קורס 20417 סמסטר 2017ב

אודי ליוטשי 036745438

ממ"ן 11

1) הבעיה שקולה לבעיית השידוך היציב הנדונה בפרק 1 בספר, נסביר:
על פי דרישות החברה ונתוני השאלה (מספר הספינות והנמלים שווה, במצב הסופי כל ספינה עוגנת בנמל ואסור ששתי ספינות ישהו בנמל באותו זמן), במצב הסופי כל ספינה תעגון בנמל אחר ובכל נמל תעגון ספינה, כלומר, יחס חח"ע ועל בדומה לבעיית השידוך היציב.
לכל לו"ז עלינו לבחור נמל עצירה לתחזוקה, כלומר, נמל העצירה מגדיר את קיצוץ הלו"ז של ספינה מסוימת.
פתרון קביל לבעיה הוא כזה שמתאים לכל ספינה נמל עצירה כך שתנאי איסור השהייה המשותפת בנמל לא יופר.

אם כך נשדך בין הספינות והנמלים באופן הבא:
נסמן ב-S את קבוצת הספינות וב-P את קבוצת הנמלים.
רשימת ההעדפות של ספינה תהיה מורכבת מכל הנמלים המופיעים בלוח הזמנים שלה ומסודרת לפי סדר ההגעה לנמלים מהראשון לאחרון.
רשימת ההעדפות של נמל תהיה מורכבת מכל הספינות שאותו נמל מופיע בלוח הזמנים שלהן ומסודרת בסדר הפוך לסדר הגעתן, כלומר, עדיפות גבוהה יותר לספינה שאמורה לעגון במועד מאוחר יותר.

לפי האלגוריתם לשידוך יציב כל ספינה "תבחר" את הנמל הראשון בסדר העדיפות שלה שטרם פנתה אליו, אם הוא אינו משודך אז ישנו שידוך ביניהם (על משקל אירוסין), אם הוא משודך לספינה אחרת, הנמל בוחר אם לשנות את השידוך הקיים לפי רשימת ההעדפות שלו.
בסיום התהליך, כשלא נשארו ספינות שאינן משודכות, השידוך הופך קבוע (על משקל נישואין).

הוכחת נכונות האלגוריתם:
נניח בשלילה שהפתרון המתקבל מהאלגוריתם אינו קביל, כלומר, הוא מפר את כלל איסור השהייה המשותפת, כלומר, ישנה ספינה s1 שעוברת דרך נמל p1 בזמן שספינה אחרת s2 שוהה בו.
במקרה זה s1 מעדיפה את p1 על פני נמל העצירה שלה (מכיוון שהיא מגיעה אליו במועד מוקדם יותר). כמו כן, נמל p1 מעדיף את s1 על פני s2 (מכיוון שהיא מגיעה אליו במועד מאוחר יותר). קיבלנו שידוך לא יציב שבו s1 מעדיפה את p1 על פני נמל העצירה שלה ונמל p1 מעדיף את s1 על פני הספינה שעצרה בו.
זוהי סתירה לעובדה שהאלגוריתם לשידוך יציב מפיק בהכרח שידוך יציב ולכן מצב שכזה אינו יכול להתרחש, כלומר, הפתרון שהתקבל קביל בהכרח.

זמן ריצה:
זמן סריקת לוחות הזמנים יהיה מכפלת מספר הספינות במספר הימים: θ(m*n)
זמן ריצת אלגוריתם השידוך היציב : O(n^2)
סה"כ נקבל: O(n^2+m*n) = O(m*n) m>n

2) נצמצם את הפתרון לגרף קשיר, קל מאוד יהיה להכליל אותו לגרף לא קשיר ע"י מציאת פתרון לכל רכיב קשירות בנפרד. הגרף כולו יהיה ניתן לכיוון אם כל רכיבי הקשירות ניתנים לכיוון.

אלגוריתם: נבחר קודקוד באופן אקראי ונריץ עליו DFS, נקבל עץ עם קשתות עץ שמוגדרות כקשת מקודקוד אב לבנו וקשתות חוזרות שמוגדרות כקשת שאנה קשת עץ.

אם לא קיבלנו לפחות קשת חוזרת אחת נסיק שלא ניתן לכוון את הגרף לפי הנדרש מכיוון שבעץ ישנם n קודקודים ורק n-1 קשתות.

אם קיבלנו קשת חוזרת, נניח בין קודקודים x ו- y, נכוון אותה כך (x,y), כעת ניקח את קודקוד y ונריץ עליו BFS, כעת נכוון את כל הקשתות שטרם כוונו מהרמה הקרובה ל-y לכיוון הרמה הרחוקה מ-y.

נכונות: מכיוון שהקשתות מכל רמה שהתקבלה מה-BFS מכוונות לרמה הרחוקה יותר, מובטח לנו שדרגת הכניסה של כל קודקוד תהיה גדולה מ-1 פרט לרמה מספר 0 שבה נמצא קודקוד y, אך גם רמת הכניסה של קודקוד y גדולה מ-0 כי קיימת הקש (x,y) . אם כך מובטח לנו שלכל קודקוד יש לפחות קשת אחת שמוכוונת אליו , כלומר, דרגת כניסה גדולה מ-0 כמבוקש.

זמן ריצה: DFS+BFS = O(m+n). m- מספר צלעות, n- מספר קודקודים.

3) ניקח כל פסוקית בנפרד ונמיר אותה ל- 2 קשרים של גרירה. לדוגמה במקום $x1 \vee x2$ ניתן לכתוב 2 פסוקיות גרירה: (¬x1 → x2) ∧ (¬x2 → x1) . שתי הצורות שקולות לוגית.
כעת נבנה גרף מכוון G כך: הצמתים יהיו הליטרלים של הפסוקים השקולים לאחר המרה, הקשתות מייצגות את קשר הגרירה (לפי הדוגמא לעיל נקבל גרף עם 4 צמתים: {x1,¬x1,x2,¬x2} ו-2 קשתות: {(¬x1,x2), (¬x2,x1)}).

מטרת האלגוריתם הינה למצוא השמה לכל הליטרלים כך שלא תהיה סתירה בין ליטרל ושלילתו.
האלגוריתם:
בוחרים משתנה x באופן שרירותי ומריצים עליו DFS.
אם ¬x אינו ישיג מ-x *נציב ערך אמת ב-x ובכל ליטרל שהתקבל מה-DFS (או ערך שקר בליטרל עם שלילה). ברור שאין סתירה במקרה זה בין x ושלילתו.

אם ¬x ישיג מ-x נריץ שוב DFS אבל הפעם מ- ¬x. אם x ישיג מ-¬x אז קיבלנו סתירה לוגית ולכן הנוסחה אינה ספיקה ונחזיק תשובה שלילית.
אם x אינו ישיג מ- ¬x נפעל באופן דומה ל-*.

כעת נחזור על התהליך על כל הליטרלים שטרם נקבעה להם השמה.

הוכחת נכונות:
לכאורה אנו עלולים להיתקל במצב שבו טיפול בליטרל מסוים חסר השמה y דורש השמה של z השונה מההשמה הקיימת של z שנקבעה במהלך טיפול בליטרל אחר x.
זה יקרה במידה ו-z ישיג מ-x וגם ¬z ישיג מ-y ואז נצטרך להציב ערך שקר ב-z בסתירה לערך האמת שהושם לו קודם לכן.
אם ¬z ישיג מ-y אז גם ¬y ישיג מ-z לפי כלל ההמרה של הפסוקית.
אזי, אם z ישיג מ-x וגם ¬y ישיג מ-z נקבל לפי טרנזיטיביות ש- ¬y ישיג מ-x, כלומר, לאחר הטיפול ב-x, y כבר קיבל השמה בסתירה להנחה ש-y חסר השמה.
אם כן ראינו שלא תתכן סתירה בהרצת DFS על ליטרלים שונים וכן סתירה המתקבלת בהרצת DFS על ליטרל מסוים מטופלת ע"י החזרת תשובה שלילית ומכאן שהאלגוריתם נכון.

זמן ריצה:
עבור m פסוקיות בנוסחה φ ומספר קבוע c של ליטרלים בכל פסוקית : θ(c*m)= θ(m)
נבנה את הגרף עם מקסימום 2c צמתים ו -2m קשתות: O(2n+2m)
נריץ את DFS בזמן ריצה $O(2m + 2n)$
סה"כ: $O(m + n)$

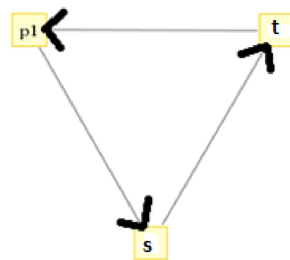
4) בהינתן גרף G נבנה גרף חדש `G שבו נסיר את כל הצלעות המכוונות לקודקודים המועדפים.
כעת נשרשר 3 גרפים כאלה G`1, G`2,G`3 באופן הבא: לכל צלע שהייתה מכוונת לקודקוד מועדף בגרף המקורי נייצר צלע המכוונת מקודקוד המקור בגרף G`1 לקודקוד היעד שלה ב-G`2 וכן מ-G`2 ל-G`3.

כמו כן נוסיף 2 קודקודים s`,t` ונייצר את הצלעות הבאות (s`,s) בין s` ל-s בגרף G`1 ו-(t,t`) בין t בגרף G`3 ל- t`.

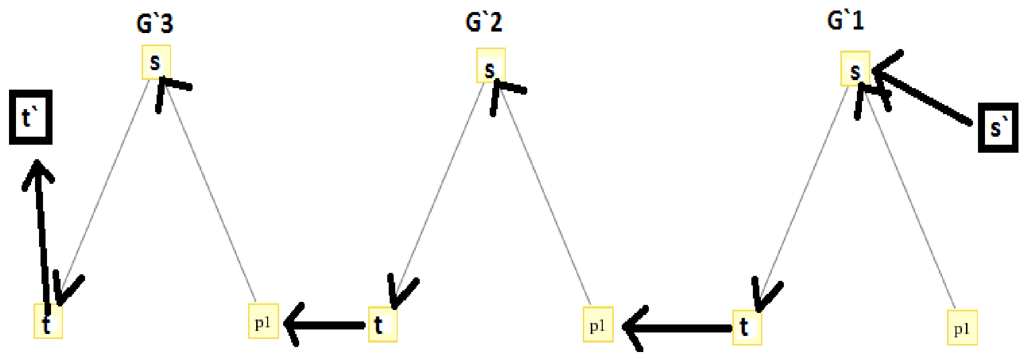
לדוגמא:

G = ({s,t,p1},{(s,t),(t,p1),(p1,s)})

כך ש: p1ϵU



G` = ({s,t,p1},{(s,t),(p1,s)})



כפי שניתן לראות על מנת להגיע לקודקוד מועדף נאלץ לעבור מגרף אחד למשנהו. ניתן לבצע 2 מעברים בלבד מה שמבטיח ביקור בקודקוד מועדף פעמיים בדיוק כנדרש.

כעת נריץ BFS מ-s` ל-t` ונקבל את המסלול הקצר ביותר ביניהם. נראה שהמסלול שקיבלנו לאחר הסרת הצמתים s` ו-t` מכיל שני מופעים של קודקוד מועדף והוא גם הקצר ביותר בין s ל-t בגרף המקורי G:
נאחד את שלושת ההעתקים של G` על ידי העתקת כל הצלעות משלושת הגרפים לגרף המקורי G תוך התעלמות משייכות הקודקודים לגרף מסוים.
קל לראות שקיימת התאמה חח"ע ועל בין המסלולים בגרף החדש והגרף המקורי שהרי לאחר הסרת s` ו-t` כל הצמתים במסלול קיימים ב-G וכל הצלעות במסלול קיימות גם הן ב- G.
לכן אם היה מסלול קצר יותר ב-G הכולל שני ביקורים בקודקודים מועדפים היינו מקבלים אותו בסריקת ה-BFS שהרצנו על הגרף החדש.

זמן ריצה:
בניית הגרף החדש: O(m+n) , מעבר על כל הקשתות והקודקודים, תהליך היצירה מוסיף מכפלה בקבוע.
זמן ריצת BFS: O(m+n), זמן זהה לריצה על G כי הראינו התאמה חח"ע ועל
סה"כ זמן ריצה: O(m+n)