**ממ"ן 11 אלגוריתמים**

**שאלה 1**

ניתן להיעזר באלגוריתם הזיווג היציב לפתרון הבעיה כאשר אם מקבילים את הבעיה הנוכחית לבעיית הזיווג היציב כפי שהיא מתוארת בעמוד 4 בספר אז הספינות מייצגות את הגברים, הנמלים מייצגים את הנשים ועגינה של ספינה בנמל מייצגת חתונה של גבר עם אישה. נשים לב שמספר הנמלים ומספר הספינות זהה (n) כדרוש באלגוריתם הזיווג היציב, ואכן לא יתכנו 2 ספינות באותו הנמל.

בשביל לפתור את הבעיה בצורה נכונה בעזרת אלגוריתם הזיווג היציב נצטרך לבנות רשימת העדפות בצורה מתאימה.

העדפות של ספינה – נגדיר את רשימת העדפות של הספינות לפי סדר ההגעה שלה לנמלים בלוח הזמנים המקורי כך שהנמל שכל ספינה מגיעה אליו ראשונה יהיה הכי גבוה ברשימת העדפות. בצורה יותר פורמלית: לכל ספינה S נגדיר את העדפה של ספינה לנמל P1 על פני נמל P2 אם"ם נמל P1 נמצא בלוח הזמנים של S לפני נמל P2.

רשימת העדפות של נמל – נגדיר את רשימת העדפות של הנמלים לפי סדר הספינות שמגיעות אליהם כך שהספינה שמגיעה לנמל אחרונה תהיה הכי גבוהה ברשימת העדפות. בצורה יותר פורמלית: לכל נמל P נגדיר את העדפה של נמל לספינה S1 על פני ספינה S2 אם ספינה S1 מגיעה לנמל לפי לוח הזמנים המקורי אחרי ספינה S2.

**האלגוריתם:**

תחילה נבנה את רשימות ההעדפות,

לכל ספינה Si(כאשר , (1≤i≤n נעבור על הימים של הלו"ז המקורי שלה מ1 עד m , כאשר כל נמל שמופיע בלו"ז יכנס לסוף רשימת ההעדפות.

נעבור על הימים d מm עד 1 ולכל ספינה Si(כאשר (1≤i≤nאם היא עוגנת בP ביום d בלו"ז המקורי אז נכניס אותה לסוף רשימת ההעדפות של P.

כעת יש לנו רשימת העדפות ולכן אנחנו יכולים פשוט להשתמש באלגוריתם הזיווג היציב כפי שהוא מוגדר בעמודים 6-7 בספר ונקבל את כל ההגינות של הספינות בנמלים כך שכל תנאי השאלה מתקיימים.

נכונות האלגוריתם - נתון שכל ספינה עוגנת בכל נמל בדיוק פעם אחת לכן ברשימת ההעדפה של כל נמל מופיעות כלל הספינות בדיוק פעם אחת ובאופן דומה ברשימת ההעדפות של כל ספינה מופיעים כלל הנמלים, בנוסף העגינה היא עד סוף החודש לכן כל ספינה עוגנת רק בנמל אחד (ובכל נמל עוגנת רק ספינה אחת) וקיבלנו שמתקיימים התנאים של אלגוריתם הזיווג היציב.

נראה כי מתקיים התנאי שאסור לשתי ספינות להיות באותו יום באותו נמל, כלומר שאם s ו- p "נשואים" כאשר s מגיעה לp ביום d לא תגיע אף ספינה לנמל p בימים: d≤t≤m:

נניח בשלילה שs,p "נשואים" כאשר s מגיעה ל p ביום d, וספינה s’ מגיעה ביום t כאשר

, נובע מכך שהנמל p היה ברשימת ההעדפות של s’ במקום גבוה יותר מאשר הנמל p' שבו תוכנן שs' תעגון וגם שהספינה s' הייתה במקום גבוה יותר מהספינה s ברשימת ההעדפות של p בסתירה לכך שs,p "נשואים" בזיווג יציב.

סיבוכיות האלגוריתם –

חישוב רשימת ההעדפות של הספינות מתבצעת עבור כל אחת מ n הספינות באמצעות מעבר על כל אחד מm הימים כלומר .

חישוב רשימת ההעדפות של הנמלים מתבצעת עבור כל אחד מm הימים בחודש, עבורו סורקים את כל n הספינות כלומר .

זמן הריצה של אלגו' הזיווג היציב הוא לפי משפט 2.10 בעמוד 52 בספר.

לאחר סכימה של זמני הריצה נקבל

ולפי הנתון לכן נקבל: .

**שאלה 2**

ניתן לכוון את הגרף כך שלכל קודקוד דרגת הכניסה תהיה גדולה מאפס אם ורק אם יש בגרף מעגל לכן נשתמש בDFS בשביל למצוא מעגל ולכוון את הגרף.

**אלגוריתם –** נריץ DFS על קודקוד כלשהוא בגרף G ונסמן את העץ שקיבלנו בT, אם כל הקשתות בG נמצאות גם בT (כלומר אין קשתות חוזרות) אז אין מעגל והאלגוריתם יחזיר False. אחרת קיימת קשת חוזרת . נכוון אותה כ(u,v). *ניגש לכל הקשתות הלא מכוונות שיוצאות מv ונכוון אותן כך שיצאו מv ונמשיך ככה בצורה רקורסיבית (דומה לביצוע BFS על v כך שלכל רמה i תשלח קשת מכוונת לרמה i+1)*, בסוף הריצה (שנגיע חזרה לv) כל הקשתות יהיו מכוונות ונחזיר True.

נכונות האלגוריתם – T הוא תוצאה של סריקת DFS ולכן כל הקשתות הן קשתות עץ או קשתות חוזרות, אם כל הקשתות הן קשתות עץ אז** , ולכן G הוא עץ ואם נסמן אז לפי *טענה 3.1 מעמוד 84 מה שאומר שסכום דרגות הכניסה הוא n-1 ובהכרח קיים קודקוד שדרגת הכניסה שלו היא 0. אם קיימת לפחות קשת חוזרת אחת, שהאלגוריתם מכוון אותה כ, לv בהכרח יש קשתות-יוצאות (אחרת לא היתה קשת חוזרת) והמשך הכוונת הקשתות בדרך מבוססת BFS מבטיחה שלכל צומת דרגת הכניסה תהיה 1 לפחות.*

*זמן ריצה - שימוש ב DFS לסריקת הגרף G צורך וכיוון הקשתות בצורה מבוססת BFS לכן זמן הריצה הכולל הוא .*

**שאלה 3**

רעיון האלגוריתם הוא להמיר את הפסוקיות לקשרי גרירה, ליצור גרף מכוון שבו הקשתות מסמלות גרירה ואז בעזרת DFS לבדוק אם יש ליטרל שגורר לשלילתו מה שמראה על סתירה.

**אלגוריתם –**

*תחילה* נבנה גרף מכוון , לכל ליטרל ניצור את הקודקודים ו *. לכל פסוקית מהצורה u V v=φ מתקיימת השקילות לכן ניצור את הקשתות* , (.

כעת *נבחר קודקוד (שעוד לא עבר השמה) ונריץ סריקת DFS ממנו,* ונסמן את העץ שקיבלנו ב*K*. *אם לא נמצא ב, נבצע את ההשמה T לכל קודקוד בקבוצה שנמצאה, (ו-F לכל הקודקודים הנגדיים שלהם כמובן)*, *אחרת נמצא ב* מה שאומר שקיבלנו סתירה וצריך לבדוק את האפשרות שהערך של *הוא F לכן נריץ סריקת DFS על* ונסמן את העץ שקיבלנו ב*R. אם לא נמצא ב, נבצע את ההשמה T לכל קודקוד בקבוצה שנמצאה (ו-F לכל הקודקודים הנגדיים שלהם), אחרת מחזיר False.*

*נמשיך בתהליך זה בלולאה כל עוד קיימים קודקודים שלא עברו השמה (יכול להיות שהגרף לא קשיר) ואם הגענו לסוף הלולאה זה אומר שלא קיבלנו סתירות ומצאנו השמה מוצלחת לכן נחזיר True.*

נכונות האלגוריתם –

*בכל איטרציה על צומת v שעדיין אין לו השמה אנחנו מבצעים DFS כדי למצוא את כל הצמתים ברכיב הקשירות שלו כלומר את כל הליטרלים שנובעים מv, אם הגענו ל זה אומר שקיבלנו סתירה פסוקית ואם לא זה אומר שאין בתוצאת הDFS ליטרל והיפוכו (נניח בשלילה שקיבלנו וגם זה אומר שלפי טרנזיטיביות וגם לכן ולבסוף ). אבל גם המסלולו ההפוך יכול להתקיים לכן במקרה של סתירה בהרצה על הצומת v נריץ על הצומת ואז אם שוב קיבלנו סתירה אין השמה אפשרית ונחזיר False ואם לא קיבלנו השמה אפשרית ונמשיך באיטרציה.*

*זמן ריצה –*

*בבניית הגרף G עוברים על כל הליטרלים וכל הפסוקיות סך הכל .*

*זמן ריצת הDFS הוא גם (במקרה הרע מריצים פעמיים אבל זה לא משפיע על הסדר גודל של הזמן ריצה).*

סה"כ - .

**שאלה 4**

*נחלק את המסלול ל-5 שלבים* כאשר  *(X קבוצת קודקודים מתוך V שאינם קודקודים מועדפים, לא s ולא t)* מה שיבטיח ש ונשתמש בBFS בשביל למצוא את המסלול הקצר ביותר.

**אלגוריתם –**

*נבנה מ גרף מכוון חדש באופן הבא:*

*כל קודקוד מועדף , נוסיף פעמיים ל כך , . נוסיף את s וt לגרף החדש ולכל קודקוד ניצור בגרף החדש את .*

* *לכל קשת ,*
* *לכל קשת ,*
* *לכל קשת ,*
* *לכל קשת ,*
* *לכל קשת ,*
* *לכל קשת ,*
* *לכל קשת ,*

*כעת נריץ סריקת BFS על הגרף החל מ s. עבור ההופעה הראשונה של t בסריקה, המסלול מs עד t(לאחר הורדת התיוגים בשביל שיהיה מסלול בG) הוא המסלול הקצר ביותר שמקיים את תנאי השאלה. אם לא נמצא t בסריקה – אזי לא קיים מסלול כזה.*

**נכונות האלגוריתם -**

- באמצעות הגדרת הגרף G' אנחנו מבטיחים שכל מסלול מs לt יעבור בדיוק פעמיים *בקדקודים שלU (אפשר לראות זאת לפי תיוג הקודקודים לשלבים 1-5 והקשתות שתמיד מתקדמות בשלבים בלי אפשרות לדלג על שלבים 2 ו4 שבהם יש קודקודים של U).*

*המסלול הקצר ביותר- שימוש בסריקת BFS מבטיח קבלה של המסלול הקצר ביותר , מכיוון שהסריקה מתבצעת ברמות כאשר קודקוד שהופיע ברמה מוקדמת – לא יופיע ברמה מאוחרת יותר.*

*לבסוף מורידים את התיוגים בשביל להבטיח שהמסלול נמצא בG.*

סיבוכיות האלגוריתם -

בניית הגרף G’ מתבצעת ע"י מעבר על כל הקודקודים והקשתות של G פעם אחת ולכן רצה ב , סריקת BFS מעל G’ רצה ב (לפי זמן ריצה של BFS) וקיבלנו שזמן הריצה הכולל הוא .