מזל אישטה – 305778466

**פתרון ממן 11 – אלגוריתמים-20417**

1. **בעיית השידוך היציב**

* רעיון האלגוריתם

נשתמש באלגוריתם הזיווג היציב המתואר בספר (אלגוריתם G-S) כאשר S קבוצת הספינות(במקביל לקבוצת הגברים M) ו -P קבוצת הנמלים(במקביל לקבוצת הנשים W)

"נשדך" כל ספינה לנמל כך שכל זיווג ייצג ספינה שהגיע לנמל ונשארת שם לצורך תחזוקה עד סוף החודש.

מספר הנמלים = למספר הספינות כנדרש לאלג' הזיווג היציב כך שלא ייתכן ש2 ספינות יהיו משויכות לאותו הנמל.

נגדיר את ההעדפות של S ו-P בצורה הבאה:

* + העדפות של ספינה:
    - ספינה תעדיף את נמל על פני נמל אם נמל נמצא בלוח הזמנים של לפני נמל , כאשר .
    - כך שהנמל הראשון אליו מגיעה הספינה נמצאת בעדיפות גבוהה יותר שלה, הנמל השני בעדיפות השנייה וכן הלאה.
  + העדפות של נמל:
    - נמל יעדיף את ספינה על פני ספינה אם ספינה מגיעה לנמל לפי לוח הזמנים המקורי **אחרי** ספינה , כאשר .
    - כך שהספינה שצריכה להגיע אחרונה לנמל p- תהיה בעדיפות הראשונה שלו, הספינה שצריכה להגיע לפניה תהיה בעדיפות השנייה וכן הלאה.
* תיאור האלגוריתם
  + בניית רשימת ההעדפות של הספינות מתוך הלו"ז המקורי.

לכל ספינה נסרוק את לוח הזמנים ובכל פעם שנפגוש בנמל נוסיף אותה לסוף רשימת העדפות של ספינה

* + בניית רשימת ההעדפות של הנמלים מתוך הלו"ז המקורי.
    - נעבור בלולאה על כל m הימים של החודש בסדר יורד - 1<=i<=m
      * נעבור על כל n הספינות בסדר יורד 1<=j<=n
        + נבדוק אם ספינה j עוגנת בנמל כלשהו p ביום i

אם כן – נכניס לסוף רשימת ההעדפות של p את j.

אחרת – נמשיך לאיטרציה הבאה.

* + - אלגו' הזיווג היציב(אלגוריתם G-S):
      * בתחילת הריצה – הספינות והנמלים "פנויים"
      * כל עוד קיימת ספינה s פנויה שלא הציעה נישואין לנמל כלשהו היא תציע נישואין לנמל הראשון ברשימת ההעדפות שלה, שלא הציע לו בעבר, נסמן אותו ב p
        + אם p לא מאורס – הוא יסכים להצעה, וs,p - יתארסו.
        + אם p מאורס לספינה אחרת אך s גבוהה יותר ברשימת ההעדפות שלו 🡨 האירוסין עם s’ יתבטלו וs,p- יתארסו.
        + אחרת (p מאורס ל s’ וגם s נמוכה ברשימת ההעדפות של p) – לא יתבצעו אירוסין וs תעבור בלולאה לנמל הבא ברשימת ההעדפות שלה.
      * בסיום הלולאה – כל קשרי האירוסין יהפכו לנישואין וסיימנו את תהליך השידוך.
* נכונות
  + נכונות האלגוריתם נובעת מנכונות אלגוריתם הזיווג היציב G-S. עלינו להראות שאילוצי האלגוריתם מתקיימים:
    1. רשימת העדפה של כל ספינה מכילה את כל הנמלים בדיוק פעם אחת – **מתקיים מהנתון כי כל ספינה נמצאת בכל נמל בדיוק יום אחד במהלך החודש.**
    2. באותו אופן רשימת העדפה של כל נמל מכילה את כל הספינות – **נובע בדיוק מהנתון שהובא לעיל**
  + כעת נראה כי לא ייתכנו 2 ספינות באותו הנמל באותו היום (בדיוק כמו שלא ייתכן כי 2 נשים נשואות לאותו הגבר או להיפך)

לכל ספינה לוח הזמנים המקורי שלה, אך כאשר היא מגיעה לנמל אליו היא "נשואה", היא עוצרת בו לתחזוקה עד לסוף החודש. צ"ל שכאשר ספינה עוצרת לתחזוקה בנמל מסוים, לא תבקר שום ספינה אחרת באותו נמל (שכן אסור ל2 ספינות להיות יחדיו באותו נמל ממניעי בטיחות).

במילים אחרות, עלינו להראות שאם ספינה s "נשואה" לנמל p, והיא מגיעה אליו ביום t, אז לא קיים

שעוגן בנמל p בזמן .

נניח בשלילה שקיים כזה. אם קיים כזה, אז פירוש הדבר הוא שבלוח הזמנים המקורי הספינה אמור לעגון בנמל p בזמן מאוחר יותר מאשר *הספינה s. אם כך הדבר, אז הדירוג של הוא גבוה יותר משל s ברשימת ההעדפות של הנמל p, לפי הגדרת רשימת ההעדפות של הנמלים. לפיכך, כאשר הנמל יצטרך לברור מבין s ל (כאשר לאחד מהם הוא "מאורס", והאחר בדיוק "מציע נישואין"), הוא יעדיף את , שכן הוא מדורג גבוה יותר. הנמל p יעדיף "להתארס" ל במקום לs. אך ידוע לנו שp וs "נשואים", בסתירה להנחה.*

*לכן – לא* קיים שעוגן בנמל p בזמן .

מש"ל.

* סיבוכיות
  + **רשימת העדפות של ספינות-** עבור הגדרת רשימת העדפות מתבצעת סריקה עבור n ספינות כך שעבור כל ספינה מתבצעת סריקה על m הימים (לולאה מקוננת) -> זמן הריצה הוא
  + **רשימת העדפות של נמלים-** עבור הגדרת רשימת העדפות מתבצעת סריקה עבור m הימים בחודש כך שעבור כל יום מתבצעת סריקה על n הספינות(לולאה מקוננת) -> זמן הריצה הוא
  + זמן הריצה של אלגו' הזיווג היציב הוא לפי עמוד 52 בספר.
  + נסכם ונקבל כי :
  + נתון כי לכן:

1. **הכוונת צלעות**

* רעיון האלגוריתם

ע"מ לכוון גרף כלשהו G כך שדרגות הכניסה של כל צומת תהיה גדולה מ0 , תנאי מספיק שיהיה לפחות מעגל אחד **בכל** רכיב קשירות של הגרף G.

במקרה בו אכן קיים מעגל בכל רכיב קשירות נסמן צומת כלשהו במעגל כשורש ונבצע ממנו סריקה לרוחב ונכוון את הקשתות כך שתצא קשת מצומת ברמה ה-i לצומת ברמה ה-i+1 (למעט קשתות חוזרות-שאותם נכוון "ידנית")

* תיאור האלגוריתם
  + במידה ויש יותר מרכיב קשירות אחד נריץ את האלגוריתם הא על כל רכיב קשירות בנפרד
  + נבחר קדקוד כלשהו (שלא בוצעה עליו סריקת DFS) ונבצע עליו סריקת DFS , נקבל עץ T.
    - //
    - אחרת קיימת קשת חוזרת כלומר קיימת לפחות קשת אחת בG שלא קיימת בT), וכוון קשת זו
      * כעת נכוון את הקשתות הלא מכוונות בגרף באמצעות סריקת BFS החל מהצומת כך שכל צומת היוצאת מ תצביע "החוצה" כלפי הצומת ברמה הבאה, כלומר כל צומת ברמה הi יצביע העל הצמתי הרמה הi+1
  + אם הוחזר FALSE עבור אחד מרכיבי הקשירות האלגוריתם יחזיר FALSE אחרת TRUE
* נכונות
  + T הוא תוצאה של סריקת DFS ולכן יש לו 2 סוגי קשתות: קשתות עץ וקשתות חוזרות.
  + אם כל הקשתות הן קשתות עץ נקבל כי * , ולכן גם G עץ.*
    - *נסמן ולכן לפי משפט 3.1 , .*
    - *נניח בשלילה שניתן להגדיר גרף מכוון -*

*אז מתקיים ,* ***אבל*** *אם סך דרגות הכניסה הוא* **n-1** *ולכן בהכרח קיימת צומת שדרגת הכניסה שלה היא 0 והאלגוריתם יחזיר False*

* + אחרת – *קיימת לפחות קשת חוזרת אחת, נסמנה . ונכוון אותה כ.* 
    - *לv בהכרח יש קשתות-יוצאות (אחרת לא הייתה קשת חוזרת), נכוון את הקשתות (היוצאות) בעזרת BFS לפי שכבות, כאשר v הוא השורש (כל הקשתות של צמתים ברמה מצביעים על צמתים ברמה )*

*בצורה זו אכפנו לכל צומת - דרגת כניסה בגודל 1 לפחות.*

* סיבוכיות
  + סריקת רכיבי הקשירות השונים באמצעות סריקת DFS/BFS - (הוכחה בספר)
  + בניית עץ DFS -
  + כיוון הקשתות בעזרת BFS -
  + *סה"כ זמן ריצה* ***a***

1. **בעיית הספיקות(2-SAT)**

* רעיון האלגוריתם

נמיר בעיה זו לגרף מכוון וננסה לפתור אותה בעזרת אלגוריתמים ידועים.

ההמרה לגרף תיעשה בצורה הבאה:

**צמתים** :נגדיר 2 צמתים לכל ליטרל (הליטרל והשלילה של אותו ליטרל)

**קשתות**: נתרגם כל פסוקית לקשרי גרירה :

כלומר לכל פסוקית מהצורה נגדיר 2 קשתות מכוונות:

נחפש בגרף מסלול שאין בו "סתירות" כלומר קיים מסלול שהליטרל והשלילה שלו לא באותו המסלול(לכל ליטרל). אם מצאנו אחד כזה הרי שישנה השמה לליטרלים כך שהנוסחא תחזיר TRUE. אחרת אם לא מצאנו מסלול אחד כזה הרי שישנם "סתירות" בכל המסלולים, כלומר בכל מסלול קיים ליטרל שהוא ושלילתו באותו המסלול ונקבל וזהו משפט סתירה

* תיאור האלגוריתם
  + בניית הגרף המכוון G=(V,E) באופן שתואר לעיל:
    1. לכל ליטרל x נוסיף לגרף את הצמתים .
    2. לכל פסוקית מהצורה נגדיר 2 קשתות מכוונות:
  + נעבור על כל קדקוד שלא עבר השמה(שלא סומן כ True/False )
    1. נבצע סריקת DFS(v), נסמן את תוצאות הסריקה T
       1. אם נבצע את ההשמה הבאה: T לכל צומת ב וF לכל צומת שלא נמצאת ב
       2. אחרת נבצע סריקת DFS() ,נסמן את תוצאות הסריקה INV
          1. אם נבצע את ההשמה הבאה: T לכל צומת ב וF לכל צומת שלא נמצאת ב
          2. אחרת נחזיר False
  + אם לא קיבלנו False הרי שנמצאה השמה מתאימה לנוסחא ונחזיר TRUE
* נכונות
  + בכל איטרציה על צומת v שעדיין אין לו השמה נבצע עליו סריקת DFS ונקבל את רכיב הקשירות שלו. אם נמצא בסריקה זו אזי אי אפשר לבצע השמה כיון שבהמשך נקבל את הסתירה שלו באותו המסלול. אם אכן קיבלנו סתירה ישנה אפשרות שישנה השמה במסלול ההפוך- כלומר נבצע סריקת DFS על . אם נמצא בסריקה זו אזי אי אפשר לבצע השמה כיון שבהמשך נקבל את הסתירה שלו באותו המסלול. ולא קיימת השמה אפשרית לנוסחא זו.

אחרת קיימת השמה ונוכל להגדיר ערכי T\F לכל הליטרלים כך שלא תהיה סתירה לוגית

בסוף הריצה האלגוריתם הצליח להגדיר ערכי אמת ושקר **לכל הליטרלים,** ולא נמצאו שום סתירות (ז"א לא קיים מסלולי שקר -> אמת או להיפך). לפי האופן בו בנינו את הגרף, המסלולים שקולים לתנאים הבולינאים המקוריים, ולכן ערכי הליטרלים מספקים את הנוסחה המקורית.

* סיבוכיות
  + בניית הגרף - (עבור m פסוקיות וn ליטרלים)
  + מעבר על כל הצמתים שלא עברו השמה(מספר הצמתים קבוע)
    1. ביצוע סריקת DFS במקרה הגרוע פעמיים -
    2. ביצוע השמה T\F :מתבצעת השמה ל2k צמתים(לכל ליטרל והיפוכו) כאשר k<=n . מספר הצמתים הוא קבוע ולכן ההשמה מתבצעת בזמן ליניארי כלומר
  + סה"כ

1. **מסלולים מזעריים דרך קדקודים מועדפים**

* רעיון האלגוריתם

נמיר את הגרף G לגרף G' ונפתור בעזרת רדוקציה.

"נשכפל" את הגרף 3 פעמים כאשר מעבר מגרף לגרף יתבצע דרך מעבר בקודקוד מועדף בלבד כלומר כדי לעבור בין 3 הגרפים צריכים לעבור בין 2 קודקודים מועדפים **בדיוק**.

נסמן ב את קבוצת הקודקודים שאינם קודקודים מועדפים ואינם הקודקודים s,t

מעברים יוגדרו בצורה הבאה :

*(\*) לאפשר מעבר ישיר מקודקוד מועדף בשלב 2 לקודקוד מועדף בשלב 4.*

אם קיים מסלול מ- בגרף הראשון ל- בגרף השלישי אזי מתקיימים תנאי השאלה. מסלול זה ניתן למצוא בעזרת סריקת BFS שבעזרתה נקבל את המסלול הקצר ביותר

* תיאור האלגוריתם
  + נסמן ונבנה את הגרף בצורה הבאה:
    1. **קודקודים**:
       1. נגדיר את הקודקודים ב V'
       2. לכל נוסיף את הקודקודים ב V'
       3. *לכל* נוסיף את הקודקודים ב V'
    2. ***קשתות:***
       1. *לכל קשת ניצור בגרף החדש את*
       2. *לכל קשת ניצור בגרף החדש את*
       3. *לכל קשת ניצור בגרף החדש את*
       4. *לכל קשת ניצור בגרף החדש את*
       5. *לכל קשת ניצור בגרף החדש את*
       6. *לכל קשת ניצור בגרף החדש את*
       7. *לכל קשת ניצור בגרף החדש את*
       8. *לכל קשת ניצור בגרף החדש את*
       9. *לכל קשת ניצור בגרף החדש את*
       10. *לכל קשת ניצור בגרף החדש את*
       11. *לכל קשת ניצור בגרף החדש את*
  + *נריץ BFS() בגרף החדש . אם נגיע לקודקוד בסריקה הרי שהו המסלול הקצר ביותר בין s ל t (במסלול המקורי זהו הגרף ללא הסימונים שהוספנו). אם לא הגענו ל אזי לא קיים מסלול מ s לt העובר דרך ד קודקודים מועדפים בדיוק.*
* נכונות
  + אילוץ 1 -
    1. ע"י הגדרת הקודקודים והקשתות לעיל אנו אוכפים אילוץ זה.
    2. הסבר המעברים בצורה סכמתית:
    3. *כל מסלול בG’ המסתיים בשלב מקיים את תנאי השאלה.*
  + *אילוץ 2 – המסלול הקצר ביותר*
    1. *שימוש בסריקת BFS מבטיח שנקבל את המסלול הקצר ביותר בגרף G'* *(מהגדרת BFS שעובר רמה אחרי רמה בגרף)*
  + *אילוץ 3 – המסלול קיים ב G*
    1. *ע"י הסרה של התיוגים של הקודקודים והסימונים מהקשתות –נקבל מסלול תקין בG המקורי.*
* סיבוכיות
  + בניית הגרף G’ מתבצעת ע"י מעבר על הקודקודים והקשתות של G פעם אחת ולכן
  + סריקת BFS מעל G’ לכל היותר (לכל היותר פי 3 קודקודים וקשתות מגרף המקור G. זהו סכום קבוע ולכן לא משפיע על זמן הריצה)
  + סיכום זמן הריצה :