$

$\Leftarrow$ קיימת חלוקה של A ל- A_1 ו- A_2 כך ש- $\sum_{a_i\in A_1} a_i = \sum_{a_i\in A_2} a_i = \frac{1}{2}\sum_{a_i\in A} a_i$

$\Leftarrow$ קיימת ריצה של M בה תבחר את A_1 ותבדוק שהסכום שלה שווה חצי הסכום של A

$\Leftarrow$ קיימת ריצה של M בה תקבל את $\langle A\rangle$.

אם $\langle A\rangle \notin PARTITION$

$\Leftarrow$ לא קיימת חלוקה של A ל- A_1 ו- A_2 כך ש- $\sum_{a_i\in A_1} a_i = \sum_{a_i\in A_2} a_i = \frac{1}{2}\sum_{a_i\in A} a_i$

$\Leftarrow$ בכל ריצה של M על A היא תבחר תת-קבוצה A_1 ותבדוק ותדחה

$\Leftarrow$ בכל ריצה של M על $\langle A\rangle$, M תדחה את $\langle A\rangle$.

זמן הריצה של M פולינומיאלי בגודל הקלט $\langle A\rangle$.

<u>שאלה 5</u>

א) $M_B =$ על קלט $w' = \sigma_1 \dots \sigma_n$:

1) אם $w' = \varepsilon$ מריץ את M_A על w'.

- אם M_A מקבל $\Leftarrow$ M_B מקבל.
- אם M_A דוחה $\Leftarrow$ M_B דוחה.

2) $i \leftarrow 1$

3) בודק האם $\sigma_1 \cdots \sigma_i = \sigma_{i+1} \cdots \sigma_n$ (או לבדוק האם $i = \frac{n}{2}$)

- אם כן $\Leftarrow$ מריץ את M_A על $\sigma_1 \cdots \sigma_i$.
  - אם M_A מקבל $\Leftarrow$ M_B מקבל.
  - אם M_A דוחה $\Leftarrow$ M_B דוחה.

4) $i \leftarrow i+1$.

5)
- אם $i < n$ $\Leftarrow$ חוזר ל- (3).
- אחרת $\Leftarrow$ M_B דוחה.

<u>נכונות</u>

אם $w' \in B \Leftarrow$ שני מקרים:

- $w' = \varepsilon$ וגם $A \ni \varepsilon \Leftarrow$ M_B מקבלת את w'.
- $w' = ww \neq \varepsilon$ וגם $w \in A \Leftarrow$ עבור $i = \frac{|w'|}{2}$ מתקיים $\sigma_1 \cdots \sigma_i = \sigma_{i+1} \cdots \sigma_n$ וגם $\sigma_1 \cdots \sigma_i \in A$ $\Leftarrow$ באיטרציה i, M_B מקבלת את w'.

אם $w' \notin B \Leftarrow$ שני מקרים:

- $w' = \varepsilon$ וגם $\varepsilon \notin A \Leftarrow$ M_B דוחה את w'.
- $w' \neq \varepsilon \Leftarrow$ שני מקרים
  - עבור $i = \frac{|w'|}{2}$ מתקיים $\sigma_1 \cdots \sigma_i \neq \sigma_{i+1} \cdots \sigma_n \Leftarrow$ M_B דוחה את w'.
  - עבור $i = \frac{|w'|}{2}$ מתקיים $\sigma_1 \cdots \sigma_i = \sigma_{i+1} \cdots \sigma_n$ אבל $\sigma_1 \cdots \sigma_i \notin A \Leftarrow$ M_B דוחה את w'.

ב) נסמן ב- p_A הפולינום של M_A.

מבצעים לכל היותר $|w'|$ איטרציות ובכל איטרציה עושים בדיקה האם $\sigma_1 \cdots \sigma_i = \sigma_{i+1} \cdots \sigma_n$ בזמן $O(|w'|)$, ואם כן, מריצים את M_A על $\sigma_1 \cdots \sigma_i$ בזמן $p_A(|w'|)$.

ולכן זמן הריצה הוא

$$O\left(|w'|^2 + p_A(|w'|)\right)$$

<u>שאלה 6</u> הטענה נכונה.

ניתן לבנות אלגוריתם שיעבור על כל התתח-קבוצות בגודל 1000 קודקודים מ- G ויבדוק לכל תת-קבוצה האם היא קבוצה בלתי תלויה בזמן פולינומיאלי ויחזיר תשובה בהתאם.

מכיוון שמספר התת-קבוצות בגודל 1000 שווה $\approx 2^{1000}$ שזה קבוע, זמן הריצה של האלגוריתם פולינמיאלי.

<u>שאלה 7</u>

<u>בניית המכונה</u>

נבנה מ"ט דטרמיניסטית M שמכריעה את $2COLOR$ בזמן פולינומיאלי.
$M =$ " על קלט x:

1) בודקת אם $x = \langle G \rangle$ כאשר $G = (V, E)$ גרף לא מכוון.

- אם לא $\Leftarrow$ דוחה.

2) מאתחלת כל קודקוד בקבוצת הקודקודים כ "לא צבוע".

3) לכל קודקוד $s \in V$:

- אם s לא צבוע אז מגדירה
$c(s) = \text{red}$.
- לכל צלע $(u,w) \in E$:
  - אם u צבוע ו- w לא צבוע אז: $c(w) =$ הצבע ההפוך מ- $c(u)$
  - אם u לא צבוע ו- w צבוע אז: $c(u) =$ הצבע ההפוך מ- $c(w)$
  - אם u צבוע וגם w צבוע וגם $c(u) = c(w)$ אז דוחה.

4) עם בסוף הלולאה M לא דחתה אז מקבלת."

<u>הוכחת הנכונות</u>

<u>הוכחה לכיוון $\Leftarrow$</u>

אם $x \in 2COLOR$

$\Leftarrow$ $x = \langle G \rangle$ כאשר G גרף לא מכוון ו- $\langle G \rangle \in 2COLOR$.

$\Leftarrow$ קיימת פונקציית בצביעה כך ש- $c(u) \neq c(w)$ לכל $(u,w) \in E$.

$\Leftarrow$ לכל קודקוד $s \in V$, כל צלע (u,w) שניתן להגיע אליה מ- s מקיימת: $c(u) \neq c(w)$.

$\Leftarrow$ באף איטרציה של הלולאה M לא תדחה.

$\Leftarrow$ M תקבל.

<u>הוכחה לכיוון $\Rightarrow$</u>

אם $x \notin 2COLOR$

$\Leftarrow$ $x = \langle G \rangle$ כאשר G גרף לא מכוון ו- $\langle G \rangle \notin 2COLOR$.

$\Leftarrow$ לא קיימת פונקציית צביעה כך ש- $c(u) \neq c(w)$ לכל $(u,w) \in E$.

$\Leftarrow$ אם נצבע את הקודקודים של G לפי האלגוריתם של M, נמצא קודקוד $s \in V$, כך שקיימת צלע (u,w) שניתן להגיע אליה מ- s עבורה $c(u) = c(w)$.

$\Leftarrow$ קיימת איטרציה בלולאה של M שבה M תדחה.

10