סיבוכיות / תרגיל בית #1

אריאל סטולרמן

(1)

(a) $(5n)! = O(n!^5)$: **לא נכון**

תחילה נסתכל על משמעות קיום הביטוי $f = O(g)$. אם זה מתקיים, אז לכל n גדול מספיק יתקיים: $\frac{f(n)}{c\cdot g(n)} \xrightarrow[n\to\infty]{} a$ כאשר a הוא קבוע כלשהו ויכול להיות 0. אם זה לא מתקיים, אז: $\frac{f(n)}{c\cdot g(n)} \xrightarrow[n\to\infty]{} \infty$ לכל $c > 0$, כי $f(n) > c\cdot g(n)$.

נסתכל על הקירוב לעצרת לפי נוסחת סטרלינג, ונבדוק אסימפטוטית את הביטוי $\frac{(5n)!}{n!^5}$:

$$\frac{(5n)!}{n!^5} \approx \frac{\sqrt{5\cdot 2\pi n}\cdot\left(\frac{5n}{e}\right)^{5n}}{\sqrt{2\pi n}^5\cdot\left(\frac{n}{e}\right)^{5n}} = \frac{5^{\frac{1}{2}}\cdot\sqrt{2\pi n}\cdot\left(\frac{n}{e}\right)^{5n}\cdot 5^{5n}}{4\pi^2 n^2\cdot\sqrt{2\pi n}\cdot\left(\frac{n}{e}\right)^{5n}} = \frac{5^{5n+\frac{1}{2}}}{4\pi^2 n^2} \xrightarrow[n\to\infty]{} \infty$$

ולכן $(5n)! \neq O(n!^5)$.

(b) $f(n) = O(n) \Rightarrow 10^{f(n)} = O(2^n)$: **לא נכון**

אם $f(n) = O(n)$, אז קיים c_1 עבורו $f(n) \le c_1\cdot n$ (לכל n גדול מספיק). נראה שאם $c_1 > log_{10} 2$ אז הגרירה לא נכונה. בה"כ נבחר $c_1 = 2$:

$$\frac{10^{f(n)}}{2^n} = \frac{10^{2n}}{2^n} = \left(\frac{100}{2}\right)^n \xrightarrow[n\to\infty]{} \infty$$

לכן לא תמיד תתקיים הגרירה ולכן הטענה אינה נכונה.

(c) $log(n!) = \Theta(n\cdot log\, n)$: **נכון**

- נראה תחילה ש-$log(n!) = O(n\cdot log\, n)$:

$log(n!) = log(1\cdot 2\cdot \ldots\cdot n) \le log(\underbrace{n\cdot n\cdot \ldots\cdot n}_{n\ times}) = log\, n^n = n\cdot log\, n$ ומכאן שמתקיים: $log\, n = O(n\cdot log\, n)$.

- נראה כעת כי $log(n!) = \Omega(n\cdot log\, n)$:

$n! = 1\cdot 2\cdot\ldots\cdot(n-1)\cdot n$ ⇐ מחצית מהאיברים (בערך) גדולים ממש מ-$\frac{n}{2}$, ולכן: $n! \ge \underbrace{\frac{n}{2}\cdot\frac{n}{2}\cdot\ldots\cdot\frac{n}{2}}_{\frac{n}{2}\ times} = \left(\frac{n}{2}\right)^{\frac{n}{2}}$.

מכאן שמתקיים: $log(n!) \ge log\left(\left(\frac{n}{2}\right)^{\frac{n}{2}}\right) = \frac{n}{2}\cdot log\left(\frac{n}{2}\right)$, והרי $n\cdot log\, n = \Theta(\frac{n}{2}\cdot log\left(\frac{n}{2}\right))$, ומכאן: $n\cdot log\, n = \Omega(log(n!))$.

⇐ $log(n!) = \Theta(n\cdot log\, n)$, והטענה נכונה.

(d) לכל f, g פונקציות מתקיים: $f = O(g) \vee g = O(f)$: **לא נכון**

נקח למשל את $f = \lambda n.\begin{cases} 2^n, & n\in\mathbb{N}_{even} \\ log\, n, & n\in\mathbb{N}_{odd}\end{cases}$ ואת $g = \lambda n.\, n$. ברור כי אף פונקציה אינה חסם לשניה לכל n (עבור n גדול מספיק). דוגמא לתוכנית הרצה לפי f: תוכנית שבהינתן n מספר טבעי, אם הוא זוגי סופרת עד 2^n, ואם אי-זוגי מחזירה את סכום ספרותיו.

(e) $\exists f\ s.t: \forall\varepsilon > 0.\, f(n) = O(n^{1+\varepsilon}) \wedge f(n) = \omega(n)$: **לא נכון**

הביטוי $f(n) = O(n^{1+\varepsilon})$ יתקיים לכל $\varepsilon > 0$ רק אם $f(n) \le d\cdot n$, עבור $d > 0$ קבוע כלשהו. אבל, כדי שיתקיים $f(n) = \omega(n)$ צריך להתקיים ש-$f(n) > d\cdot n$. נקח למשל את $f(n) = d\cdot n$: צריך להתקיים ש-$\forall\varepsilon > 0.\, n \le \varepsilon\cdot d\cdot n$, והרי אם נקח $\varepsilon = \frac{1}{d+1}$ נקבל סתירה. לפיכך הטענה אינה נכונה.

(2) נניח שכל השפות מעל $\Sigma = \{0,1\}^*$.

(a) טענה: **המחלקה P סגורה תחת Δ**

יהיו $L_1, L_2 \in P$. נראה כי $L_1 \Delta L_2 \in P$:

$L_1, L_2 \in P$ ולכן לכל $x \in \{0,1\}^*$ ניתן להכריע בזמן פולינומיאלי האם $x \in L_1$ או $x \in L_2$. כמו כן ידוע כי $P = coP$ ולכן ניתן להכריע בזמן פולינומיאלי האם $x \notin L_1$ או $x \notin L_2$. כדי להכריע האם $x \in L_1 \Delta L_2$ יש לבדוק האם $(x \in L_1 \wedge x \notin L_2) \vee (x \in L_2 \wedge x \notin L_1)$, ולכן גם זאת ניתן להכריע בזמן פולינומיאלי, ומכאן ש-$L_1 \Delta L_2 \in P$ לכל $L_1, L_2 \in P$. לכן P סגורה תחת Δ.

(b) טענה: **המחלקה NP אינה סגורה תחת Δ**

נניח בשלילה שהטענה נכונה, אזי עבור $L_1, L_2 \in NP$ מתקיים: $L_1 \Delta L_2 = (L_1 \cup L_2) \setminus (L_1 \cap L_2) \in NP$. תהי $L_2 \in NP$ שפה כלשהי. נבחר את $L_1 = \{0,1\}^*$ (שפת כל המחרוזות). ברור כי מתקיים $L_1 \cup L_2 = L_1 \in P \subseteq NP$ וכי $L_1 \cap L_2 = L_2 \in NP$. נראה שלכל $x \notin L_2$, ניתן להכריע עם עדות פולי באורך x ובזמן פולי ש-$x \notin L_2$:

$x \notin L_2 \Leftrightarrow x \in L_1$ וגם $x \notin L_2 \Leftrightarrow x \in L_1 \cup L_2$ וגם $x \notin L_1 \cap L_2 \Leftrightarrow x \in L_1 \Delta L_2$, והרי לפי ההנחה ניתן להכריע עם עדות באורך פולי ובזמן פולי ש-$x \in L_1 \Delta L_2$. מכאן מקבלים: $L \in NP \Leftarrow L \in coNP$ ומכאן $NP \subseteq coNP$, אך זו שאלה פתוחה. לכן הטענה אינה נכונה.

(c) טענה: **המחלקה $NP \cap coNP$ סגורה תחת Δ**

יהיו $L_1, L_2 \in NP \cap coNP$, אזי $L_1 \Delta L_2 = (L_1 \cup L_2) \setminus (L_1 \cap L_2)$, וברור ש-$L_1 \cup L_2, L_1 \cap L_2 \in NP \cap coNP$. מתקיים ש-$x \in L_1 \Delta L_2$ אמ"מ $x \in L_1 \cup L_2$ וגם $x \notin L_1 \cap L_2$:

- $x \in L_1 \cup L_2$: $L_1 \cup L_2 \in NP \cap coNP \subseteq NP$ ולכן קיימת עדות פולי באורך x ואלג' הרץ בזמן פולי המוודא $x \in L_1 \cup L_2$.
- $x \notin L_1 \cap L_2$: $L_1 \cap L_2 \in NP \cap coNP \subseteq coNP$ ולכן קיימת עדות פולי באורך x ואלג' הרץ בזמן פולי המוודא $x \notin L_1 \cap L_2$.

לכן הבעיה השפה $L_1 \Delta L_2$ היא NP. באופן דומה ניתן להראות אי-שייכות ל-$L_1 \cup L_2$ (מתכונת ה-$coNP$יות שלה) ושייכות ל-$L_1 \cap L_2$ (מתכונת ה-NPיות שלה), ולכן $L_1 \Delta L_2$ היא גם $coNP$. לכן $L_1 \Delta L_2 \in NP \cap coNP$, ולכן הטענה נכונה.

(3)

(a) יהי קלט $G = (V, E)$ גרף (נניח לא מכוון) ו-k-מספר טבעי. להלן אלגוריתם המחשב בזמן פולינומיאלי האם קיים ב-G קליק בגודל k או צומת בעל דרגה $\le log_2(|V|)$:

- נחזיק מערך בגודל $|V|$ מאותחל ל-0 ונמספר את הצמתים באופן שרירותי $v_0, \dots, v_n$ ($|V| = n$). נעבור על כל הקשתות ולכל קשת $(u, v) \in E$ נוסיף 1 ל-u ול-v במערך במקום המתאים. אם בדרך אחד הצמתים קיבל ערך $\le log_2(n)$, נעצור ונחזיר "כן". הערה: בגרף מכוון נוסיף 1 רק לצומת u. אם לא, נעבור לשלב הבא.
- נבדוק מהי הדרגה המקסימלית לפי מערך דרגות הצמתים שבנינו. נסמן דרגה זו d_{max}. אם $k > d_{max}$, נחזיר "לא". אחרת:
- לכל קודקוד $i \in \{1, \dots, n\}$ נסתכל על קבוצת שכניו $V_i \subseteq V$. נבדוק את כל תתי הקבוצות בגודל k מתוך V_i, ואם אחת מהן היא $Clique$, נחזיר "כן". אם לכל הקודקודים ולכל הקבוצות שנבדקו לא נמצא $clique$, נחזיר "לא".

זמן ריצה:

- השלב הראשון סה"כ עובר על כל הקודקודים וכל הקשתות פעם אחת, לכן מבצע מספר לינארי של פעולות באורך הקלט.
- השלב השני עובר על רשימת הדרגות של כל הקודקודים, גם כן לינארי באורך הקלט.
- השלב השלישי מבצע $|V|$ פעמים את הפעולות הבאות: כיוון שהגענו לשלב זה, כל קודקוד הוא מדרגה לכל היותר $log_2(|V|) - 1$. לכן מספיק לבדוק לכל קודקוד $Clique$ הכולל אותו מתוך לכל היותר $log_2(|V|)$ (פחות 1) קודקודים – הוא ושכניו, ובטוח שלא ישתתף ב-$Clique$ גדול יותר. לכן נבצע בדיקות (בזמן פולי) על לכל היותר $\binom{log_2(|V|)}{k}$ תתי קבוצות (לכל קודקוד). כמו כן $k \in \{1, \dots, log_2(|V|)\}$.
  נשים לב כי מתקיים: $|V| = 2^{log(|V|)} = \sum_{i=1}^{log(|V|)} \binom{log(|V|)}{i} \ge \binom{log(|V|)}{k}$, ולכן סה"כ יתבצעו $\le |V| \cdot |V|$ פעולות בשלב זה.

סה"כ: זמן ריצה פולי.

נכונות:

- מוחזר "כן" בשלב הראשון אמ"מ קיים קודקוד בעל דרגה $\geq log_2(|V|)$.
- מוחזר "לא" בשלב השני אמ"מ לא קיים קודקוד בדרגה זו, וגם לא קיים קליק בגודל k, כי כלל לא קיים קודקוד בגרף שדרגתו $\geq k$.
- מוחזר "כן" בשלב השלישי אמ"מ קיים קליק בגודל k, ו"לא" אמ"מ לא קיים קליק בגודל k (ואין קודקוד מדרגה $\geq log_2(|V|)$ בגרף).

(b) תהי L רשימת מספרים טבעיים $a_1, \dots, a_n$, ויהי T מספר (טבעי), כולם מיוצגים בייצוג אונרי. להלן אלגוריתם פולי' המכריע האם קיימת $S \subseteq \{1, \dots, n\}$ כך ש-$\sum_{i \in S} a_i = T$:

- נחזיק מטריצה מסדר $n \times T$ שנסמנה A, ונגדירה כך: $A[i,j] = \begin{cases} 1, \exists S \subseteq \{1, \dots, i\}. \sum_{k \in S} a_k = j \\ 0, \quad o/w \end{cases}$.
- נמלא את A בריצה על השורות והעמודות (בשתי לולאות מקוננות), באופן הבא:

$$A[i,j] = \begin{cases} max\{A[i-1,j], A[i-1, j-a_i\}, & i,j > 0 \\ 1, & j = 0 \\ 0, & i = 0, j \neq 0 \end{cases}$$

- נחזיר "כן" אם $A[n,T] = 1$, אחרת נחזיר "לא".

זמן ריצה:

ברור כי מילוי המטריצה נעשה בזמן $O(n \cdot T)$, וכיוון שהקלט הינו בייצוג אונרי, ונניח כל a_i שונים מ-0, הקלט הקטן ביותר שיתקבל הוא בגודל $n + T$. על כן זמן הריצה הוא פולי' בגודל הקלט (לא היה כך לו הקלט היה בייצוג שאינו אונרי, למשל בינארי).

נכונות:

באינדוקציה על השורה במטריצה A אותה ממלאים.

עבור $k = 0$: אם $j = 0$, אזי ברור כי $A[0,0] = 1$, כיוון שניתן פשוט לקחת את הסכום הריק, דהיינו $\sum_{l \in \emptyset} a_l$. אם $j \neq 0$, ברור כי $A[0,j] = 0$ כיוון שלא ניתן לקבל סכום גדול מ-0 ע"י הסכום הריק.

נניח נכונות ל-k, נוכיח נכונות ל-$k + 1$: אם $j = 0$, ברור. אחרת: נניח ש-$A[k+1,j] = 1$, אז a_{k+1} יכול להיות חלק מהאיברים בסכום או לא. אם a_{k+1} נמצא בסכום שתוצאתו j, אזי שאר האיברים בסכום שלפני a_{k+1} נסכמים עד $j - a_{k+1}$ ומהווים תת קבוצה של $\{1, \dots, k\}$, ויתקיים כי $A[i-1, j-a_{k+1}] = 1$. אם a_{k+1} לא בסכום, יתקיים על אותו עקרון ש-$A[i-1,j] = 1$. לכן לקיחת המקסימום מביניהם תתן ערך נכון ל-$A[k+1,j]$.

לבסוף מתקיים כי: $A[n,T] = 1$ אמ"מ קיימת תת-קבוצה של האיברים $a_1, \dots, a_n$ שסכום איבריה שווה ל-T, כנדרש.

(c) תהי φ נוסחה מ-$3CNF$ בה כל פסוקית מכילה 3 ליטרלים שונים, וכל משתנה מופיע 3 פעמים בדיוק בנוסחה. להלן אלגוריתם פולי' המכריע האם φ ספיקה:

בהינתן נוסחה φ כמתואר לעיל, נבנה את הגרף הבא:

- V: לכל פסוקית נשים צומת ולכל משתנה נשים צומת.
- E: נחבר כל צומת משתנה לצומת פסוקית בו מופיע המשתנה (הוא או ההופכי שלו).

הגרף שהתקבל הוא גרף דו צדדי, כיוון שקיימת חלוקה ל-V: $V = \{\text{צמתים המייצגים פסוקיות}\} \uplus \{\text{צמתים המייצגים משתנים}\}$ כך שכל קשת היא מקבוצה אחת לשניה. כמו כן, לכל צומת בדיוק 3 שכנים: אם צומת הוא פסוקית, שלושה משתנים מתחברים אליו; אם צומת הוא משתנה, הוא יתחבר בדיוק לשלוש פסוקיות לפי הנתונים על φ. מכאן שהגרף הוא גרף רגיל.

לפי הרמז הנתון, כיוון שהגרף הוא דו-צדדי ורגיל, קיים זיווג מושלם. זיווג מושלם בגרף זה משמעו שלכל פסוקית ניתן להתאים משתנה אחד שהוא שיתן לה ערך אמת T. לכן השאלה היא טרוויאלית, ולכל נוסחה כמוגדרת לעיל יש השמה מספקת. לפיכך אלגוריתם מכריע לשפה הוא האלג' המחזיר לכל נוסחה כנ"ל "כן".

(4)

תהי $A \subseteq \{0,1\}^*$ כך ש-$|A \cap \{0,1\}^n| = n^3$ לכל $n \geq 10$. מכאן שלכל $n \geq 10$, A מכילה n^3 מילים באורך n. נראה כי בהינתן $A \in NP$ אזי גם $A \in coNP$:

כיוון ש-A ב-NP, מתקיים כי לכל קלט $x \in \{0,1\}^*$ קיים פולינום p ואלגוריתם M הרץ בזמן פולי' באורך x כך ש: $x \in A \Leftrightarrow \exists w \in \{0,1\}^{p(|x|)}. M(x,w) = 1$.

כדי להראות ש-$A = coNP$ נראה שלכל קלט $x \in \{0,1\}^*$ קיים פולינום q ואלגוריתם N הרץ בזמן פולי' באורך x כך ש: $x \notin A \Leftrightarrow \exists w \in \{0,1\}^{q(|x|)}. N(x,w) = 1$.

עבור קלט x באורך n: כיוון שלכל $n \geq 10$ יש n^3 מילים באורך n השייכות ל-A, ניקח את w להיות רצף אותן n^3 מילים באורך n שנסמנן x_i, ורצף n^3 עדויות שנסמנן w_i באורך $p(n)$ עבור אותן n^3 מילים, כדי להראות שהן אכן מילים ב-A.

אלגוריתם N יפעל באופן הבא: יוודא באמצעות M לכל מילה x_i עם העדות המתאימה לה w_i שאכן $x_i \in A$. לאחר מכן יוודא ש-$\forall i \in \{1, \dots, n^3\}. x \neq x_i$, ואם אכן כך יחזיר 1, אחרת יחזיר 0.

זמן ריצה:

- העדות w הינה באורך פולי' לאורך x: $w = x_1 w_1 x_2 w_2 \dots x_{n^3} w_{n^3}$ ולכן $|w| = n^3 \cdot (n + p(n))$ – פולי' באורך x, הוא q.
- N רץ בזמן פולי': ידוע כי M רץ בזמן פולי' לאורך x, וכל ש-N עושה הוא מריץ את M סה"כ n^3 פעמים, ולבסוף מבצע n^3 השוואות של שתי מחרוזות באורך n כל אחת (x משווה לכל x_i). לכן N רץ סה"כ גם כן בזמן פולי', כנדרש.

נכונות:

$x \notin A \Leftrightarrow$ קיימות $x_1, \dots, x_{n^3}$ מילים מ-A באורך n, כאשר כל אחת שונה מ-x, וכיוון שהן מ-A יש להן עדויות $w_1, \dots, w_{n^3}$ כך שלכל i: $M(x_i, w_i) = 1 \Leftrightarrow$ ריצת N על הקלט x יחד עם $w = x_1 w_1 \dots x_{n^3} w_{n^3}$ מחזירה 1.