ממ"ן 11 – לאה בן צבי

**שאלה 1**

**תקציר הבעיה:** בהינתן ספינות (נסמן אותן ) ו- נמלים (נסמן אותם ). לכל ספינה ישנו לוח זמנים בן ימים, כשלכל יום רשום "בלב ים" או שם של נמל. יש לכתוב אלגוריתם שיוצר עבור כל ספינה לוח זמנים חדש, כך ש:

א) ספינה לא יכולה לעגון בנמל בו ספינה אחרת כבר עוגנת.

ב) כל ספינה עוגנת לפי לוח הזמנים בנמל אחד, בדיוק יום אחד בחודש.

**רעיון האלגוריתם:**

נשתמש באלגוריתם "זיווג יציב" שהוגדר בפרק 1 עם המימוש שהוגדר בפרק 2 (כאמור, מספר הנמלים שווה למספר הספינות) הזיווג מתבצע לפי רשימת ההעדפות של הספינות ושל הנמלים המוגדרות כך:

1. לכל ספינה , נגדיר שהיא מעדיפה את נמל על פני נמל אם מופיע בלוח הזמנים של לפני .
2. לכל נמל , נגדיר שהוא מעדיף את על פני אם מופיע, ע"פ לוח הזמנים (הלא מקוצץ), אחרי .

יהי ו- ובסוף האלגוריתם נקבל את הזוגות הסדורים ו-. נגדיר מצב שבו מגיעה ל- לפני כאי-יציבות (כי אז הייתה צריכה לבחור את על פני ).

**תיאור האלגוריתם:**

1. **בניית רשימת ההעדפות של הספינות:** עבור כל ספינה , נעבור על כל :
   1. אם מופיע שם של נמל ביום אז מוסיפים אותו לסוף רשימת ההעדפות של הספינה .
   2. אחרת – עוברים ליום הבא.
2. **בניית רשימת ההעדפות של הנמלים:** עבור כל נמל נבנה רשימת העדפות כך:
   1. נעבור בסדר יורד על כל יום בחודש בסדר יורד (מיום ליום 1):
      1. נעבור בסדר הפוך על הספינות ב- (מ- ל-1) ועבור על ספינה נבדוק ב- אם ביום הספינה תבקר בנמל .
         1. אם הספינה מבקרת בנמל – מוסיפים את הספינה לתחתית רשימת ההעדפות של הנמל.
         2. אחרת – ממשיכים לספינה הבאה.
3. נפעיל את אלגוריתם "זיווג יציב" כאשר קבוצת הגברים וקבוצת הנשים באלגוריתם המקורי הן קבוצות הספינות והנמלים בהתאמה. בסוף האלגוריתם נקבל קבוצה של זוגות סדורים:

.

1. עבור כל ספינה , נסרוק את כל ימי החודש (בסדר עולה):
   1. אם בלוח הזמנים שלה כתוב שהיא תעגון ב- כך ש- אז נעדכן ביום זה כי עליה להישאר בנמל זה עד סוף החודש ולא להמשיך בלו"ז של חודש זה.
   2. אחרת – עוברים ליום הבא.

**נכונות האלגוריתם:**

1. *נתון כי כל ספינה עוגנת בכל נמל בדיוק יום אחד. אז כשנסרוק את הלו"ז של הספינות, נמצא בדיוק נמלים וכל נמל יופיע שם רק פעם אחת, אז ברשימת ההעדפות של כל ספינה יופיע מספר זה של נמלים כך שכל נמל יופיע פעם אחת בלבד (סה"כ נמלים ברשימה).*
2. *כאמור, נתון כי כל ספינה עוגנת בכל נמל בדיוק יום אחד. אז בכל רשימת העדפות של כל נמל, יופיעו כל הספינות (מכיוון שכל ספינה צריכה להגיע ביום כלשהו לכל נמל), כך שכל ספינה תופיע בדיוק פעם אחת (כי כל ספינה תבקר בכל בנמל בדיוק פעם אחת).*
3. נכונות האלגוריתם "זיווג יציב" הוכחה בספר הלימוד.
4. מסעיף 3 נובע כי לכל ספינה יזווג רק נמל אחד (וההפך), ובנוסף נתון כי קיים רק יום אחד שבו ספינה צריכה לבקר בנמל (לפי לוח הזמנים המקורי). על כן, עבור כל ספינה ימצא רק יום אחד שבו הספינה תישאר בנמל . על כן, בסוף סעיף זה, לכל ספינה יהיה לוח זמנים מעודכן שבו יוקצה לך נמל אחד בדיוק להישאר בו.

*נשאר להוכיח כי אם הוא זיווג שנוצר ע"י האלגוריתם ו- הגיעה ל- ביום , לא תהיה ספינה שתגיע לנמל ביום המקיים :*

*נניח, על דרך השלילה, כי קיימת ספינה כזו . כי כי לא יתכן ששתי ספינות אמורות להגיע ל- באותו יום. אם , אז הספינה הגיעה לנמל לפני אז זו היא אי-יציבות (כפי שהגדרנו). קיבלנו אי-יציבות אחרי ביצוע האלגוריתם "זיווג יציב" וזאת בסתירה ליציבות של הזיווג שהוא תוצאה של אלגוריתם זה (הוא בהכרח נטול אי-יציבויות). על כן, ההנחה כי קיימת כזו היא שגויה.*

***סיבוכיות האלגוריתם:***

1. *בניית רשימת ההעדפות של הספינות כרוכה במעבר על כל הספינות וכל- הימים לכל ספינה - .*
2. *בניית רשימת ההעדפות של הנמלים כרוכה במעבר על כל הימים, ועבור כל יום כל הספינות - .*
3. *יעילות האלגוריתם "זיווג יציב" הוא .*
4. *עדכון לוחות הזמנים כרוך במעבר על הזיווגים ועבור כל זיווג על (מספר היום שבו הספינה צריכה לעגון בנמל שלה) ולכן היעילות היא.*

*נסכום ונקבל . היות ו- נאמר שהיעילות של האלגוריתם היא .*

***שאלה 2***

***רעיון האלגוריתם:***

*כדי לכוון גרף בלתי-מכוון כך שלכל צומת תהיה לפחות קשת אחת שנכנסת אליה, יש לוודא שבכל רכיבי הקשירות שלו יהיה לפחות מעגל אחד. נשתמש באלגוריתם המוגדר במדריך הלמידה בע"מ 27, היוצר עבור כל רכיב קשירות עץ-סריקה לעומק. בכל עץ כזה נבדוק האם הוא מכיל מעגל, אם לפחות אחד מהם מכיל מעגל – אזי קיים צומת (או יותר) שדרגת הכניסה שלו היא 0. נסמן ב- את קבוצת רכיבי הקשירויות.*

***תיאור האלגוריתם:***

1. *נריץ את אלגוריתם על גרף (גרף הקלט).*
2. *עבור כל רכיב-קשירות , יהי קבוצת עצי הסריקה-לעומק שנוצר בסעיף 1. עבור כל נבדוק אם הוא מכיל מעגל:*
   1. *אם כל הקשתות הן קשתות עץ (ואין קשתות חוזרות) – נחזיר* False *ונסיים את האלגוריתם.*
   2. *אחרת – קיימת קשת חוזרת ונבצע את הפעולות הבאות:*
      1. *נכוון אותה כ-. ונדפיס את הכיוון.*
      2. *עבור כל קשת היוצאת מ-, נכוון אותה כ- ונדפיס את הכיוון הנבחר. וכל הקשת היוצאת מ- נכון אותן שיצאו מ- לשכנים שלהם וכך הלאה באופן רקורסיבי עד שכל הקשתות מכוונות וכל הכיוונים מודפסים.*
      3. *כאשר נגיע חזרה ל- נעבור לרכיב הקשירות הבא.*
3. *נדפיס* True *ונסיים את האלגוריתם.*

***נכונות אלגוריתם:***

*נכונות האלגוריתם של* DFS-Loop *מוכחת בספר. בסוף אלגוריתם זה נקבל את עצי הסריקה-לעומק של כל רכיבי הקשירות בגרף. עצים אלה מכילים קשתות עץ וקשתות חוזרות (אם יש כאלו).*

*יהי רכיב-קשירות. אם אין קשתות חוזרות בעץ של (כל הקשתות הן קשתות עץ), אז וגם כלומר הוא עץ כמו . מתקיים וגם כלומר עץ ועל כן אינו מכיל מעגלים. נסמן ולפי משפט 3.1 מתקיים . נניח על דרך השלילה כי ניתן לכוון את כך לכל צומת דרגת הכניסה שלה תהיה לגדולה מ-0. אזי מתקיים . אך אם סה"כ דרגות הכניסה הוא בגרף הוא ויש צמתים אז בהכרח קיים קודקוד שדרגת הכניסה שלו היא 0. אם לפחות רכיב קשירות אחד יכיל צומת כזו, הגרף מן סתם יכיל צומת כזו על כן נחזיר False.*

*אם לא כל הקשתות הן קשתות עץ אז יש לפחות קשת חוזרת אחת , נכוון אותה כ- ואז נכוון את כל הקשתות השכנות שלה שיצביעו לכיוון שכניהם וכן האלה באופן רקורסיבי, בדומה לאלגוריתם BFS, כך שכל צומת בדרגה תצביע לצומת בדרגה . וכך כל הצמתים ברכיב הקשירות (כולל צומת ההתחלה, שכן קיום המעגל מבטיח שהוא ישתייך לרמה האחרונה) יהיו בעלי דרגת כניסה הגדולה מ-0. אם כל הרכיבי קשירות ניתנים לכיוון כך שכל הצמתים יהיו בעלי דרגת כניסה גדולה מ-0, הגרף בהכרח לא יכיל צומת כזו ונוכל להחזיר הכוונת צלעות נכונה.*

***סיבוכיות האלגוריתם:***

*סיבוכיות האלגוריתם DFS על גרף היא . עבור כל רכיב קשירות בגרף מתקיים לכן סיבוכיות האלגוריתם היא .*

*עבור כיוון האלגוריתם השתמשנו באלגוריתם הפועל כמו לכן סיבוכיותם זהה - . הסיבוכיות של האלגוריתם היא .*

***שאלה 3***

***רעיון האלגוריתם:***

*ניצור גרף שהצמתים בו הם הליטרלים בנוסחה, הקשתות יהיו קשרי הגרירה כך שאם היא פסוקית ב- אז מוסיפים את הקשת עבור אם וגם עבור . עבור כל משתנה נבצע:*

1. *סריקה-לעומק בעזרת אלגוריתם DFS מ-.*
2. *סריקה-לעומק בעזרת אלגוריתם DFS מ-.*

*לשני התהליכים האלה יש 3 תוצאות אפשריות:*

1. *אם לא נמצא בעץ הסריקה מ- אז בהשמה המספקת ויש לסמן את כל הפסוקים הנגררים לוגית אחריו כ- אחרת הפסוק יקבל את הערך .*
2. *אם יהיה בסריקה מ- ו- לא נמצאת בסריקה מ- אז הביטוי נכון עבור השמה - וכל פסוקים שנגררים לוגית אחריו גם כן יקבלו את הערך .*
3. *אם יהיה בסריקה מ- ו- גם נמצאת בסריקה מ- אז ו- ואז ל- אין השמה מספקת.*

*אם למשתנה אחד אין השמה, אין השמה מספקת לכל .*

***תיאור האלגוריתם:***

1. *נבנה גרף מכוון כך ש:*
   1. *צמתי הגרף יהיו כל הליטרלים בנוסחה כלומר, עבור כל משתנה נוסיף את הליטרלים לצמתי הגרף.*
   2. *לכל פסוקים מהצורה נוסיף את הקשתות .*
2. *כל עוד קיים צומת שלא עבר השמה נבצע:*
   1. *נריץ את האלגוריתם DFS על צומת זה, נסמן ב- את העץ המתקבל מסריקה זו:*
      1. *אם לא נמצא ב-:*
         1. *נבצע את ההשמה לכל הצמתים הנמצאים בעץ הסריקה.*
         2. *נבצע את ההשמה לכל הצמתים הנגדיים שלהם.*
      2. *אחרת – נריץ סריקת DFS על , נסמן ב- את העץ המתקבל מסריקה זו.*
      3. *אם לא נמצא ב-:*
         1. *נגדיר לכל צומת בעץ הסריקה.*
         2. *נגדיר לכל הצמתים הנגדיים להם.*
      4. *אחרת – נחזיר False.*
3. *נחזיר*

***נכונות האלגוריתם:***

*באלגוריתם זה נסתמך על כך שקשר הגרירה הוא טרנזיטיבי (כלומר עבור פסוקים אם ו- אז מתקיים ) ואז אם במסלול סריקה-לעומק של מצומת נתקל בצמתים ו- לפי הסדר, אז יתקיים .*

*כדי למצוא את ההשמה המספקת עבור יש לבדוק את כל הליטרלים עבור ערך הנכון להם בהשמה. בכל איטרציה על צומת שעדיין אין לו השמה אנו מבצעים כדי למצוא אם בעץ הסריקה-לעומק שלו.*

*אם לא נמצא אז ניתן ל- את ערך האמת ולכל הקשרים הנגררים לוגים ממנו ניתן הערך . אם נתקלנו ב- בסריקה ונתנו לו את הערך יש לתת ל את הערך .*

*אם נמצא, בגלל טרנזיטיביות קשר הגרירה מתקיים כי . נבצע את הסריקה-לעומק מ-.*

*אם לא נמצא את אז, כאמור, ואז כדי שהפסוק יהיה נכון צריך לקבל את הערך ו- את הערך וכך גם כל הפסוקים הנגררים לוגית אחריו. כאמור, הצמתים של הפסוקים הנגדיים צריכים לקבל את הערך .*

*אם קיבלנו ש- נמצא בסריקה-לעומק מ- אז מטרנזיטיביות קשר הגרירה נובע כי אך גם וזו סתירה מכיוון ש- לא יכול לקבל את הערך וגם לא את הערך ולכן לליטרל זה אין השמה. אם לאחד הליטרלים אין השמה, אין השמה מספקת לכל הפסוק ולכן נחזיר .*

***סיבוכיות האלגוריתם:***

*בניית הגרף היא הזמן של הוספת צמתים ו- קשתות ולכן .* ***(המשך בעמוד הבא)***

*במקרה הגרוע (אם הוא בן של ) אז נבצע DFS פעמיים על כל איטרציה בלולאה. היות וברגע שהאלגוריתם נתקל בסתירה הוא מחזיר אז יש מספר קבוע של צמתים עליהם ירוץ האלגוריתם עד שיצא ולכן מתבצעות השמות לליטרלים. וכן, סיבוכיות האלגוריתם DFS הוא . על כן סיבוכיות האלגוריתם כולו הוא .*

***שאלה 4***

**רעיון האלגוריתם:**

עבור הגרף , מסלול מ- ל- העובר דרך 2 צמתים מועדפים בדיוק צריך לעבור לפחות שני שלבים:

1. שלב – קשת נכנסת לצומת וקשת שתצא מצומת זה.
2. שלב – קשת נכנסת לצומת ( יכול להיות שווה ל-) וקשת שתצא ממנו.

במקרה שבו המסלול מ- ל- יכיל רק את שני הצמתים מ- המסלול יראה כך: .

מסלול שעונה לתנאי השאלה יכול לעבור בכל אחד משלושת השלבים הבאים:

1. שלב – יהיו צמתים השייכים ל- שהקשתות היוצאות מהם יחברו בין ל-.
2. שלב – יהיו צמתים השייכים ל- שהקשתות היוצאות מהם יחברו בין ל-.
3. שלב – יהיו צמתים השייכים ל- שהקשתות היוצאות מהם יחברו בין ל-.

נבנה גרף מכוון כך שגרף זה יסמן את קשתות לפי חמשת השלבים שפירטנו. נריץ סריקה-לרוחב BFS על שתביא לנו את המסלול הקצר ביותר מ- ל- (מסלול כזה ב- בהכרח יעבור בשני צמתים מועדפים).

**תיאור האלגוריתם:**

1. *נגדיר את*
2. *נבנה מ- גרף מכוון חדש באופן הבא:*
   1. *לכל קודקוד מועדף ב- נוסיף את הצמתים ל-.*
   2. *נעתיק ל- את .*
   3. *לכל קודקוד ניצור ב- את הקודקודים .*
   4. *לכל צומת :*
      1. *אם אז*
   5. *עבור כל צומת :*
      1. *אם אז*
   6. *עבור כל צומת :*
      1. *אם אז*
   7. *עבור כל צומת , :*
      1. *אם אז*
      2. *אם אז*
   8. *לכל צומת :*
      1. *אם אז*
   9. *לכל צומת :*
      1. *אם אז*
   10. *לכל צומת :*
       1. *אם אז*
3. *נריץ סריקת BFS על גרף החל מ-. ברגע שהאלגוריתם ימצא את זה יהיה המסלול הקצר ביותר עם כל התנאים שאנחנו מחפשים. אם לא נמצא מסלול אז לא קיים כזה.*

***נכונות האלגוריתם:***

1. *:*

*קשת מאחד מהקודקודים של קודקוד או לאחד הקודקודים של או בגרף היא קשת בין שני קודקודים אלה גם בגרף . על כן אם נמצא מסלול ב- עבר את כל השלבים מ- ל- הוא יהיה מסלול ב- שעובר דרך שני צמתים מומלצים.*

1. *המסלול הנמצא הוא המסלול הקצר ביותר:*

*BFS מוצא את המסלול הקצר ביותר, האלגוריתם מחפש בדרגות של מרחק, אם היה מסלול קצר יותר היה מופיע ברמה קודמת לרמה שהאלגוריתם מצא.*

1. *המסלול שמצאנו קיים ב-:*

*הוכח בסעיף 1.*

***סיבוכיות האלגוריתם:***

*בניית הגרף מתבצעת ע"י מעבר על כל הצמתים והקשתות של ולכן סיבוכיותה . סיבוכיות ה של סריקה-לרוחב היא . סיבוכיות האלגוריתם היא .*