שאלה 1

כדי להתאים את הבעיה לאלגוריתם גייל-שפלי, נתייחס לכל הספינות כאל גברים, ואל כל נמלי הים כאל נשים. את ההעדפות שלהם נבנה על פי לוחות הזמנים: כל ספינה מעדיפה להישאר לתחזוקה בנמל זמן רב ככל האפשר (ועל כן סדר ההעדפות שלה זהה לסדר שבו היא מבקרת בנמלים השונים), בעוד שכל נמל מעדיף שהספינה שתישאר בו לתיקונים תשהה זמן מועט ככל האפשר (ועל כן סדר ההעדפות שלו הפוך לסדר שבו מבקרות בו הספינות השונות). כלומר, עבור לוח הזמנים לדוגמה המוצג בשאלה, שתי הספינות תעדפנה את נמל P1 על פני נמל P2, בעוד ששני נמלי הים יעדיפו את הספינה S2 על פני הספינה S1.

לאחר שנבנה את רשימת ההעדפות, נריץ את אלגוריתם גייל-שפלי ונקבל זוגות של ספינות ונמלים - כל ספינה תשמור על לוח הזמנים שלה, עד שתגיע לנמל שאיתו זווגה ובו היא תישאר עד סוף החודש.

לוחות הזמנים הם בגודל של m2, ועל כן זהו סדר הגודל של בניית סדרי העדיפויות עבור הנמלים והספינות. זמן הריצה של אלגוריתם גייל-שפלי הוא O(n2). כיוון ש m≥n, ניתן לחסום את זמן הריצה הכולל ב-O(m2).

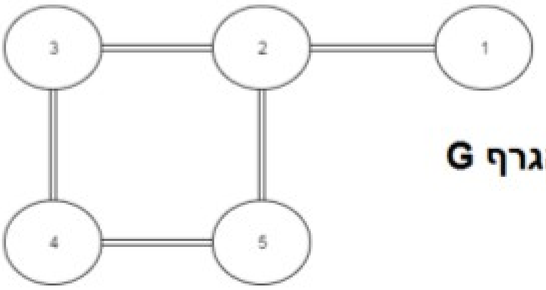
נוכיח כי לוח הזמנים המתקבל מקיים את התנאי הנדרש:
נניח בשלילה כי קיים נמל Pi וספינה Si שזווגה לו. בתאריך כלשהו לאחר היום שבו הספינה Si עוגנת בנמל ונותרת בו, מגיעה אל הנמל Pi ספינה Sj ומפירה את התנאי - כלומר, הנמל Pj שזווג לספינה Sj נמצא בהמשך המסלול, לאחר ההגעה ל-Pi.
על פי לוחות הזמנים, הנמל Pi מדורג במקום גבוה יותר מאשר Pj בסדר ההעדפות של Sj, כיוון שהיא מגיעה אליו קודם לכן. כמו כן, הספינה Sj תדורג במקום גבוה יותר מאשר Si בסדר ההעדפות של Pi, כיוון שהיא עוגנת בו בשלב מאוחר יותר. כלומר, Pj ו-Si מעדיפים זה את זה על פני "בני הזוג" הנוכחיים שלהם, משמע שאלגוריתם גייל-שפלי הפיק זיווג בלתי יציב, בניגוד לטענה 1.6 שהוכחה בעמ' 9-10 בספר הלימוד.

שאלה 2

כפועל יוצא מטענה 3.19 שהוכחה בעמ' 110-111 בספר הלימוד, גרף מכוון שבו דרגת הכניסה של כל צומת גדולה מאפס הוא גרף שכל אחד מרכיביו הקשירים מכיל מעגל (שאלמלא כן, רכיב זה היה גמ"ל, ואחד מצמתיו היה בעל דרגת כניסה של 0).

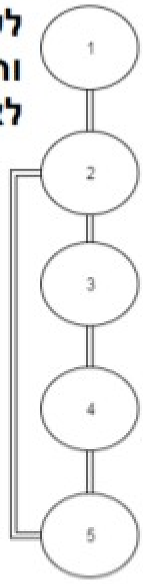
נבחר צומת v ונבצע ממנו סריקה לעומק של הרכיב הקשיר שלו 'G, כך שיווצר העץ T.
לאחר מכן, עבור כל צומת ברכיב הקשיר, נבדוק אם יש לו קשת לצומת שכבר נסרק. אם לא מצאנו קשת לאחור שכזו באף אחד מן הצמתים, האלגוריתם יחזיר FALSE, כיוון שהגרף נטול מעגלים.
אם קיימת קשת (v1,v2) כאשר v1,v2 ∈G′ אך (v1,v2) ∉T, נכוון אותה כך שתצא מ-v2 ותיכנס ל-v1.
נבצע סריקה לעומק של T, ונכוון את כל הקשתות לכיוון הסריקה. כלומר, קיבלנו גרף שבו דרגת הכניסה של כל צומת מלבד השורש היא 1, שכן לכל צומת מלבד השורש יש קשת היוצאת מן ההורה שלו ונכנסת אליו. אך גם דרגת הכניסה של השורש היא 1, שכן הצומת (v1,v2) נכנס אליו. כלומר, כל צמתי T -שהם גם צמתי 'G - מקיימים כעת את התנאי המבוקש.
נבצע את אותם הצעדים בכל אחד מן הרכיבים הקשירים של G.
את יתר הקשתות נכוון בצורה שרירותית - אין חשיבות לכיוון, שכן הרכיבים הקשירים של G מקיימים בין כה וכה את התנאי המבוקש.

סריקה לעומק של הרכיבים הקשירים מתבצעת בזמן O(m+n). אנו מבצעים סריקה שכזו פעמיים: פעם אחת על G ופעם נוספת על T (סריקה הכוללת גם הכוונת קשתות). הכוונת הקשתות לאחור (בין אם באופן מכוון עבור הקשת-לאחור הראשונה שמצאנו ובין אם באופן שרירותי עבור יתר הקשתות) חסומה אף היא ב- O(m), ועל כן האלגוריתם כולו מתבצע בזמן O(m+n).

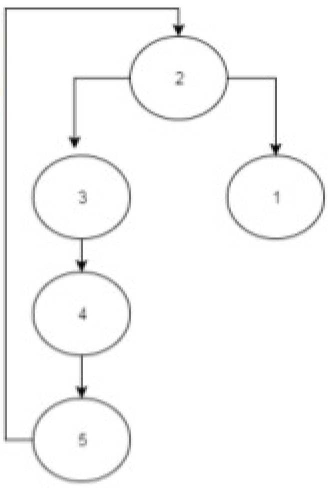


הגרף G

העץ T (עץ סריקה לעומק) והקשת לאחור (2,5)



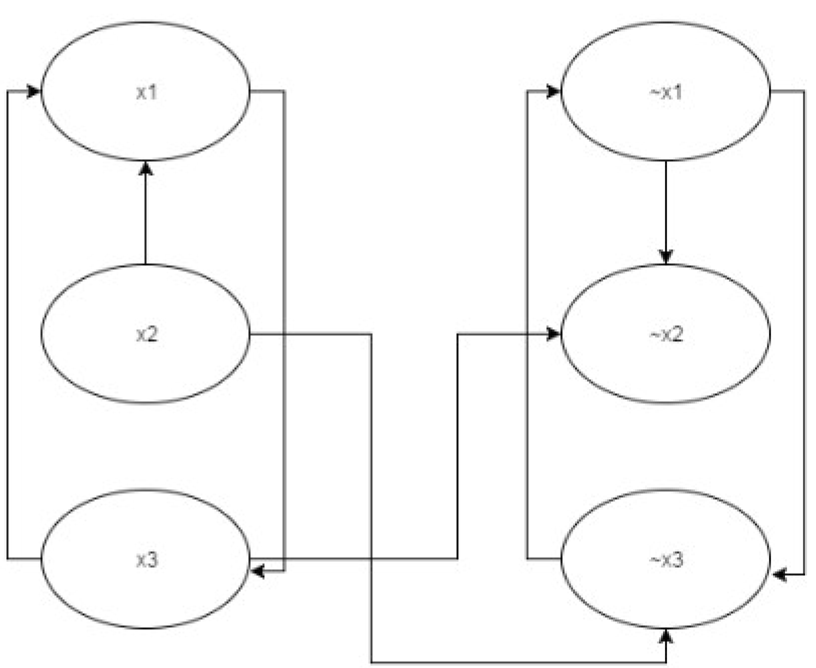
סריקה של T והכוונת קשתות



דוגמה להכוונת קשתות ע"פ האלגוריתם

<u>שאלה 3</u>

על פי חוקי תחשיב הפסוקים, הביטוי $(X \vee Y)$ שקול לביטוי $(\sim X \rightarrow Y)$ כמו גם לביטוי $(\sim Y \rightarrow X)$.
נבנה על בסיס הנוסחה φ גרף בן $2n$ צמתים ו-$2m$ קשתות באופן הבא:
כל צומת יציין את אחד מהליטרלים $x_1 \ldots x_n$, $\sim x_1 \ldots \sim x_n$. לכל פסוקית φ_i נבנה שתי קשתות שייצגו את יחסי הגרירה השקולים לפסוקית.



הגרף המתקבל עבור נוסחת 2-CNF מן השאלה: $(x_1 \vee \sim x_2) \wedge (x_1 \vee \sim x_3) \wedge (\sim x_1 \vee x_3) \wedge (\sim x_2 \vee \sim x_3)$

כיוון שיחס הגרירה הוא טרנזנטיבי, הרי שאם קיים ב-G מסלול (x,y,z…w) הרי שמתקיים

$x \rightarrow y \rightarrow z \ldots \rightarrow w$

ועל כן מתקיים

$x \rightarrow w$

הנחת עבודה: נוסחת 2-CNF אינה ספיקה אם"ם היא שקולה לסתירה, היינו אם קיים בה ליטרל x שעבורו מתקיים $x \Leftrightarrow \sim X$.
הכיוון הראשון טריוויאלי, שכן במידה והנוסחה שקולה ל-$x \Leftrightarrow \sim X$, אזי אין השמה של ערך כלשהו ל-x שתתן לנוסחה ערך אמת.
על פי הגרף שהגדרנו, מצב כזה מתקיים אם קיים מסלול בין x ל-x~ ובין x~ ל-x, כלומר $(x \rightarrow \sim x) \wedge (\sim x \rightarrow x)$.
את הכיוון השני אוכיח בהמשך.

האלגוריתם יעבוד כך:

- עבור כל ליטרל $x \in \{\sim x_1..\sim x_n, x_1..x_n\}$:
  - באמצעות סריקה לעומק/רוחב, חפש מסלול בין הצומת x לצומת x~.
  - אם נמצא מסלול שכזה (היינו, אם $x \rightarrow \sim x$)
    - באמצעות סריקה לעומק/רוחב, חפש מסלול בין הצומת x~ לצומת x.
    - אם נמצא מסלול שכזה: (היינו, אם $\sim x \rightarrow x$)
      - החזר ערך שקר (שכן קיבלנו $x \Leftrightarrow \sim x$)

הלולאה החיצונית מתבצעת O(n) פעמים, וכוללת בתוכה עד שתי סריקות של של הגרף בעלות זמן של O(n+m) כל אחת.
מכאן, שקביעה אם הנוסחה ספיקה או לא לפי אלגוריתם זה אורכת O[n(n+m)].

בשלב הבא, נניח שהנוסחה ספיקה. בשלב זה, נגדיר שני סוגי רכיבים קשירים:
רכיב קשיר של צומת x ייקרא ***שקרי בהכרח*** אם קיים ליטרל y שעבורו מתקיים $(x \rightarrow y) \wedge (x \rightarrow \sim y)$ – כלומר, שב-G קיים מסלול מ-x ל-y ומ-x ל-y~. ביטוי שכזה יכול לקבל ערך אמת רק אם x מקבל ערך שקר.

אם לא קיים ברכיב קשיר של צומת x ליטרל y שכזה, הרכיב ייקרא ***לא שקרי בהכרח***.

בדיקה האם רכיב הוא שקרי בהכרח תעשה כך: בזמן הסריקה של הרכיב, בכל פעם שצומת x מתווסף לעץ הסריקה, ייצבעו הצמתים x ו-x~ בצבע מסויים למשך הסריקה. אם במהלך הסריקה נתקל בצומת צבוע אשר טרם סרקנו, סימן שכבר סרקנו את ההופכי שלו, כלומר שקיים ברכיב מסלול הן ל-x והן ל-x~, ומכאן שהרכיב שקרי בהכרח. כיוון שהוספנו מספר קבוע של פעולות לכל איטרציה של לולאת הסריקה, ניתן לקבוע אם רכיב קשיר הוא שקרי בהכרח באותו סדר גודל זמן של סריקה רגילה – $O(m+n)$.

מציאת השמה מספקת תיעשה כך:

- עבור כל ליטרל $x \in \{ \sim x_1..\sim x_n, x_1..x_n\}$ שלא הושם לו ערך אמת או שקר:
  - באמצעות סריקה לעומק/רוחב, מצא את הרכיב הקשיר של x.
    - אם הרכיב הקשיר של x שקרי בהכרח או אם יש בו ליטרלים שקיבלו ערך שקר:
      - השם ערך שקר ב-x וערך אמת ב-x~.
    - אם הרכיב הקשיר של x אינו שקרי בהכרח ואם אין בו ליטרלים שקיבלו ערך שקר:
      - השם ערך אמת בכל צמתי הרכיב הקשיר של x, וערך שקר בכל הצמתים ההופכיים שלהם.
- כאשר לכל ליטרל הוצב ערך אמת או שקר, החזר את ההשמה שהתקבלה.

בדומה לחלק הראשון של האלגוריתם, גם כאן מתבצעות $O(n)$ חקירות של הרכיב הקשיר, ולכן העלות של חלק זה אף היא $O[n(m+n)]$.

נראה כי האלגוריתם נכון (ובכך גם את הכיוון השני של הנחת העבודה):
נניח בשלילה כי קיימים ליטרלים x_i,x_j המקיימים $(\sim x_i \rightarrow \sim x_j) \wedge (\sim x_i \rightarrow x_j) \wedge (x_i \rightarrow \sim x_j) \wedge (x_i \rightarrow x_j)$
כל ביטוי מסוג $x \rightarrow y$ נוצר כתוצאה מביטוי דוגמת $(\sim x \vee y)$, ועל כן אם קיים מסלול בין x ל-y אזי קיים גם מסלול בין y~ ל-x~.
מכאן שמתקיים $x_i \rightarrow \sim x_j \rightarrow \sim x_i$ – אך לא אכן היה כך הדבר, היה בגרף מסלול בין x_i ל-$\sim x_i$, והחלק הראשון של האלגוריתם היה מחזיר ערך אמת.
מכאן, שאם רכיב קשיר של x הוא שקרי בהכרח, אזי הרכיב הקשיר של x~ אינו שקרי בהכרח – ועל כן אפשר להציב ב-x ערך שקר וב-x~ ערך אמת מבלי לגרום לסתירה.

בגרף המתקבל, אם הוא עבר את השלב הראשון, אין צומת שלא ניתן לצבוע כאמת או כשקר. ועל כן, כשנסיים לצבוע את כל הצמתים ואת ההופכיים שלהם כאמת וכשקר, נקבל השמה מספקת ל-φ .

שאלה 4

(א) נגדיר קבוצה 'V בת 2n צמתים. כל צומת מקדד בתוכו צומת ב-G ומספר בין 1 ל-0 המעיד על מספר הביקורים ב-U.
נגדיר קבוצה 'E בת 2m צמתים לכל היותר:
עבור כל קשת $(v1,v2) \in E$, הקבוצה 'E תכיל את הקשתות הבאות:
אם $v1,v2 \notin U$: $([v1,0],[v2,0])$ ו-$([v1,1][v2,1])$
אם $v1 \notin U$, $v2 \in U$: $([v1,0],[v2,1])$
אם $v1 \in U$, $v2 \notin U$: $([v1,1],[v2,1])$
אם $v1,v2 \in U$: $\emptyset$

כעת נבצע סריקה לרוחב של הגרף $G'=(V',E')$ בכדי למצוא את המסלול הקצר ביותר מ-$(s,0)$ ל-$(t,1)$. מובטח לנו כי המסלול הוא מסלול באורך מזערי, כיוון שהוא נוצר כתוצאה מסריקה לרוחב. כמו כן, המסלול עובר ב-U פעם אחת בלבד, שכן אין קשתות המאפשרות מעבר בין צמתים שונים ב-U, או מעבר אל תוך U לאחר שהתבצע ביקור אחד ב-U.

בניית הגרף עוברת פעם אחת על כל צומת ועל כל קשת. הגרף שנוצר הוא בסדר גודל של m+n, ועל כן בניית הגרף 'G ומציאת המסלול בו מתבצעת בזמן של $O(m+n)$.

(ג) בדומה לסעיף א', נגדיר קבוצה V' הכוללת צומת ומונה. המונה עדיין מקבל את אחד מן הערכים 0 ו-1, אך כעת הם מייצגים את השארית מהביטוי $\#P(U)/2$.
בהתאם לכך, נגדיר את הקשתות הבאות ב-E', לכל קשת מסוג $(v1,v2)\in E$:
אם $v2\notin U$: $([v1,0],[v2,0])$ ו-$([v1,1][v2,1])$
אם $v2 \in U$: $([v1,1],[v2,0])$ ו-$([v1,0][v2,1])$

כלומר, בכל פעם שאנו עוברים באחד מצמתי U, המונה מקבל 1 אם זהו ביקור אי-זוגי, או 0 אם זהו ביקור זוגי.

בדומה לסעיף א', נבצע סריקה לרוחב של הגרף $G'=(V',E')$, אך הפעם נחפש את המסלול המזערי מ-$(s,0)$ ל-$(t,0)$.
במסלול כזה, מובטח לנו שלא עברנו דרך צמתי U מספר אי-זוגי של פעמים (שאם לא כן היינו מגיעים אל t,1).
באופן אנלוגי לסעיף א', זמן הריצה של האלגוריתם הוא $O(m+n)$.

(ב) לצורך סעיף זה, נבצע סריקה לרוחב של הגרף G החל מהצומת s. בכל פעם שאנו מוסיפים צומת לעץ הסריקה, נקודד יחד איתו את המידע הבא:
א. מרחק: המרחק מן השורש s. המרחק של הצומת s הוא 0.
ב. $\#P(U)$ - מספר הצמתים ב-U שהמסלול עובר דרכם. בשורש s, מספר זה עומד על 0.

נשנה מעט את אלגוריתם הסריקה לרוחב, כך שבכל פעם שאנו סורקים צומת v, אנו עוברים על כל הקשתות שלו מסוג (v,u) - גם אלו המקשרות לצמתים שכבר נסרקו.

- אם הצומת u לא נסרק בעבר, הוא מתווסף אל העץ. הוא מקבל את המרחק של ההורה v ועוד אחת, ואת $\#P(U)$ של ההורה v.
- אם הצומת u שייך ל-U, מתווסף אחת לשדה $\#P(U)$ של הצומת.
- אם הצומת נסרק בעבר באמצעות הקשת (v',u):
  - אם שדה $\#P(U)$ של הצומת v' בעץ הסריקה גדול משדה $\#P(U)$ של ההורה v, הסריקה נמשכת כרגיל.
  - אם שדה $\#P(U)$ של הצומת v' בעץ הסריקה קטן ממש משדה $\#P(U)$ של v
  וגם שדה המרחק של הצומת v' בעץ הסריקה שווה לשדה המרחק של v :
    - אזי משנים את מיקומו של הצומת u בעץ הסריקה כך שההורה שלו הוא v ולא v'. מעדכנים את שדה $\#P(U)$ בהתאם.

כאשר מגיעים לצומת t, ממשיכים בסריקה לפי אותם הכללים, עד אשר המרחק של הצומת הנסרק u שווה לשדה המרחק של t בעץ הסריקה.

המסלול היורד משורש העץ ל-t הוא מסלול מזערי (שכן הסריקה שמרה על הסדר בין הרמות המתקיימים בסריקה לרוחב).
הוא עובר דרך מספר מקסימלי של צמתים ב-U, שכן שינינו את המבנה שלו כך שימקסם את השדה $\#P(U)$.
כיוון שכל ההשוואות ושינויי המקום בעץ הסריקה מתבצעים כחלק מן האיטרציה של הסריקה ודורשים מעבר על מספר קבוע של צמתים בכל פעם, זמן הריצה של האלגוריתם נותר $O(m+n)$.