# שאלה 1.

יהי קלט לבעית השידוך היציב אשר עבורו מתקיים:

1. משודכת ל- ע"י האלגוריתם GS.
2. מעדיפה את על פני .

נניח כי תשקר בנוגע להעדפתה באמצעות החלפת מקומותיהם של ו- (וזהו השקר היחיד של ובכלל) ונראה כי לא יתכן כי באלגוריתם GS, תשודך לאחר השקר ל- במקום ל- – בכך נראה כי לא קיים קלט מעודד שקרים לבעית השידוך היציב.

רשימת ההעדפות של הייתה מהצורה:

לאחר השקר רשימת ההעדפות תהיה:

(\*)לא יתכן ש- לא קיבלה הצעת נישואים מהגברים אותם היא מעדיפה על פני שכן אחרת הייתה מקבלת אותה ודוחה את או מבטלת את האירוסין ל- והשידוך שלה לא היה .

ההבדל לאחר השקר הוא בתשובות האם מעדיפה את על פני שהיו כן וכעת הן לא, ובתשובות האם מעדיפה את על פני שהיו לא וכעת הן כן,

לא יתכן כי נשאלת את אחת השאלות שהתשובה אליה שונה, שכן עד לנקודה בה תישאל השאלה הרי ש-GS ירוץ באותו אופן כפי שרץ לפני השקר, ואף אחת מהשאלות הללו לא נשאלה בריצה המקורית של GS, שכן אם הייתה נשאלת היינו מקבלים סתירה ל-(\*).

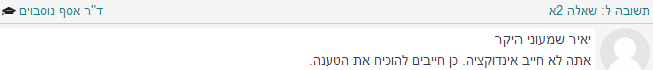
לכן בסיכומו של דבר ריצת GS תהיה זהה לאחר השקר ו- תשודך ל- ולא ל-.

ובכך הראנו כי לא קיים קלט מעודד שקרים לבעית השידוך היציב.

# שאלה 2.

א.

בהתאם לאמור באתר הקורס נוכיח את הטענה, אך **לא** באינדוקציה.



יהי גרף לא מכוון, קשיר, עם קודקודים. עלינו להראות כי קיים קודקוד שלאחר הסרתו מ- עדיין מתקבל גרף קשיר.

יהי קודקוד ב-, ונבחן את היער שנוצר מסריקת של מהקודקוד . מאחר ו- קשיר, כל הקודקודים נמצאים ברכיב הקשירות של , נקבל עץ .

מתכונות עץ, ב- ישנם קודקודים ו- קשתות.

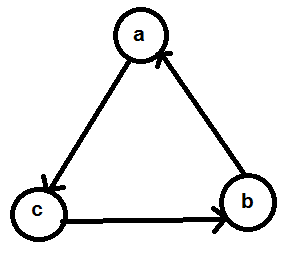
ב- ישנו עלה אחד לפחות, , אחרת כל הקודקודים בדרגה 2 לפחות ואז סכום הדרגות הוא לפחות ולכן מספר הקשתות הוא לפחות , בסתירה לכך שמספר הקשתות הוא .

אם נסיר את מ- נקבל קודקודים ו- קשתות (כי הייתה רק קשת אחת מ- ב-), וכמובן שההסרה לא יכולה ליצור מעגל ב- ולכן נקבל שוב עץ, ובפרט גרף קשיר.

אך כל הקשתות ב- לאחר הסרת נמצאות גם ב- לאחר הסרת , לכן גם כל המסלולים בין כל זוג קודקודים עדיין יהיו תקפים, ולכן אם נסיר את מ- נקבל גרף קשיר.

מש"ל.

ב.



הגרף קשיר היטב אבל לאחר הסרה של כל קודקוד תישאר רק קשת אחת ויהיה מסלול מקודקוד אחר לשני אבל לא להפך, לכן לאחר הסרה של כל קודקוד נקבל גרף שאינו קשיר היטב.

# שאלה 3.

**הרעיון המרכזי:**

גרף ניתן להכוונת צלעות מתאימה אם"ם בכל רכיב קשירות שלו יש מעגל;

במקרה בו אכן קיים מעגל בכל רכיב קשירות נסרוק את המעגל כאשר אנו מכוונים כל קשת במעגל אל הקודקוד אליו היא נכנסת;

לאחר מכן נסרוק את שאר הגרף מקודקוד כלשהו במעגל, כאשר בכל קשת לא מכוונת שעוברים בה בסריקה מכוונים אותה בכיוון הסריקה, באופן כזה נקוון קשת אחת לפחות לכל קודקוד שנגלה בסריקה ובכך נוודא שלכל קודקוד נכנסת קשת אחת לפחות.

**האלגוריתם:**

1. עבור כל קודקוד בגרף 🡨 , אם מסומן שכבר נסרק דלג עליו, ואחרת:
   1. נבצע BFS מ- עם ההתאמות הבאות:
      1. לכל קודקוד שנוסף לקבוצת הקודקודים לחקירה נשמור גם את הקשת שהובילה אליו
      2. אם הגענו לקודקוד שכבר חקרנו בעבר אך לא ע"י חזרה בקשת שבה הגענו לקודקוד הנוכחי, לא נתעלם ממנו, אלא נסמנו בתור , את הקשת שהובילה אליו בפעולה הנוכחית . (1.3).
   2. חזור עם תשובה שלילית (אין מעגל ברכיב הקשירות).
   3. נעבור על הקשתות שנשמרו בשלב (1.1.1) מ- חזרה ל- (סיבוב אחד במעגל) כאשר נתחיל ב- וכל פעם נפנה את הקשת בכיוון ההתקדמות, למשל תופנה מ- לקודקוד השני בקשת וכן הלאה.
   4. נבצע BFS מ- עם ההתאמות הבאות:
      1. לכל קודקוד שנוסף לקבוצת הקודקודים לחקירה נסמן שהוא נסרק כבר.
      2. לכל קודקוד שנוסף לקבוצת הקודקודים לחקירה, נכוון את כל הקשתות שיוצאות ממנו ואינן מכוונות כבר לכיוון הקודקוד האחר בקשת.
2. נחזיר תשובה חיובית עם הקשתות לאחר הכיוון שנקבע להן.

**הסבר:**

בשלב (1) אנו בעצם דואגים לכסות כל פעם רכיב קשירות נוסף, בשלב (1.1) אנו מוצאים מעגל ברכיב הקשירות אם קיים כזה, בשלב (1.3) אנו מכוונים חלק אחד של המעגל בכיוון אחד (בכיוון ההתקדמות ואולי יותר מרק חלק במעגל כי למשל אינו דווקא חלק מהמעגל אבל זה לא משנה כי גם כך אח"כ את שאר הגרף נכוון באופן דומה מהמעגל לכיוון שאר הגרף). באופן כזה לכל אחד מקודקודי המעגל נכנסת קשת אחת בדיוק.

בשלב (1.4) אנו מתחילים מקודקוד שאנו יודעים שכבר נכנסת אליה קשת, וממנו מבצעים BFS כאשר כל קשת שמתקדמים לקודקוד חדש היא מקודקוד שכבר מכוונת אליו קשת ולכן מכוונים אותה לקודקוד החדש וככה הלאה אנו מוודאים שלכל הקודקודים ברכיב הקשירות תכוון קשת אחת לפחות.

**ניתוח זמן הריצה:**

בשלב (1.1) אנו מבצעים בכל פעם BFS של רכיב קשירות שונה, בסה"כ אנו מבצעים פעולות.

בשלב (1.3) אנו עוברים על מסלול ברכיב שקילות שונה כל פעם, בסה"כ אנו מבצעים פעולות.

בשלב (1.4) אנו סורקים את כל הקודקודים והקשתות ברכיב קשירות שונה כל פעם, בסה"כ .

כלומר בסה"כ זמן הריצה הוא .

**הוכחת נכונות:**

נוכיח שהאלגוריתם מסתיים:

בשלב (1) אנו עוברים על כל הקודקודים, מספר סופי של צעדים.

בשלב (1.1) אנו מבצעים BFS, כמובן ששלב זה מסתיים.

בשלבים (1.3) אנו חוזרים במסלול שנוצר בשלב (1.1), המסלול סופי ולכן מובן שהצעידה בו תסתיים.

בשלב (1.4) שוב אנו מבצעים BFS, כמובן ששלב זה יסתיים גם הוא.

אם ברכיב שקילות כלשהו אין מעגל אז לא ניתן לכוון את הקשתות בגרף כנדרש: אם ברכיב קשירות כלשהו אין מעגל אז הוא עץ, בעל קודקודים ו- קשתות, לכן לא יתכן כי לכל קודקוד תיכנס קשת.

צעד (1.1) ימצא מעגל ברכיב השקילות אם"ם קיים כזה: אם צעד (1.1) הגיע לאותו קודקוד בשני מסלולים שונים ב-BFS מובן שישנו מעגל בגרף המורכב מאיחוד המסלולים מהקודקוד החדש המשותף שנמצא עד לקודקוד המשותף הקודם שנמצא במסלולים (וחייב להיות כזה, לפחות ששני המסלולים מתחילים בו).

מצד שני אם ברכיב השקילות יש מעגל לא יתכן שהסריקה בצעד (1.1) לא תבחין בכך, נניח בשלילה כי רק מסלול אחד מ- בסריקה מוביל לכל קודקוד חדש, כלומר רק קשת אחת, אז נובע כי בכל הסריקה גילינו קשתות עבור רכיב קשירות בן קודרודים ואז אין מעגל ברכיב השקילות, בסתירה להנחה.

במידה ואכן באחד מרכיבי הקשירות לא קיים מעגל הרי שסריקת ה-BFS תסתיים ונעבור לשלב (1.2) ונחזיר תשובה שלילית, כנדרש.

במקרה בו אכן קיים מעגל בכל רכיב קשירות נסרוק את המעגל כאשר אנו מכוונים כל קשת במעגל אל הקודקוד אליו היא נכנסת; לאחר מכן נסרוק BFS מאחד מקודקודי המעגל לשאר רכיב הקשירות כאשר בכל קודקוד נכוון את כל הקשתות שעוד לא כוונו אל הקודקוד אליו הן מפנות; באופן כזה מצד אחד אנו מוודאים כי הפננו את כל הקשתות, מצד שני לכל קודקוד במעגל נכנסת קשת אחת לפחות, ומצד השלישי לכל שאר הקודקודים תיכנס לפחות הקשת שהפנתה אליהם ב-BFS.

כלומר אכן לכל קודקוד תכנס קשת אחת לפחות וכל הקשתות יכוונו – מש"ל.

# שאלה 4.

א.

הרעיון הכללי הוא לחלק את מציאת המסלול לשני חלקים שחובה שיתקיימו, אחד מקודקודי ל- ואחד מ- לקודקודי , המסלול המזערי הוא דרך הקודקוד ב- שסכום אורכי המסלולים הללו עבורו הוא מינימאלי.

נבצע BFS מ-s כאשר לכל קודקוד נשמור את הקשת שהובילה אליו ואת אורך המסלול אליו מ-s. קודקודים שב- לא נמשיך לסרוק מהם (כל אורכי המסלולים מאותחלים ל- בהתחלה).

נבצע BFS מ-t כאשר צועדים בכיוון ההפוך של הקשתות ולכל קודקוד נשמור את הקשת שהובילה אליו ואת אורך המסלול ההפוך שהוביל מ-t אליו. קודקודים שב- לא נמשיך לסרוק מהם.

לבסוף נמצא את הקודקוד עבורו מינימאלי.

אם או הרי שאין מסלול מ- ל- העובר דרך קודקודים מ-U ונדפיס הודעת שגיאה.

אחרת, נשחזר את המסלול הקצר ביותר ע"י כך שנוסיף בסדר הפוך את הקשתות של ה-BFS הראשון החל בקשת שהובילה ל- כאשר בכל פעם מתקדמים לקשת של הקודקוד השני בקשת ואז לקשת שהובילה אליו. נסיים כשנגיע לקודקוד .

לאחר מכן נשלים את המסלול בכך שנוסיף לפי הסדר את הקשתות של ה-BFS השני החל בקשת שהובילה ל- כאשר בכל פעם מתקדמים לקשת של הקודקוד השני בקשת ואז לקשת שהובילה אליו. נסיים כשנגיע לקודקוד .

**ניתוח זמן ריצה:**

האלגוריתם מבצע שתי סריקות ועוד מעבר על קודקודי למציאת מינימום ועוד מעבר על שני מסלולים שאורך כל אחד מהר לכל היותר . בסה"כ זמן הריצה:

**הוכחת נכונות:**

מובן שהאלגוריתם מסתיים, הוא מבצע מעבר על הקבוצה U ועוד שתי סריקות שכמובן יסתיימו.

מכיוון שכל מסלול מתאים מבקר ב-U פעם אחת בדיוק ניתן לחלק את המסלול לשני חלקים – אחד מ- ל-U (מסלול שלא עובר ב-U פעמים נוספות לפעט בסופו) ושני מ- ל- (מסלול שלא עובר ב-U פעמים נוספות למעט בתחילתו).

אם עבור אחד הקודקודים ב- קיבלנו שהמרחק מ-s או מ-u (בהנחה ולא סורקים קשתות מ-U למעט במסלול מ-u ל-t אז סורקים את קשתות u) הוא אז מובן ש- לא יכול להשתתף במסלול מתאים, אחרת סריקת ה- הייתי מוצאת מסלול באורך כלשהו.

מסלול מזערי מתאים דרך חייב להיות חיבור של שני מסלולים שמתקבלים מה-BFSים (אני לא אומר פה שהמסלולים יחידים, רק שהאורך שלהם יחיד), אחרת ניתן היה להחליף במסלול כזה את המסלול הארוך מזה שהתקבל ב-BFS במסלול שהתקבל ב-BFS ונקבל סתירה לכך שהמסלול מזערי.

לכן נסיק שהמסלול המזערי שמבקר ב-U בדיוק פעם אחת הוא אכן המסלול דרך הקודקוד עבורו סכום אורכי המסלולים אליו מ- וממנו ל- הוא מינימאלי.

ב.

נריץ BFS מ- בשינויים הבאים:

* נשמור לכל קודקוד מה הקשת שהובילה אליו, מה המרחק שלו מ- ומה מספר הקודקודים מ- שקדמו לו במסלול.
* אם מגיעים לקודקוד שכבר סומן בקשת , מרחק ו- באמצעות קשת נוספת , מרחק , דרך מסלול המקיים אזי נעדכן את הפרטים של הקודקוד לקשת ,  
  ו- החדש.
* אם הגענו לצומת נעצור בסוף סריקת הרמה הנוכחית בגרף את ריצת ה-.

לבסוף אם הרי שאין מסלול מ- ל- ובפרט אין מסלול מתאים.

**זמן הריצה** הוא זמן של , זמן של החלפת הקשתות המובילות לצומת שניתן לחסום ע"י פעולת אחת לכל קשת בגרף והזמן לשחזור המסלול המוחזר שחסום באורך המסלול, היינו מספר הקודקודים. בסה"כ .

**הוכחת נכונות:**

ההוכחה שהאלגוריתם יסיים לרוץ לכל קלט חוקי זהה להוכחה עבור .

כעת נוכיח נכונות: אם הרי שאין מסלול מ- ל- ובפרט אין מסלול מתאים – במקרה זה האלגוריתם יתן תוצאה מתאימה.

אחרת, מתכונות ידועות של הגענו ל- במסלול מזערי באורך ; נשחזר את המסלול על ידי צעידה לאחור בקשת שהוביל ל- לקודקוד שקדם לו וכן הלאה עד .

הוכחנו שמצאנו מסלול מזערי, עלינו להוכיח גם כי המסלול מבקר ב- מספר מירבי של פעמים.

נוכיח כי האלגוריתם מוצא את המסלול המזערי המבקר ב- מספר רב ככל האופשר של פעמים לכל קודקוד בגרף במרחק מ-. בפרט עבור הקודקוד .

עבור המרחק 0 הקודקוד היחיד במרחק זה הוא בעצמו, ולכן המסלול אכן מבקר מספר רב ככל האפשר של פעמים ב-U, היינו 0 פעמים.

נניח כי הטענה נכונה עבור ונוכיח עבור :

יהי קודקוד במרחק מ-, כל מסלול מזערי מ- ל- חייב להיות מורכב ממסלול לקודקוד כלשהו במרחק מ- בצירוף קשת נוספת מ- ל-.

יתרום בכל מסלול כזה אותו ערך למספר הביקורים של המסלול ב-, הערך הוא 0 אם ו-1 אחרת.

לכן המסלול המזערי מ- ל- המבקר ב- מספר רב ככל האפשר של פעמים יהיה המסלול מ- לקודקוד שמרחקו מ- הוא ומבקר ב- מספר רב ככל האפשר של פעמים בצירוף קשת נוספת ל-.

מההנחה המסלול שהאלגוריתם יחשב לכל השכנים שמרחקם מ- הוא הוא המסלול המבקר ב- מספר רב ככל האופשר של פעמים ולכן בסריקת הקשתות של קבוצת הקודקודים במרחק מ- יסרק גם המסלול הרצוי ל- ויבחר מאחר ומספר הפעמים שהוא מבקר ב- הוא הרב ביותר ולכן הוא יועדף על פני כל מסלול אחר שיבחן ל- לפניו או אחריו (כמובן הניסוח פה כאילו המסלול המתאים הוא יחיד הוא לא מדויק אבל זה לא משנה את העובדה שיבחר אחד המסלולים המתאימים).

הוכחנו את נכונות המסלול אשר מחושב ע"י האלגוריתם לכל קודקוד במרחק של עד מ- ובפרט עבור ובכך הוכחנו את נכונות האלגוריתם.

ג.

הרעיון הכללי הוא ליצור קבוצת עותקים מתויגים של קודקודי גרף הקלט ובכל קשת לקודקוד מ-U לעבור לקבוצה האחרת של הקודקודים במקום בתוך אותה קבוצה ובאופן כזה בהנתן מסלול שמתחיל בקבוצה הראשונה נדע שהמסלול מבקר ב- מספר זוגי של פעמים אם הוא מסתיים בקבוצה הראשונה ולא בשניה.

כלומר במצב כזה אם נריץ מ- נוכל לקבל את המסלול שמקיים את התנאים הנדרשים ל-.

~~

בהנתן גרף קלט נבנה גרף חדש באופן הבא: לכל נגדיר ב- שני קודקודים: ו-.

לכל אם אזי נוסיף את הקשתות ל-, ואחרת, אם , נוסיף את הקשתות .

~~

כעת נבצע ב- מ- כאשר לכל קודקוד נשמור מה הקשת שהובילה אליו על מנת שנוכל לשחזר מסלולים לקודקוד .

אם בסוף ריצת ה- מתקיים אז אין מסלול מתאים המבקר ב- מספר זוגי של פעמים ונדפיס שגיאה מתאימה.

אחרת נשחזק אחורה את המסלול שהוביל ל- מ- כאשר אנו מורידים את סימוני ה- של הקודקודים וכך נקבל את המסלול המזערי המבקר

**הוכחת נכונות:**

האלגוריתם מריץ על גרף סופי, לכן מובן שיסתיים.

נראה שלכל מסלול בגרף מ- ל- יש מסלול מתאים באותו אורך ב- מ- ל- המבקר ב- מספר זוגי של פעמים ולהפך, ולכן המסלול הקצר ביותר ב- מ- ל- מתאים למסלול הנדרש ב-.

: יהי מסלול ב- שאורכו , אז מאופן בניית הקשתות ב- נובע שב- ישנן קשתות מ- ל-, מ- ל- עבור ומ- ל-, והקשתות הללו הן מסלול באורך מ- ל-.

המסלול חייב לעבור במספר זוגי של קודקודים שנוצרו מקודקודים ב- משום שכל כניסה לקודקוד שנוצר מקודקוד ב- עוברת לקבוצת הקודקודים האחרת (1 עוברת ל-2 ו-2 חוזרת ל-1). מאחר והמסלול מתחיל ומסתיים בקבוצה 1 נובע שלכל מעבר מ-1 ל-2 יש מעבר חזרה מ-2 ל-1 ולכן סכום המעברים הוא זוגי, כלומר מספר הביקורים של המסלול בקודקודים שנוצרו מ-U הוא זוגי, ובהתאם מספר הביקורים של המסלול המתאים ב- יבקר ב-U מספר זוגי של פעמים.

: יהי מסלול ב- המבקר ב- מספר זוגי של פעמים, אז ב- קיים מסלול מתאים,  
 וזה משום שבכל כניסה לקודקוד המתאים לקודקוד ב- יתבצע ב- מעבר בין קבוצת הקודקודים 2 ל-1 או להפך. מספר המעברים הוא זוגי ולכן חייב להתקיים שהמסלול מגיע ל- אשר שייך לקבוצת הקודקודים 1, וממנו יש קשת מתאימה ל- המתאימה לקשת בגרף . אורך המסלול המתאים הוא כאורך המסלול המקורי, , כלומר .

מש"ל.

**זמן ריצה:**

מספר הקודקודים והקשתות ב- הוא בדיוק כפול מב- ולכן זמן הבניה של , וזמן הריצה של ה- הם וזמן הריצה של שחזור המסלול הוא כאורך המסלול, כלומר לכל היותר ולכן עלות שחזור המסלול היא . בסה"כ זמן הריצה של האלגוריתם הוא .