תכנון אלגוריתמים – עבודה 6

תאריך הגשה: 27.06.2013, 12:00 בצהריים (כשהשמש למעלה בשמיים וחם בחוץ)

הוראות כלליות:

- כל עוד לא נאמר אחרת, כאשר הנכם מתבקשים לתאר אלגוריתם יש לספק את הבאות:
  1. תיאור מילולי של האלגוריתם.
  2. הוכחת נכונות.
  3. ניתוח זמן ריצה.
- אלגוריתם עם זמן ריצה אקספוננציאלי לא נחשב יעיל ולכן בדרך כלל לא יתקבל.
- פתרון יש לכתוב רק בדף התשובות הנלווה לעבודה.
- אם אתם נדרשים להוכיח טענה ואתם מוכיחים **במקומה** טענה אחרת שקולה, עליכם לנסח הטענה השנייה ולציין שהיא שקולה לטענה המקורית (יש לנמק זאת אם השקילות אינה טריוויאלית).
- כל **מונח** שאתם משתמשים בו חייב להיות מוגדר היטב: אם הוא לא הוגדר בקורס אחר או בקורס זה עליכם להגדיר אותו בעבודה.
- במידה ואתם רוצים להסתמך על משפט שהוכח בהרצאה יש לצטט אותו במדויק בדף התשובות. רק אז ניתן להשתמש בו בהוכחות.

שאלה 1:

נסמן ב- $S[i\ldots j]$ את תת המחרוזת של מחרוזת S מהעמדה ה- i ועד לעמדה ה- j (כולל). נסמן ב- S' את המחרוזת ההפוכה ל- S. הסעיפים הבאים הם בלתי תלוים.

סעיף א:

קלט: אליס ובוב מחזיקים מחרוזות $A=(a_0,\ldots,a_{n-1})$ ו- $B=(b_0,\ldots,b_{n-1})$ בהתאמה כך שמתקיים $A,B\in\{0,1\}^n$. כמו כן נגדיר ש- $A[i..j]=(a_i,\ldots,a_j)$. נשים לב שמתקיים ש $A[i..i]=(a_i)$ ועבור $j>i$ מתקיים $A[j..i]=(\,)$.

נאמר כי B היא מודיפיקציה של A אם קיימים $0\le k_1\le k_2\le n-1$ כך ש-
$A=B[0\ldots k_1-1]\circ(B[k_1\ldots k_2])'\circ B[k_2+1\ldots n-1]$ (כאשר $\circ$ הוא שירשור מחרוזות).
שימו לב שעל פי ההגדרה כל $B\in\{0,1\}^n$ היא מודיפיקציה של עצמה.

לדוגמא, המחרוזת $B=(010101)$ היא מודיפיקציה של $A=(011001)$ מאחר ועבור $k_1=2,k_2=3$ מתקיים כי $A=B[0..1]\circ(B[2..3])'\circ B[4..5]$.

מטרה: למצוא פרוטוקול בו אליס שולחת הודעה אחת בלבד לבוב, ובוב צריך להחליט אם B היא מודיפיקציה של A.

הציעו אלגוריתם הסתברותי לבעיה, בו אליס שולחת הודעה באורך $O(\log n)$ לבוב וההסתברות להצלחה היא לפחות $\frac{3}{4}$.
הוכיחו את נכונות האלגוריתם שהצעתם, נתחו את מספר הביטים שאליס שולחת, והסבירו מדוע החישוב של אליס ובוב הינו פולינומיאלי.
הערה: בתשובתכם יש לציין באופן מפורש באיזה תחום לבחור את הראשוניים כך שהשגיאה ואורך ההודעות יהיו כנדרש.

סעיף ב:

בהמשך לסעיף הקודם עליכם לענות על השאלה הבאה:

נניח שהמחשבים של אליס ובוב מאפשרים ביצוע פעולות אריתמטיות $\bmod p$ אך ורק אם p הוא לא יותר גדול מ-2^{64}. מהו ערכו הגדול ביותר של n עבורו יסתדרו התוכניות של אליס ובוב הבנויות בהתאם לאלגוריתם שהצעתם?

סעיף ג:

נזכר באלגוריתם קארפ-רבין למציאת כל ההופעות של תבנית P בטקסט T כך ש- $|P| = n$ ו- $|T| = m$, עם טעות חד-כיוונית. נניח כי הפעלנו את האלג' הזה k פעמים עבור אותו הזוג (P, T).

הוכחנו בהרצאות כי אם ניקח את k התשובות $S_1, \dots, S_k$ שהאלגוריתם מחזיר, ונחזיר את $S_1 \cap \dots \cap S_k$, אזי ההסתברות לשגיאה הינה לכל היותר $(\frac{nm}{\pi(I)})^k$, כאשר המספרים הראשוניים נבחרים מתוך האינטרוול $[2 \dots I]$ ו- $\pi(I)$ הינו מספר הראשוניים מ-2 עד I.

הוכיחו חסם נמוך יותר להסתברות לשגיאה של האלגוריתם הנ"ל, $m(\frac{n}{\pi(I)})^k$.

הדרכה: התבוננו בהסתברות ששגיאה מתרחשת בעמדה קבועה r ב- T, והוכיחו כי היא לכל היותר $(\frac{n}{\pi(I)})^k$.

המשיכו והוכיחו כי ההסתברות לכך שהאלג' יטעה בלפחות עמדה אחת r ב-T מתרחשת בהסתברות הנדרשת.

הערה: מותר להשתמש בעובדה הבאה מתורת ההסתברות

"חסם האיחוד": אם יש r מאורעות עם הסתברות a לכל אחת, אזי ההסתברות שלפחות אחד מהם יתרחש היא לכל היותר ra.

שאלה 2:

הגדרה: שפת ה- Binary-Inequalities או בקיצור BI היא אוסף כל מערכות האי-שוויונים כך שקיימת הצבה **בינארית** למשתנים במערכת כך שכל אי-שוויון במערכת יתקיים/יסתפק.

לדוגמא: עבור המערכת $\begin{bmatrix} x_1 + 2x_3 + 8x_4 \geq 7 \\ 4x_1 - 8x_3 \geq 4 \\ -x_1 - x_2 + 2x_3 \leq 2 \end{bmatrix}$

ההצבה הבאה תקיים/תספק את המערכת: $x_1 = 1; x_2 = 0; x_3 = 0; x_4 = 1$.

תזכורת:

- שפת SAT היא שפת כל הפסוקים שלהם ישנם הצבה מספקת.
- שפת VERTEX COVER היא שפת כל הצמדים של גרף לא מכוון G ומספר טבעי k כך שקיימת קבוצת קודקודים C בגודל k כך שלכל צלע ב G יש לפחות קודקוד אחד בקבוצה C.

סעיף א

הראו כי השפה BI שייכת למחלקה NP.

סעיף ב

תארו והוכיחו רדוקציה פולינומיאלית משפת SAT לשפת BI.

סעיף ג

תארו והוכיחו רדוקציה פולינומיאלית משפת Vertex Cover לשפת BI.

שאלה 3:

הגדרה: עבור שפה L שפה, השפה המשלימה $\bar{L}$ מוגדרת ע"י $\overline{L} = \{x \mid x \notin L\}$ או לחלופין האופן הבא

$x \in \overline{L} \leftrightarrow x \notin L$.

דוגמה:

$clique = \{(G = (V, E), k) | there\ exists\ a\ clique\ of at\ least\ size\ k\ in\ G\}.$

$\overline{clique} = \{(G = (V, E), k)\ | there\ is\ no\ clique\ of\ size\ k\ in\ graph\ G\}$

הגדרה: נאמר ששפה L שייכת ל $co - NP$ אם ורק אם השפה המשלימה $\bar{L}$ שייכת ל NP.

לאורך כל השאלה <u>נניח</u> ש: $NP \neq co - NP$ (עובדה זו אינה ידועה)

סעיף א':

טענות שהוכחו בכיתה ונשתמש בהם:

1. $P \subseteq NP$.
2. $L_1 \in NP \Leftarrow (L_1 \leq_p L_2) \wedge (L_2 \in NP)$.

עליכם להוכיח את שלושת הטענות הבאות:

טענה 3: $\bar{L} \in P \Leftarrow L \in P$.

טענה 4: $P \subseteq co - NP$.

<u>טענה עיקרית:</u> $NP \neq P$.

סעיף ב':

<u>הוכיחו את הטענה הבאה:</u> $L_1 \leq_p L_2 \Leftrightarrow \bar{L}_1 \leq_p \bar{L}_2$.

סעיף ג':

<u>הוכיחו את הטענה הבאה:</u> לא קיימת שפה L כך ש- $L \in NPC$ וגם $\bar{L} \in NPC$.

שאלה 4:

בשאלה זו נסתכל על שפה E-2-SAT. הקלט לשפה E-2-SAT הוא פסוק 2-CNF מוכלל המכיל, במקום ליטרלים, "ליטרלים מוכללים" מהצורה $x_i=b$, כאשר $b\in\{0,1,2\}$. כלומר, פסוק 2-CNF מוכלל הוא מהצורה $\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_n$, כאשר כל C_k היא פסוקית מהצורה $(x_i = b \vee x_j = c)$. הצבה אפשרית לכל משתנה היא מהקבוצה $\{0,1,2\}$והצבה מספקת לפסוק מוגדרת בצורה הטבעית. למשל לביטוי:

$$(x_1 = 2 \vee x_2 = 0) \wedge (x_3 = 1 \vee x_2 = 1) \wedge (x_1 = 1 \vee x_3 = 2),$$

ההצבה $x_1 = 1,\ x_2 = 0,\ x_3 = 1$ היא הצבה מספקת.

השפה E-2-SAT היא שפת כל פסוקי 2-CNF מוכללים שקיימת להם הצבה מספקת.

במטרה להוכיח $\text{E-2-SAT} \leq_P \text{2-SAT}$ הוצעה פונקצית הרדוקציה f הבאה.

יהי $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ מופע של בעיית E-2-SAT, כאשר משתני הנוסחה הם $x_1, x_2, \ldots, x_n$. הפונקציה f בונה פסוק $\psi = D_1 \wedge D_2 \wedge \ldots \wedge D_m$ באופן הבא:

- משתני הפסוק הם $x_{1,0}, x_{2,0}, \ldots, x_{n,0}, x_{1,1}, x_{2,1}, \ldots, x_{n,1}, x_{1,2}, x_{2,2}, \ldots, x_{n,2}$, כלומר לכל משתנה x_i הרדוקציה מגדירה שלושה משתנים $x_{i,0}, x_{i,1}, x_{i,2}$,
- לכל פסוקית מהצורה $(x_i = b \vee x_j = c)$ ב- φ, הרדוקציה בונה פסוקית $(x_{i,b} \vee x_{j,c})$ ב- ψ.

<u>סעיף א</u>

הראו כי הרדוקציה המוצעת שגויה.

רמז: קיימת נוסחה עם שלוש פסוקיות המראה זאת.

<u>סעיף ב</u>

בהינתן שלושה משתנים x_1, x_2, x_3 (המקבלים ערכים מהקבוצה $\{0,1\}$), בנו נוסחת 2-CNF המסתפקת אם ורק אם <u>לכל היותר אחד</u> מהמשתנים מקבל ערך 1. אין צורך להוכיח את נכונות הנוסחה.

<u>סעיף ג</u>

שנו את פונקציית הרדוקציה f רק ע"י <u>הוספת פסוקיות</u> לפסוק ψ כך שהרדוקציה תהיה נכונה. תהי g פונקציית הרדוקציה המתוקנת. הוכיחו כי g היא רדוקציה מ E-2-SAT ל- 2-SAT.

<u>סעיף ד</u>

בהסתמך על סעיף ג' ועל העובדה ש-2-SAT היא ב- P, הוכיחו שבעיית E-2-SAT היא ב- P.

שאלה 5:

סעיף א

הגדרה: בהינתן גרף $G=(V,E)$ קבוצת קודקודים $V'\subseteq V$ תקרא קבוצת בסיס ב-G אם לכל $v\in V$ או ש- $v\in V'$ או שקיים $u\in V'$ כך ש- $(u,v)\in E$.

בעיית Base(G,k):

קלט: גרף מכוון G ומספר שלם k.

יש למצוא: האם קיימת ב-G קבוצת בסיס בגודל k.

הוכיחו כי בעיית Base היא NP שלמה.

הדרכה: כדי להראות NP-קושי אפשר להשתמש בעובדה כי Set-Cover היא NP-שלמה.

תזכורת:

בעיית Set-Cover($S_1,S_2,\ldots,S_\ell,k$): בהינתן ℓ קבוצות $S_1,S_2,\ldots,S_\ell$ ומספר טבעי k, נסמן $U=\bigcup_{i=1}^{\ell}S_i$.

האם קיימות k קבוצות $S_{i_1},S_{i_2},\ldots,S_{i_k}$ כך ש $\bigcup_{j=1}^{\ell}S_{i_j}=U$.

סעיף ב

בעיית Sudoku($M_{n^2*n^2}$): מופע של בעיית Sudoku הינו מטריצה בגדול n^2*n^2 אשר חלק מהתאים בה ריקים וכל השאר מכילים מספרים טבעיים בין 1 ל n.

פתרון חוקי לבעיה (אם קיים עבור מופע נתון) הינו לוח כאשר בכל אחד מהתאים מציבים מספר טבעי בין 1 ל n. כך שבכל שורה ובכל עמודה של המטריצה יופיע כל מספר פעם אחת בדיוק. כמו כן אם נפרק את המטריצה לתתי מטריצות זרות בגודל $n*n$ אז בכל תת מטריצה יופיע גם כל אחד מהמספרים פעם אחת בדיוק.

לפרטים נוספים ראו http://en.wikipedia.org/wiki/Sudoku.

דוגמה ללוח בגודל 3 (מימין המופע ומשמאל הפתרון):

		4	6		8	9	1	2
	7	2				3	4	8
1			3	4	2	5		7
	5	9	7		1	4	2	
	2	6		5		7	9	
	1	3	9		4	8	5	
9		1	5	3	7			4
2	8	7				6	3	
3	4	5	2		6	1		

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

נסתכל על שתי גרסאות של בעיית ה Sudoku.

בעיית ההכרעה Sudoku($M_{n^2*n^2}$): אשר בהינתן לוח, מחזירה האם יש ללוח פתרון חוקי או לא.

בעיית החיפוש FindSudoku($M_{n^2*n^2}$): בו בהינתן לוח, יש למצוא לו פתרון במידה וקיים, אחרת להחזיר שאין פתרון.

הוכיחו כי אם קיים אלגוריתם פולינומי לבעיית ההכרעה Sudoku, אזי קיים אלגוריתם פולינומי לבעיית החיפוש FindSudoku.

יש לתאר את האלגוריתם לבעיית החיפוש ולהסביר מדוע הוא נכון.

בהצלחה!