**אלגוריתמים**

**ממ"ן 15 - תשובות**

מספר קורס: 20417

מנחה: אורן רות

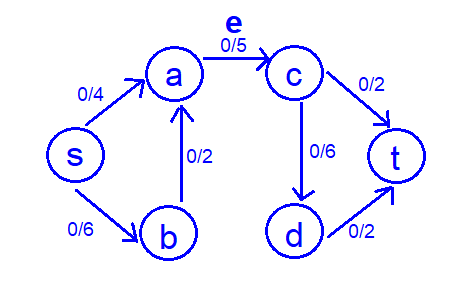
שם הסטודנטית: ברית בן-דוד

ת"ז: 204879365

**שאלה 1:**

**צלעות שמוסרות, נוספות מחדש, ושוב מוסרות מהרשת השיורית.**

להלן דוגמא פשוטה לרשת זרימה G המקיימת את שתי התכונות הנדרשות בשאלה:

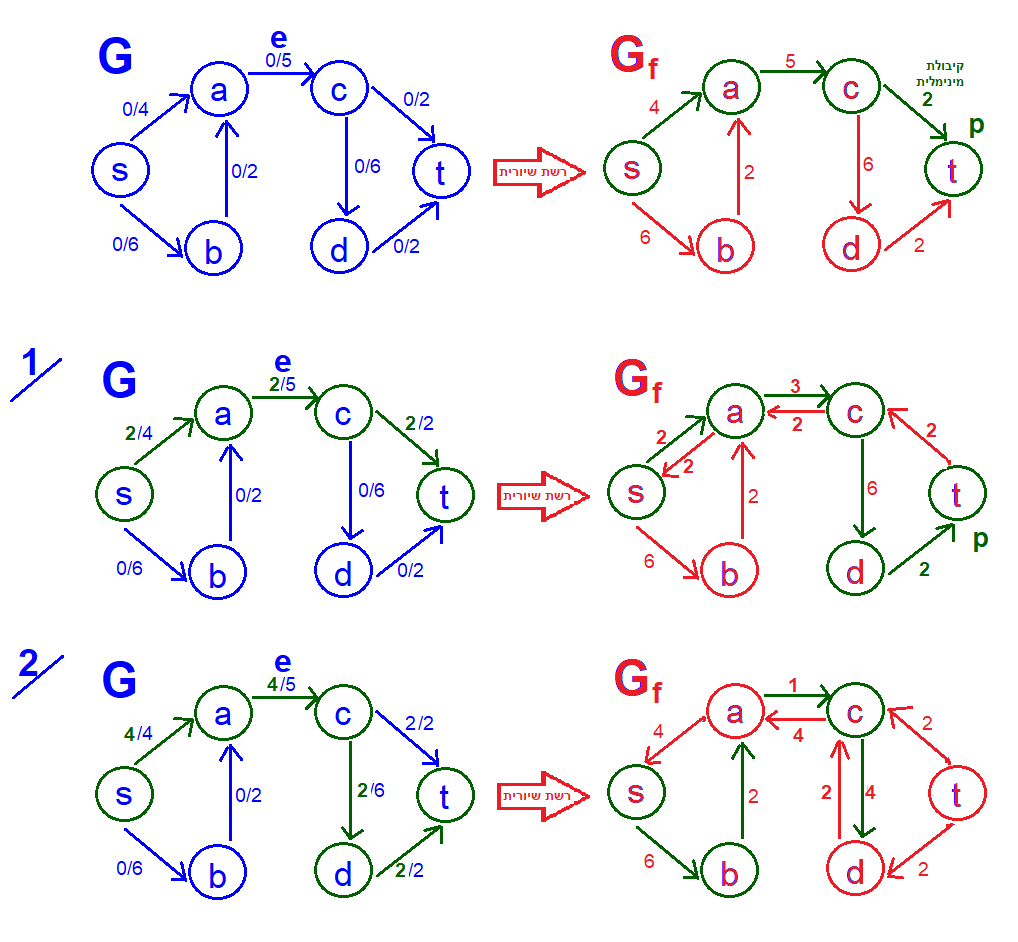


צלע e הנדרשת היא הצלע (a,c).

1. באיטרציה הראשונה של הרצת אלגוריתם אדמונס-קרפ יבחר ברשת השיורית המסלול הקצר ביותר s🡪a🡪c🡪t, הקיבולת המינימלית שניתן להזרים במסלול זה (צוואר הבקבוק) היא 2, לכן נעדכן את הגרף המקורי, כאשר ברור שבצלע e תעבור זרימה של 2 (זהה לקיבולת השיורית).
2. באיטרציה השנייה יבחר ברשת השיורית המסלול הקצר ביותר s🡪a🡪c🡪d🡪t, גם במסלול זה הקיבולת המינימלית תהיה 2, וזה מה שנזרים בגרף המקורי על הצלע e.
3. לאחר 2 איטרציות אלו לא נמצא מסלולים נוספים מ-s ל-t ולכן סיימנו.

הראנו כי רשת הזרימה G שבחרנו עונה על דרישות השאלה.

להלן ציור של הרצת אלגוריתם אדמונס-קרפ (לצורך המחשה בלבד):



**שאלה 2:**

**זרימה מזערית כשקיבולות חוסמות מלרע את הזרימה הנדרשת.**

1. טענה: אם קיימת זרימה חוקית ברשת אז קיימת ברשת זרימה חוקית גדולה כרצוננו.

הוכחה:

נוכיח באמצעות דרך השלילה.

נניח בשלילה כי אם קיימת זרימה חוקית ברשת אז לא קיימת ברשת זרימה חוקית גדולה כרצוננו, נסיק את הסתירה לנתון ומכך נראה שהמשפט מתקיים.

הנחנו כי **לא קיימת ברשת זרימה חוקית גדולה כרצוננו**, מהנחה זו ניתן להסיק כי מתקיימת אחת משתי אפשרויות:

1. לא קיימת כלל זרימה חוקית ברשת.
2. קיימת זרימה חוקית אחת ברשת, נסמן אותה כ- (זרימה בלעדית, אינה גדולה כרצוננו).

נפרט את המסקנה עבור כל אחת מהאפשרויות שהסקנו:

1. במקרה ובחרנו באפשרות זו, נגיע לסתירה באופן מידי. הנתון הוא כי קיימת זרימה חוקית ברשת וכאן אנו מסיקים שזרימה זו לא קיימת. לכן, ההנחה אינה נכונה והמשפט המקורי שרצינו להוכיח הוא המשפט הנכון.
2. במקרה ובחרנו באפשרות זו – קיימת זרימה חוקית אחת ברשת, סימנו אותה כ- . מנתוני השאלה ידוע כי לכל צלע ברשת מתקיים ובמילים אחרות כל קיבולת חוסמת את הזרימה דווקא מלמטה.

כעת, נבחר להזרים בכל צלע זרימה גדולה ב- מהזרימה שהזרמנו קודם לכן, ברור כי מתקיים עבור כל צלע: . זרימה : מקיימת את תנאי השאלה (גדולה מהקיבולת שהוגדרה עבור כל צלע) ולכן היא חוקית. בהתאם לכך, הזרימה הכוללת ברשת גם כן תגדל ב-y ונקבל את הזרימה הבאה: שהיא חוקית גם כן.

הראנו כי קיימות עבור רשת הזרימה יותר מזרימה חוקית אחת, ולכן הגענו לסתירה לאפשרות השנייה. מכאן, שגם עבור אפשרות זו המשפט המקורי אותו רצינו להוכיח הוא המשפט הנכון.

הראינו סתירה עבור 2 האופציות האפשרויות להנחת השלילה ולכן המשפט נכון, כנדרש.

1. מטרה: הצגת אלגוריתם למציאת זרימה חוקית (לאו דווקא מזערית) ברשת.

רעיון האלגוריתם:

נשתמש באלגוריתם פורד-פולקרסון כאשר נבצע בו שינוי בודד – השינוי יהיה שלאחר שמצאנו את הרשת השיורית ובחרנו במסלול שיפור כלשהו, נגדיר את "צוואר הבקבוק" – הזרימה שנרצה להעביר בגרף המקורי – לפי הקיבולת המקסימלית במסלול (הקיבולת הגבוהה ביותר של אחת מהקשתות השייכות למסלול), זאת בניגוד לזרימה שבחרנו לפי הקיבולת המינימלית באלגוריתם המקורי.

בנוסף, נבדוק בסוף ההרצה של האלגוריתם אם נותרו קשתות שלא נעלמו לנו (ז"א, לא ניצלנו את מקסימום הקיבולת שניתן להעביר עליהן) – במידה וקיימות קשתות כאלו, נבדוק מהי הזרימה החסרה ונמצא מסלול כלשהו (מהגרף המקורי) מ-s ל-t שהקשת שמצאנו היא חלק ממנו ונוסיף את הזרימה החסרה לאורך כל המסלול (על כל הקשתות). נבצע זאת, עד שלא נמצא מסלולים כלל ברשת השיורית (נעלמו כל הקשתות).

האלגוריתם:

1. לכל קשת בצע:
2. כל עוד קיים מסלול מ-s ל-t ברשת השיורית המושרית ע"י f, בצע עבור מסלול **p מסוים**:

   2. לכל קשת בצע:
3. נגדיר את הרשת השיורית הנותרת כ-GS.
4. כל עוד קיימת קשת כלשהי ברשת GS, בצע עבור קשת זו:
   1. מצא מסלול **p כלשהו** מ-s ל-t בגרף המקורי G המכיל את הקשת הזו (מתקיים: ).
   3. לכל קשת בצע:
   4. נמחק את הקשת מהרשת GS.

הוכחת נכונות:

1+2. סעיפים 1+2 זהים לאלגוריתם של פורד-פולקרסון המקורי, מלבד השינוי בהגדרת הקיבולת של מסלול השיפור הנבחר. מהוכחת אלגוריתם פורד-פולקרסון בשילוב נתוני השאלה – שכל קיבולת ברשת זרימה זו חוסמת את הזרימה דווקא מלמטה, ברור כי כל קיבולת שנגדיר, שגדולה מהקיבולת המינימלית לפי פורד-פולקרסון, תהיה נכונה, ולכן ברור כי חלק זה של האלגוריתם מתקיים.

1. נסמן את הרשת השיורית הנותרת כ-GS על-מנת שנוכל להשתמש בה בסעיפים הבאים.
2. מכיוון שעל כל צלע בגרף המקורי G עלינו להזרים כמות זרימה הגדולה או שווה לקיבולת המוגדרת עליה, ייתכן שלאחר שהזרמנו קיבולות מקסימליות דרך מסלולי שיפור רנדומליים, עדיין יישארו קשתות שלא הזרמנו עליהן את מינימום הקיבולת הנדרשת, ולכן נותר לטפל בהן. לכן נעבור על כל הקשתות שנותרו ברשת השיורית GS עד שלא יישארו קשתות כלל בגרף זה (ידוע שברגע שמזרימים על קשת מסוימת זרימה השווה לקיבולת, היא נעלמת, בצורה זו נעלים את כל הקשתות).
   1. אם מצאנו קשת כלשהי ברשת השיורית GE נמצא בגרף G מסלול שמכיל קשת זו, ברור שמסלול כזה קיים מנתוני השאלה.
   2. נגדיר את קיבולת המסלול כולו להיות הקיבולת שנותרה בקשת שמצאנו.
   3. נוסיף על כל אחת מהקשתות במסלול את הקיבולת שהגדרנו ובכך נממש 2 דרישות:

* נצליח להעלים את הקשת (x,y) ברשת השיורית הבאה במידה והיינו מחפשים רשת שיורית כזאת (כיוון שנזרים עליה זרימה שווה לקיבולת).
* בכל הקשתות האחרות במסלול p שבחרנו נזרים זרימה גדולה ב- מהזרימה שהייתה מוזרמת עד כה, מכיוון שקיימת בשאלה חסם מלרע של הזרימה שניתן להזרים, הזרימה החדשה שנזרים תהיה חוקית גם היא.
  1. 4.4. נמחק את הקשת כיוון שהיא מקיימת את דרישות השאלה ולכן אין צורך לטפל בה שוב.

לסיום, נראה כי כל הדרישות מרשת זרימה חוקית מתקיימות, לאחר ביצוע אלגוריתם זה:

* אילוצי קיבול: לכל קשת e קיים הקיבול המיוחס לו (הקיבול שהוגדר לו בתחילת השאלה).
* שימור זרימה: מכיוון שהוספנו זרימות עבור מסלולים שלמים ברור שעבור כל צומת יתקיים שכלל הזרימה הנכנסת שווה לכלל הזרימה היוצאת.
* אנטי סימטריות: גם כאן, כמו באלגוריתם פורד-פולקרסון המקורי, תכונה זו תשמר.

חישוב סיבוכיות זמן ריצה:

1+2. פורד-פולקרסון המקורי רץ בזמן של ברור ש-f באלגוריתם שלנו שונה מ- בפורד-פולקרסון כיוון שלא מדובר בזרימה המקסימלית. לכן באלגוריתם שלנו נשתמש ב- אותה נגדיר כזרימה המינימלית. לכן שלבים 1 ו-2 יבוצעו בזמן ריצה של .

1. זמן קבוע.
2. את סעיף 4 נריץ כמספר הפעמים של הקשתות שנותרו ב-GS - (GS=(ES,VS, כאשר ברור כי . תתי הסעיפים בסעיף 4 דומים מאוד לאלגוריתם פורד-פולקרסון מלבד השוני בהגדרת הקיבולת ולכן יורצו סה"כ בזמן של .

לכן, סה"כ זמן הריצה יהיה .

1. מטרה: הצגת אלגוריתם למציאת זרימה חוקית **מזערית** ברשת.

רעיון האלגוריתם:

כמו ההבדל בין האלגוריתם של פורד-פולקרסון לבין האלגוריתם של אדמונס-קרפ, שהראשון מוצא זרימה כלשהי והשני מוצא את הזרימה המזערית. גם כאן הרעיון זהה, ולכן עבור מציאת זרימה חוקית מזערית נשתמש באלגוריתם של אדמונס-קרפ, כאשר השינויים במימוש שלו יהיו זהים לשינויים שביצעו בסעיף ב' עבור פורד-פולקרסון (בחירת הקיבולת המקסימלית ברשת השיורית, במקום המינימלית, ומעבר בסוף על כל הקשתות שנותרו ללא טיפול).

האלגוריתם:

האלגוריתם זהה לאלגוריתם בסעיף ב', מלבד 2 שינויים:

* סעיף 2 – כל עוד קיים מסלול מ-s ל-t ברשת השיורית המושרית ע"י f, בצע עבור מסלול **p שהוא המסלול הקצר ביותר (לאחר הרצת BFS)**...
* סעיף 4.1. – מצא מסלול **p קצר ביותר (לאחר הרצת BFS)** מ-s ל-t בגרף המקורי G המכיל את הקשת הזו (מתקיים: ).

הוכחת נכונות:

מאותה סיבה שהראנו שפורד-פולקרסון יעבוד עם השינוי הנ"ל גם אדמונס-קרפ יעבוד עם השינוי הנ"ל.

חישוב סיבוכיות זמן ריצה:

גם כאן, הסיבוכיות זהה לסיבוכיות האלגוריתם המקורי של אדמונס-קרפ מכיוון שהתוספת של מעבר על הקשתות שנותרו ברשת השיורית GE קטנה מהסיבוכיות של הסעיפים הקודמים לה. לכן, הסיבוכיות כולה תהיה .

**שאלה 3:**

**תיקון זרימה מרבית נתונה.**

מטרה:

1. מציאת אלגוריתם שרץ בסיבוכיות זמן לינארית ומוצא זרימה מרבית ברשת, המתקבלת מהגדלת הקיבולת של ב-1.
2. מציאת אלגוריתם שרץ בסיבוכיות זמן לינארית ומוצא זרימה מרבית ברשת, המתקבלת מהקטנת הקיבולת של ב-1.

**ראשית, זה לא כתוב חד-משמעית אבל כן כתוב תיקון, לכן אני מניחה שהגדלת/הקטנת הקיבולת של ב-1 מתרחשת לאחר שסיימנו להריץ על רשת הזרימה אלגוריתם למציאת הזרימה המרבית.**

1. האלגוריתם:
2. נוסיף לקיבולת של 1, בהתאם לנדרש ונקבל: . בהתאם לכך נקבל גרף חדש שנגדיר אותו G’ (גרף זה מורכב מהגרף ממנו התחלנו, בו מצאנו קודם לכן את הזרימה האופטימלית, בתוספת 1 לקיבולת של ).
3. נבנה רשת שיורית מהגרף החדש שקיבלנו G’.
4. נחפש מסלול p כלשהו מ-s ל-t (בגרף השיורי) שמכיל את הצלע (ברור כי אם קיים מסלול שיפור כזה ניתן להזרים לאורכו לפחות 1).
   1. אם מצאנו מסלול כזה:
      1. לכל קשת נזרים עוד 1: .
      2. נעדכן את הזרימה הסופית להיות הזרימה שמצאנו קודם לכן +1 : . ונחזיר את f’ (הזרימה העדכנית).
   2. אם לא מצאנו מסלול כזה – נחזיר את f (הזרימה המקסימלית לא תשתנה).

הוכחת נכונות:

סעיפים 1 עד 3 התקיימו, מכיוון שמדובר בפעולות מוכרות (אנו מבצעים אותן גם באלגוריתם פורק-פולקרסון), ואין מניעה שהן התקיימו.

נראה כי בכל מקרה האלגוריתם ייעצר:

במידה ונמצא מסלול p שמקיים את מה שדרשנו בסעיף 3, נעדכן את הזרימה של כל אחת מהצלעות במסלול, וכן את הזרימה הסופית (תוספת של 1 לכל זרימה), נחזיר את הזרימה המרבית ברשת, כפי שנדרשנו למצוא, ונסיים.

במידה ולא נמצא מסלול p כזה, נחזיר את הזרימה שמצאנו קודם לכן, אמנם הגדלת הקיבולת של לא סייעה להגדיל את הזרימה של הרשת ולכן הזרימה המרבית תישאר הזרימה שמצאנו קודם לכן, נחזיר זרימה זו ונסיים.

בכל מקרה האלגוריתם יסיים את ריצתו ויחזיר את התוצאה הנכונה המתאימה.

ניתוח זמן ריצה:

1. הגדלת קיבולת לצלע - .
2. ידוע כי בניית רשת שיורית לוקחת זמן לינארי - .
3. חיפוש מסלול p יקח עד - .
   1. אם מצאנו מסלול כזה:
      1. טיפול בכל קשת במסלול, עד - .
      2. עדכון זרימה סופית .
   2. אם לא מצאנו מסלול כזה – עדכון זרימה סופית .

סה"כ, זמן הריצה יהיה: שזהו זמן ריצה לינארי, כנדרש.

1. האלגוריתם:
2. נחסיר מהקיבולת של 1, בהתאם לנדרש ונקבל: .
3. נבדוק תחילה האם הזרימה שזורמת על צלע זו קטנה או שווה מהקיבולת החדשה שניתן להזרים עליה: ?
   1. אם כן – ז"א שניתן עדין להזרים את הזרימה המרבית שמצאנו קודם לכן, למרות שהקטנו את הקיבולת של הצלע, ולכן נסיים את ריצת האלגוריתם ונחזיר את f שקיבלנו קודם.
   2. אחרת – ז"א שצריך לעשות תיקון בזרימה המוזרמת על גבי הצלעות, ואולי גם בזרימה המרבית של רשת הזרימה:
      1. נבנה גרף חדש G’ בתצורה הבאה:

* תהי הצלע , בגרף החדש G’.
* צומת המקור בגרף החדש – s’ תהיה הצומת x’.
* צומת הבור בגרף החדש תהיה – t’ (זהה לצומת הבור הקודמת -t).
* הגרף כולו יהיה מורכב מכל הצלעות והמסלולים שיצאו מ-x’ ל-t’ (ז"א נחתוך את כל הצלעות והמסלולים מ-s ל-x’ שלא ניתן להגיע באמצעותם ל-t’).
  + 1. כלל הקיבולות והזרימות על הצלעות ב- G’ יהיו זהות לקיבולות ולזרימות שהיו על הצלעות לפני שהקטנו את . שני השינויים היחידים שנבצע הם:
       1. (שכבר בוצע...).
       2. .
    2. נבנה רשת שיורית לגרף החדש שקיבלנו G’.
    3. נחפש מסלול 'p כלשהו מ-s’ ל-t’ (בגרף השיורי):
       1. אם מצאנו מסלול p’ כזה – נזרים 1 דרך כל הצלעות במסלול זה, , ונעדכן את הזרימות: . בנוסף, נחזיר את f שמצאנו קודם לכן, כיוון שעדין ניתן להזרים את אותה זרימה, אמנם דרך צלעות שונות. ונסיים.
       2. אחרת – נמצא מסלול p ברשת הזרימה המקורית G, מסלול שמכיל את , נעדכן את כל הזרימות בצלעות ששייכות למסלול זה, : . בנוסף, נעדכן את הזרימה הסופית להיות הזרימה החדשה המרבית שמצאנו : . ונחזיר את f’ (הזרימה העדכנית).

הוכחת נכונות:

נראה שבכל מקרה האלגוריתם ייעצר ויחזיר את התוצאה הנדרשת (הזרימה המרבית).

במידה ו2.3. מתקיים – ברור שנסיים, והזרימה הרלוונטית זו הזרימה שמצאנו קודם לכן.

אחרת מסעיף 2.4. קיימת אופציה למצוא מסלול אחר שיזרים 1 נוסף מצלע x’ לצלע t’, במצב כזה נסיים ונחזיר את הזרימה הקודמת. אחרת, במידה ולא נמצא כלל דרך להזרים את ה-1 העודף (בעקבות הקטנת הקיבולת של ) אז נחסיר ממסלול שלם בגרף המקורי זרימה 1 וכן נקטין את הזרימה המרבית ב-1 וזה מה שנחזיר.

בכל מקרה כזה האלגוריתם ייעצר, ולכן הוא מתקיים.

ניתוח זמן ריצה:

1. הקטנת קיבולת לצלע - .
2. בדיקה - .
   1. אם כן – החזרת ערך זרימה ידוע - .
   2. אחרת – תיקון:
      1. בניית גרף G’ לכל היותר - (אם הצלע היא אחת מהצלעות הראשונות בגרף).
      2. פעולות עדכון קטנות - .
      3. בניית רשת שיורית, זמן לינארי .
      4. חיפוש מסלול p’ גם זמן לינארי .
         1. אם כן, החזרת הזרימה - .
         2. אם לא – חיפוש מסלול ב-G גם זמן לינארי, והחזרת הזרימה .

סה"כ, זמן הריצה יהיה: שזהו זמן ריצה לינארי, כנדרש.

**שאלה 4:**

**בעיית הספיקות**.

*תהיי נוסחת 3-CNF, שבה כל אחד מהמשתנים מופיע* ***בדיוק בשלוש*** *פסוקיות שונות, וכל פסוקית כוללת* ***בדיוק שלושה*** *משתנים שונים*

*מטרה:*

1. *הוכחה כי נוסחת 3-CNF הנתונה היא נוסחה ספיקה.*
2. *מציאת אלגוריתם שמוצא השמה מספקת עבור נוסחת 3-CNF הנתונה.*

*ניתן להיעזר במשפט הול.*

*פתרון:*

1. *טענה: אם קיימת נוסחת 3-CNF, שבה כל אחד מהמשתנים מופיע* ***בדיוק בשלוש*** *פסוקיות שונות, וכל פסוקית כוללת* ***בדיוק שלושה*** *משתנים שונים, אז הנוסחה ספיקה.*

*הוכחה: נוכיח בדרך השלילה.*

*נניח בשלילה שאם קיימת נוסחת 3-CNF הנ"ל, אז הנוסחה אינה ספיקה. אם הנוסחה אינה ספיקה, ז"א לא ניתן למצוא השמה של ערכי אמת המספקת אותה (משתנים שמספקים את הפסוקיות). נסתכל על קבוצת המשתנים וקבוצת הפסוקיות הנתונות לנו.*

*נסמן: n – קבוצת המשתנים, m – קבוצת הפסוקיות.*

*נעזר במשפט הול לשם ההוכחה: קיים שידוך מושלם בגרף דו-צדדי |L|=|R| אמ"מ לכל מתקיים .*

*נקביל את נוסחת 3-CNF הנתונה לנו לגרף דו-צדדי ובו קבוצות n ו-m.*

*אם לא קיימת ספיקה, ז"א לא קיים שידוך מושלם ביניהם, ז"א מתקיימת אחת מבין 2 האפשרויות הבאות:*

1. *עבור קבוצת המשתנים – n: מתקיים:*

*אבל, נתון שכל משתנה מופיע ב-3 פסוקיות שונות (ז"א יש לו בדיוק 3 שכנים), ולכן ברור שמצב זה אינו אפשרי – הגענו לסתירה.*

1. *עבור קבוצת הפסוקיות – m: מתקיים:*

*אבל, נתון שכל פסוקית כוללת בדיוק 3 משתנים שונים (ז"א יש לה בדיוק 3 שכנים), ולכן ברור שמצב זה אינו אפשרי – הגענו לסתירה.*

*מכיוון שכל אחת מהאפשרויות הנ"ל אינה אפשרית, ברור שמצאנו סתירה לנתון ולכן המשפט המקורי נכון – ז"א נוסחת 3-CNF הנתונה ספיקה.*

1. *האלגוריתם:*
2. *נבנה גרף דו-צדדי G(V,E) בתצורה הבאה:*
   1. *הצמתים של G אלו הם קבוצת המשתנים וקבוצת הפסוקיות של הנוסחה הנתונה - V. כאשר בצד אחד נמצאת קבוצת המשתנים (n) ובצד הנגדי נמצאת קבוצת הפסוקיות (m).*
   2. *את הקשתות נבנה לפי התאמה בין משתנה לפסוקית בה הוא מוכל, עבור כל אחד מהמשתנים ועבור כל פסוקית.*

*(ז"א לכל צומת שהיא משתנה יהיו 3 קשתות שיחברו בינה לבין הפסוקיות המכילות אותה, ולכל פסוקית גם כן יהיו 3 קשתות שיחברו בין לבין כל המשתנים שהיא מכילה).*

1. *נריץ אלגוריתם לפתרון בעיית הזיווג הדו-צדדי, כפי שמופיע בעמ' 397 בספר (נוסיף את הצמתים s ו-t, נכוון את הגרף בהתאם ונריץ עליו את אלגוריתם פורד-פולקרסון למציאת בעיית הזרימה המקסימלית).*

*במקביל, ניצור גרף חדש G’ המכיל את הקשתות שזורמת עליהם זרימה 1, לפי מה שמצאנו לאחר הרצת האלגוריתם שציינו.*

*מכיוון שקבוצת המשתנים זהה לקבוצת הפסוקיות נקבל בגרף G’ זיווג מושלם (לכל משתנה תהיה זרימה – קשת – לפסוקית אחת בלבד).*

1. *נעבור על כל הקשתות בגרף G – כל קשת (u,v) מקיימת , ונבדוק האם הקשת שייכת לגרף G’ -:*
   1. *אם כן – נציב במשתנה u=T (משתנה אמת).*
   2. *אם לא – נציב במשתנה u=F (משתנה שקר).*

*הוכחת נכונות:*

1. *סעיף זו הינו סעיף טכני לבניית גרף דו-צדדי מהנתונים הקיימים, ברור כי סעיף זה מתקיים.*
2. *סעיף זה קיים ונכון לפי הוכחת הנכונות של אלגוריתם לבעיית הזיווג הדו-צדדי, כפי שמופיע בספר.*
3. *בסעיף האחרון האלגוריתם כולו ייעצר לאחר שנעבור על כל הקשתות בגרף G, מכיוון שקיימות n קשתות בגרף זה, ברור שהוא ייעצר.*

*כאשר, בסיום ריצתו של האלגוריתם לכל משתנה יהיה מוגדר אם הוא משתנה אמת או משתנה שקר, בהתאם למה שהצבנו, וזו תהיה ההשמה המספקת אותה רצינו למצוא, כנדרש.*

*השמה זו היא השמה מספקת מכיוון שהראנו שההשמה T נעשית על משתנים שהם חלק מזיווג מושלם, ז"א לפחות בכל אחת מהפסוקיות יהיה משתנה שהוא T, תנאי זה מספק על-מנת לקבל מכל הפסוקית T. ולכן סיימנו.*

*ניתוח סיבוכיות זמן ריצה:*

1. *בניית הגרף - (מכל משתנה יוצאות 3 קשתות לפסוקיות שונות).*
2. *לפי עמ' 399 בספר - .*
3. *מעבר על כל הקשתות ב-G במקביל להצבת ההשמה - .*

*סה"כ מצאנו אלגוריתם שרץ בזמן ריצה לינארי .*