**ממ"ן 11 – קורס אלגוריתמים – האו"פ**

**מגיש:** יוחאי מזוז

**ת"ז:** 324962125

1. אשתמש ברדוקציה לבעיית השידוך היציב,

**תיאור האלגוריתם:**

אשתמש באלגוריתם השידוך היציב(G-S), על מנת לשדך כל ספינה לנמל שבה היא תעגון ותישאר למשך החודש, נשים לב לכך שלכל נמל אכן מותאמת ספינה אחת בדיוק שתעצור למשך החודש אצלו, ואת רשימת ההעדפות של כל נמל וספינה נבנה כך –

כל ספינה תעדיף להיות "משודכת" לנמל שהיא מבקרת בו קודם(כך שהנמל הכי מועדף יהיה הראשון בו תבקר) והאינטואיציה לכך היא שכך אנו לא מבקרים נמלים "סתם" ולא חוסמים לספינות אחרות את האפשרות להישאר בנמל(על ידי זה שנצטרך לבקר בו).

וההגדרה המתמטית הוא שבהינתן ספינה s המבקרת את הנמלים בימים בהתאמה, אם אז הספינה s מעדיפה את על פני .

כל נמל יעדיף להיות "משודך" לספינה שמבקרת מאוחר יותר(כך הספינה שתבקר בו אחרונה תהיה המועדפת ביותר) והאינטואיציה לכך היא שלא נחסום ספינות מלבקר בנמל בכך שתעצור בו ספינה שמבקרת בו מוקדם יותר.

ההגדרה המתמטית להעדפה הזו היא שבהינתן נמל וספינות המבקרות בו בימים בהתאמה, אם אז הנמל יעדיף את על פני .

**מבנה האלגוריתם:**

ראשית נגדיר את רשימת ההעדפות לכל נמל וספינה:

רשימת העדפות של כל ספינה:

ניצור מערך דו-מימדי S (נניח לצורך האלגוריתם שמערכים מתחילים מ1 ולא מ0 - סמנטיקה) באורך ורוחב כך שהשורה ה היא רשימת ההעדפות של הספינה ה(בסדר יורד).

לכל נגדיר את השורה ה במערך S כך:

נרוץ על הלו"ז בשורה של הספינה ה מ עד

וכל פעם כשמופיע נמל נוסיף אותו לשורה ונתקדם צעד בשורה(כאשר בהתחלה המיקום בשורה הוא 1(הראשון)).

רשימת העדפות של כל נמל:

באותה צורה, ניצור מערך דו-מימדי P (נניח לצורך האלגוריתם שמערכים מתחילים מ1 ולא מ0 - סמנטיקה) באורך ורוחב כך שהשורה ה היא רשימת ההעדפות של הנמל ה(בסדר יורד).

לכל נגדיר את השורה ה במערך P כך:

נרוץ על הלו"ז בשורה של הספינה ה מ עד (מהסוף להתחלה כדי להעדיף ספינות מאוחרות)

וכל פעם כשמופיעה ספינה נוסיף אותה לשורה ונתקדם צעד בשורה(כאשר בהתחלה המיקום בשורה הוא 1(הראשון)).

ולאחר מכך נפעיל את אלגוריתם השידוך היציב(G-S) כקופסה שחורה עם מפות ההעדפות P וS.

**נכונות האלגוריתם:**

לאחר הפעלת האלגוריתם קיבלנו n זוגות שבכל אחד נמל וספינה שתעגון ותישאר בו, כעת נוכיח שלא יכול להיווצר מצב שבו התנאי ◇ (מעוין) מתקיים, נעשה זאת באמצעות הוכחה בשלילה:

נניח שלאחר ביצוע האלגוריתם,

נניח שהספינה עוצרת בנמל ביום למשך החודש, והספינה מבקרת בנמל ביום כך ש וזאת השלילה של התנאי ששתי ספינות לא יכולות עגון באותו יום באותו נמל(משום שהן נמצאות בנמל יחדיו ביום ).

וכעת משום שביצענו את האלגוריתם נגדיר את להיות הנמל בו הספינה צריכה לעצור למשך החודש כפי שהתקבל מהאלגוריתם.

אם כך התקבלו שני הזוגות הסדורים("השידוכים"), .

משום שהספינה עוגנת בנמל בהכרח מתקבל כי היא עוצרת בנמל לאחר יום (כי אם הייתה עוצרת לפני, הייתה נשארת שם ולא מבקרת את ),

ולכן לפי הגדרת רשימת העדפות, מתקבל ש עוצרת ב לפני ולכן היא מעדיפה את על פני .

ומשום ש עוגנת ב לאחר אז לפי הגדרת רשימת העדפות, מעדיף את על , משמע קיבלנו שהזיווג הוא לא יציב כי בזוגות הסדורים , מעדיף את ולהיפך.

וזאת סתירה לאלגוריתם הזיווג היציב G-S (טענה 1.6) ומכאן שקיבלו סתירה להנחה שלנו ולכן תנאי השאלה מתקיים והאלגוריתם נכון.

**ניתוח סיבוכיות זמן ריצה:**

בניית רשימת העדפות לספינות:

בניית רשימת העדפות לנמלים:

אלגוריתם הזיווג היציב G-S (1.3)

סך הכול: (משום ש)

סיבוכיות זמן הריצה של האלגוריתם הוא

וסיימנו.

1. **תיאור האלגוריתם:**

הכוונת גרף שכזה תתאפשר רק אם יש מעגל בכל רכיב קשירות בגרף, ואם יש מעגל בכל רכיב קשירות בגרף נכווין את הגרף על ידי זה שנכווין את המעגל בכיוון כלשהו, ואת הקשתות מחוץ לגרף נכוון כך שהן יוצאות מהקודקוד שקרוב למעגל לכיוון הקודקוד הרחוק.

**מבנה האלגוריתם:**

V – קודקודים, E - קשתות.

נאתחל T גרף ריק שייציג את רכיב הקשירות אשר אנו מכווינים.

וגרף V' מאותחל ריק של קודקודים שכבר היו ברכיב קשירות שהוכוון.

לכל :

אם אז continue(כבר ברכיב הקשירות שהוכוון)

אחרת:

נבצע סריקת DFS ונכניס את התוצאה לT,

נוסיף את הקודקודים ל,

אם כל הקשתות בT הן קשתות עץ(אין קשתות חוזרות) אזי אין מעגל ברכיב הקשירות ולכן לא ניתן להכווין ונחזיר false באלגוריתם וסיימנו.

אחרת קיימת קשת חוזרת

נכוון אותה ל,

ניגש לכל הקשתות הלא מכוונות שיוצאות מ ונכוון אותן החוצה(מv אל הקודקוד השני), ובאופן רקורסיבי נבצע את זה גם על הקודקוד השני בעזרת BFS.

לבסוף לאחר שרצנו על כל הקודקוד בגרף נחזיר true

**נכונות האלגוריתם:**

נתחיל מזה שנוכיח שאם ברכיב קשירות אין מעגל לא ניתן לכוון אותו: נניח שיש רכיב קשירות בלי מעגל שניתן לכוון אותו(כך שדרגת הכניסה של כל קודקוד גדולה מ0), נבחר קודקוד u, משום שדרגת הכניסה גדולה מ0, יש קודקוד שיש לו קשת מוכוונת אליו, נלך לקודקוד הזה, את אותו הדבר ניתן להגיד גם על הקודקוד הזה, ניתן לבצע את התהליך הזה איזה מספר פעמים שאנחנו רוצים, ובפרט לאחר V+1 (V זה מספר הקודקודים ברכיב הקשירות)פעמים נקבל מסלול שמספר הקודקודים בו גדול ממספר הקודקודים ברכיב הקשירות, ומעיקרון שובך היונים, היה קודקוד שחזר פעמיים ומכאן שזה מעגל וזו סתירה להנחה שברכיב הקשירות אין מעגל.

כעת נוכיח את האלגוריתם:

לכל רכיב קשירות בגרף, האלגורייתם מכוון כל קודקוד במעגל לאחד משכניו, כך שכל המעגל הוא מסלול אחד מכוון בכיוון אחד, ולכן בפרט לכל קודקוד במעגל יש שכן שמכווין אליו ולכן דרגת הכניסה שלו גדול המ0.

ועבור קודקוד שלא במעגל , משום שזה רכיב קשירות, יש לו מסלול למעגל, ומשום שהכיוון הוא מהמעגל החוצה, אזי הקודקוד שנמצא אחריו במסלול v(זה שקרוב יותר למעגל) מכוון לכיוון אותו u, ולכן דרגת הכניסה של v גדולה מאפס.

**סיבוכיות זמן ריצה**:

ריצת הDFS רצה בפרט בכל רכיב קשירות לכל צומת וקשת, אך בסך הכל היא נעשית ריצה על כל הקודקוד והקשתות(כי סכום מספר הקודקודים והקשתות בכל רכיבי הקשירות שווה למספר הקודקודים והקשתות בגרף),

שווה לזמן ריצה של , באותה צורה גם ריצת הBFS היא

אך משום שיתכן זמן ריצה בו האלגוריתם מחזיר false ולא רץ על כל הגרף נקבל כי סיבוכיות זמן הריצה היא

1. **תיאור האלגוריתם:**

נשתמש בזהות , ונבנה גרף מכוון שקודקודיו הם הליטרלים ושלילתם וניצור קשת מכוונת מהליטרל לליטרל בכיוון זה אם ישנה פסוקית ששקולה ל ) וכך גם עם כל הווריאציות של שלילת ליטרלים), ומכאן נמצא רכיב קשירות והוא ייצג קבוצה של פסוקיות שנגדיר לה השמה, ובצורה הזו נוכל "לחסוך" בדיקה של כל אפשרות בכך שנגדיר ליטרל אחד וממנו נגרור את ההשמה של כל הליטרלים האחרים שנמצאים ברכיב הקשירות שלו תוך בדיקה שהוא לא מוביל לסתירה.

**מבנה האלגוריתם:**

נבנה גרף מכוון בצורה הבאה:

* לכל ליטרל נבצע .
* לכל פסוקית
* נבצע (הקשת מכוונת מהקודקוד השמאלי לימני)
* *כל עוד קיים קודקוד שלא עבר השמה:*
* *נבחר קודקוד כזה ונבצע סריקת DFS ממנו שאת תוצאתה נסמן K,*
* *אם אז נבצע השמה true בכל הקודקודים שבK וfalse בכל הקודקודים הנגדיים להם.*

*אחרת:*

*נבצע סריקת DFS מ שאת תוצאתה נסמן L,*

*אם אז נבצע השמה true בכל הקודקודים שבL וfalse בכל הקודקודים הנגדיים להם.*

*אחרת נחזיר false(לא קיימת השמה מספקת.*

* *נחזיר true ואת הקודקודים עם ההשמה.*

***נכונות האלגוריתם:***

*בכל איטרציה על קודקוד שאין לו השמה, אנחנו משתמש בDFS כדי למצוא את כל הקודקודים ברכיב הקשירות שלו, אם נקבל את כבן של אז נקבל סתירה במסלול הפסוקי(ולכן לא ניתן לבצע השמה).*

*אבל ההיפך אפשרי, זאת אומרת שניתן לבצע השמה שמתבססת על וזאת בתנאי שלא קיים מסלול מ ל, לכן נבצע DFS מ , אם כן קיים זאת אומרת שהצמתים נגישים הדדית ותמיד תיווצר סתירה במסלול הפסוקי ולכן לא קיימת השמה ונחזיר false, כי אז מבחינה לוגית מתקיים , שזה אומר מבחינת הנוסחה שלנו שבכל הצבה שנשים ל נגיע לפסוקית שבה נצטרך "להפוך" את ההשמה ששמנו ב ולכן לא קיימת השמה מספקת.*

*אבל אם אחד המסלולים לא קיים אז נבצע השמה לפי אותו קודקוד( או )*

*ואם הגענו למצב בו יש השמה לכל צומת ואין סתירות אזי מצאנו השמה מספקת ונחזיר true ואת ההשמה.*

***ניתוח סיבוכיות זמן ריצה:***

*בניית הגרף:*

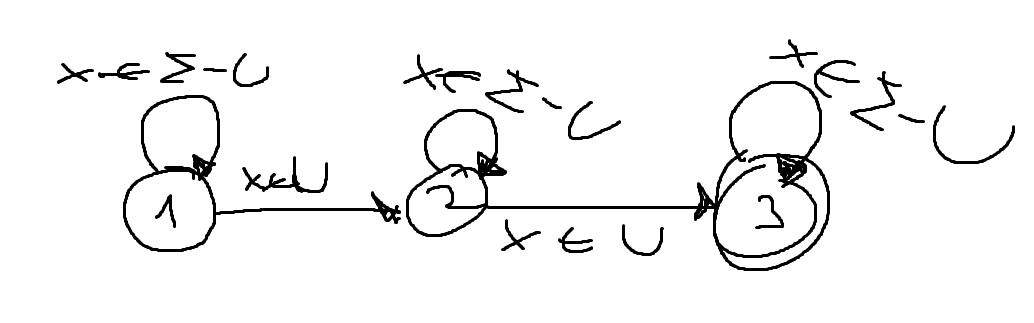
ריצה בלולאה כדי לבצע השמה: בכל מקרה אנחנו נבצע DFS במקסימום פעמיים על כל רכיב קשירות ולכן משום שסכום מספר הקודקודים בכל רכיבי הקשירות שווה לסכום הקודקודים בגרף אזי לכל היותר נרוץ פעמיים על כל הקודקודים ולכן זמן הריצה הוא

סך הכל זמן הריצה הוא ,

*ולכן נקבל שסיבוכיות זמן הריצה של האלגוריתם היא .*

1. ***תיאור האלגוריתם:***

*השאלה דומה לאוטומט סופי דטרמיניסטי שצריך לקבל קלט מסוים בדיוק פעמיים, ולכן נפתור בצורה דומה, זה האוטומט שהשאלה מקבילה אליו:*

**

*ולכן נבצע רדוקציה לגרף מכוון חדש שמקביל אל השלבים של האס"ד עם שינויים קלים כדי להתאים לגרף, ולבסוף נמצא את המסלול הקצר ביותר באמצעות שימוש באלגוריתם BFS.(נשים לב שקבלת הקלט במצב המקבל האחרון היא סטנדרטית אבל לא נתייחס אליה באלגוריתם זה משום שאנו מחפשים את המסלול הקצר ביותר, ולכן היא לא תופיע בגרף)*

***מבנה האלגוריתם:***

*נגיד*

*נבנה גרף מכוון בהתבסס על הגרף הנתון:*

* *לכל קודקוד , נבצע*
* *לכל קודקוד , נבצע*
* *לכל נבצע*
* *לכל נבצע*
* *לכל נבצע*
* *לכל נבצע*
* *לכל נבצע*
* *לכל נבצע*
* *לכל נבצע*
* *לכל נבצע*
* *לכל נבצע*
* *לכל נבצע*

*כעת נריץ סריקת BFS על גרף החל מ, ועבור ההופעה הראשונה של t בסריקה, זה המסלול הקצר ביותר מs עד t שמקיים את תנאי השאלה, אם אין הופעה של t אזי אין מסלול המקיים את תנאי השאלה.*

***נכונות האלגוריתם:***

*תנאי השאלה –*

*השאלה דורשת שהמסלול יקיים ובכן נראה למה זה נכון:*

*בדומה לאס"ד שציירתי למעלה, ישנם 5 שלבים שיש לעבור כדי להגיע מs לt בגרף החדש, כאשר ביניהם צריך לעבור בדיוק פעמיים בקודקוד מ, וניתן לדלג בין שלבים אך ורק מ למעבר פעם ראשונה בU, מעבר מה"ביקור" הראשון בU לשני, ומהשני לt, אך כאשר מתקדמים במסלול כל עוד לא ביקרת ב u אתה לא יכול להתקדם לשלב הבא ובעצם כך כל עוד לא ביקרת פעמיים בu לא תוכל להתקדם לשלב בו תוכל להגיע לt*

*והמסלול הוא המסלול הקצר יותר משום שאנחנו משתמשים באלגוריתם BFS המבטיח קבלה של המסלול הקצר ביותר.*

*והמסלול שמצאנו קיים בG על ידי הסדרה של התיוגים ומספור השלבים(מקביל להומומורפיזם) וכך אנחנו מקבלים את המסלול "המקורי", לחילופין הוספה של תיוגים ומספרים היא אפשרית גם למסלול המקיים את התנאי בG לפי השלב שבו כל קודקוד נמצא ביחס לביקור הראשון והשני בU וכך נקבל את המסלול המתאים ב.*

***סיבוכיות האלגוריתם:***

*בניית בגרף מתבצעת על ידי מעבר על הקודקודים והקשתות ולכן*

*סריקת BFS על :*

*סהכ נקבל כי סיבוכיות זמן הריצה של האלגוריתם היא*

*תודה רבה על הבדיקה ושבוע טוב*