# שאלה 1

מטרת הבעיה היא לבחור נמל בו ספינה נעצרת עד סוף החודש עבור כל לוח זמנים. כך נגדיר את קיצוצים לחודש זה. נגדיר את הבעיה כבעיית **זיווג יציב** בין ספינות ונמלים. כל ספינה מרכיבה רשימת העדפות לנמלים שרשימה זו היא הסדר הכרונולוגי בו היא מבקרת בהם. כל נמל מדרג רשימת העדפות לספינות כסדר כרונולוגי הפוך של ביקור הספינות בנמל.

רשימת ההעדפות של כל ספינה היא כרשימת הנמלים שהיא תעבור בהם עד שהיא תעגון לתחזוקה עד סוף החודש (כמטרה לעבור בכמה שיותר נמלים לפני שמתחילים תחזוקה).

רשימת ההעדפות של כל נמל היא כרשימת הספינות שעוברות בה והבחירה היא הספינה תעגון לתחזוקה (כמטרה שהספינה המשביתה את הנמל תהיה כמה שיותר מאוחרת בלוח זמנים ולכן סדר הפוך).

## הוכחת נכונות:

**טענה:** זיווג יציב בין ספינות לנמלים מגדיר קיצוץ חוקי של לוחות הזמנים (כאשר עומדים בתנאי ◊ מהשאלה).

**הוכחה:** יהי זיווג יציב בין ספינות לנמלים לפי רשימת ההעדפות שהוגדר למעלה (לפי אלגוריתם גייל-שייפלי G-S מעמוד 6 בספר). נניח בדרך השלילה כי לוח הזמנים לא עומד בתנאי ◊. כלומר, מתקיים שקיימת ספינה שנכנסת או עוגנת לנמל שיש בו כבר ספינה שעוגנת לתחזוקה. זה קורה אם"ם הספינה מעדיפה לעגון לתחזוקה בנמל והנמל מעדיף שספינה תעגון ולא . אבל זה בסתירה להנחה שהזיווג שנוצר על בסיס רשימת ההעדפות הוא יציב. ולכן יש סתירה. ולכן לוח הזמנים עומד בתנאי ◊.

מש"ל

# שאלה 2

טענה: גרף לא מכוון ניתן להכוונת קשתות כך שדרגת הכניסה גדולה מאפס אם"ם בכל רכיב קשירות בגרף יש מעגל.  
האלגוריתם לכיוון הצלעות הוא עבור כל צומת בגרף שלא נסרק עדיין נריץ BFS עם התאמה קלה שנשמור לכל צומת מאיזה צומת הגענו אליו ואם הגענו לצומת שביקרנו בו ולא שמרנו את הצומת המקדים אז נציב את הצומת שממנו הגענו ונגדיר צומת כזה כ-. לאחר הרצה של BFS, אם לא קיים צומת אז אין מעגל ונחזיר תשובה שלילית, ואם יש אז נתקדם עם הקשתות "אחורה" ונכוון אותם לצומת שאליו היא משויכת. אחר כך נריץ שוב פעם את BFS על צומת אבל הפעם כל קשת שלא כיוונו מקודם נכוון אותה מכיוון הצומת שדרכו BFS הגיע אליה.

זמן ריצה: הרצה של BFS על כל רכיב קשירות היא בסיבוכיות , המעבר על הצמתים וכיוונן הוא גם באותה סיבוכיות , ולכן סה"כ הסיבוכיות היא .

## הוכחת נכונות:

אלגוריתם BFS מבטיח לנו ריצה סופית ומעבר על כל רכיב הקשירות של צומת כלשהו. השינוי שביצענו באלגוריתם הוא רק שמירת נתונים לצומת ולכן לא פגענו בריצה שלו. הוא גם מבטיח לנו מציאת המעגל (אם קיים) שמוביל אותנו בחזרה ל-, ואז נשאר רק להכווין את המעגל, וכך קיבלנו לכל צמתי המעגל ובכלל את דרגת כניסה גדולה מאפס, ואז את שאר הקשתות ברכיב הקשירות נכווין מצומת ש-BFS הגיע דרכו ובכך הבטחנו לשאר הצמתים ברכיב הקשירות דרגת כניסה גדולה מאפס.

## הוכחת הטענה:

לא בכל רכיב קשירות בגרף יש מעגל, אם"ם קיים רכיב קשירות שבו אין מעגל, וזה אם"ם רכיב קשירות זה הוא עץ, ולכן בהכרח לא ייתכן שלכל צומת נכנסת קשת (בעץ בעל צמתים יש קשתות).

# שאלה 3

אנחנו עובדים מעל בעיית 2-SAT. נסמן מספר המשתנים שצריך למצוא עבורם השמה, ו- מספר הפסוקיות שביניהם יש פעולת וגם.

## הקדמה:

מתקיים . לפי העקרון הזה נבנה גרף מכוון שבו:

1. עבור כל משתנה כאשר נגדיר צמתים (הכוונה שניצור 2 צמתים עם השמות הללו), כלומר סה"כ .
2. עבור כל פסוקית כאשר ניצור קשתות מכוונות , כלומר סה"כ .

כעט נעבור מעל כל צומת שלא עברנו עליו, בעזרת DFS נעבור על כל הצמתים הישיגים ממנו. אם הגענו לצומת השלילה של מי שבחרנו אז נבצע את אותה הפעולה אבל על השלילה ואם שוב הגענו למקורי אז לא קיימת השמה מספקת. אחרת, החל מכל צומת ישיג נציב בהתאם לערכו אמת או שקר ונסמן שאת המשתנים הללו טיפלנו בהם.

## האלגוריתם:

1. נבנה את הגרף איך שתיארנו בהקדמה
2. נגדיר את מערך הצבות לכל (כלומר לכל משתנה מה שנציב בו)
3. לכל משתנה עבורו :
   1. הרץ ונקבל ממנו את המערך בסוף הריצה שלו.
   2. אם , אז לכל הצמתים בהם עברנו עם DFS נסמן את בהתאם לשם הצומת: לכל אם אז ואם נסמן . בפרט נסמן .
   3. אחרת, נריץ ונקבל ממנו את המערך בסוף הריצה שלו.
      1. אם מתקיים , אז לא קיימת השמה מספקת לביטוי ופשוט נחזיר "הנוסחה אינה ספיקה".
      2. אחרת, בדומה לשורה לכל הצמתים בהם עברנו עם DFS נסמן את בהתאם לשם הצומת: אם אז אחרת אם נסמן . בפרט נסמן
4. נחזיר את המערך ההצבות אשר מגיע להצבה מספקת.

## ניתוח זמן ריצה:

האלגוריתם בונה את הגרף בסיבוכיות .

עבור כל רכיב קשירות מעל צומת אנחנו מריצים את DFS לכל היותר פעמיים ועוברים על הסימונים שלו ומגדירים. כתוצאה מכך הסיבוכיות של הלולאה עבור כל תת גרף שהוא רכיב קשירות היא ואנחנו עוברים על כל רכיבי הקשירויות ולכן עבור כל הלולאה של שורה 3 קיבלנו .

ולכן קיבלנו סה"כ שהסיבוכיות היא

## הוכחת נכונות:

משמעות קשת בגרף היא שקיימת תלות בין הערכים, כלומר . ולכן אם לדוגמה קיים מסלול משמעותו היא שאם אז בהכרח .

האלגוריתם בהכרח מסיים לרוץ, כי אלגוריתם DFS הוא כמובן סופי, ולאחר כל איטרציה שלו לפחות משתנה אחד נבחר ולכן מספר האיברים במערך שלא הוגדרו קטן בלפחות 1 ולכן בשלב מסוים המערך הוגדר במלואו והאלגוריתם מסיים את ריצתו.

טענת עזר: לא קיימת השמה מספקת עבור הביטוי אם"ם קיים מסלול וגם בגרף .

על בסיס טענת העזר, אם לא קיימת השמה אז קיימים שני המסלולים הללו. בזכות נכונותו של אלגוריתם DFS הוא בהכרח ימצא את המסלול בין שני הצמתים ויסמנם אותם ב- ולכן יחזיר שאין השמה מספקת.

אם קיימת ההשמה אז לפחות אחד מהמסלולים הללו לא קיים, ועם נכונותו של DFS הוא עובר על כל הצמתים ברכיב הקשירות ובעזרת התלות ביניהם ו-"ייצוג" הצומת מסמן את הערך המתאים בהשמה.

כעט נוכיח כי אם קיימת השמה מספקת עבור הפסוק האלגוריתם מוצא השמה מספקת.

לפי טענת העזר אם קיימת השמה מספקת, לפחות מסלול אחד מתוך או לא קיימים. נניח התחלנו לבצע את הבדיקה מצומת . אם מצאנו שאין את צומת , אז הצבה של היא השמה. אם מצאנו שקיים מסלול , לפי טענת העזר, לא קיים מסלול , ולכן מבצעים באותה הדרך את הסריקה ומציבים , כלומר . ואז ברכיב הקשירות המתאים לצומת המתאים ( בהתאם לתוצאות הסריקה) אנחנו מתאימים הצבות בהתאם לצומת הקיים.

## הוכחת טענת עזר 1:

לא קיימת השמה מספקת אם"ם קיים משתנה שנסמן ב- כך שלכל הצבה ב- שנסמן ב- יש פסוקית אחרת שתלויה . משמעות המונח תלוי הוא הסימן , ולכן זה מתקיים אם"ם קיימים משתנים אחרים כך שלכל :

בשביל ההוכחה נסמן צומת בתור הצומת אם או הצומת אם

לפי הגדרת בניית הגרף:

1. אם , אז נקבל מסלול , כלומר מסלול .
2. אם , אז נקבל מסלול , כלומר מסלול .

בגלל שהתנאי שלנו היה אם"ם לכל הצבה של אז קיבלנו שזה מתקיים אם"ם שני המסלולים קיימים, כלומר הוכחנו את טענת העזר.

# שאלה 4

## הקדמה:

עבור גרף מכוון ותת קבוצה של קודקודים מועדפים נבנה גרף מכוון חדש כך ש*בונים 3 קבוצות מתויגות של (נסמן לפי מספר התגים) וכל פעם שיש קשת שמכוונת לקודקוד מועדף אז עוברים לקבוצה המתויגת הבאה (כלומר מ-0, ל-1 ולבסוף ל-2). וכל קשת שמכוונת לקודקוד לא מועדף אז היא מופיעה בכל אחת מהקבוצות.*

*על הגרף הזה אנחנו מזמנים מצומת שבכל מעבר על צומת שומר עבור הצומת את הצומת שהוביל את האלגוריתם אליו. בסוף הריצה של אנחנו בודקים את התוצאה עבור הצומת ואם יש מסלול אז הוא המסלול הכי קצר המתאים לשאלה. אם אין מסלול אז לא קיים מסלול נדרש. ואז עבור המסלול שהתקבל מורידים את כל התיוג על ידי שחזור המסלול לפי מה ששמרנו.*

## האלגוריתם:

1. שלב הבנייה של גרף חדש :
   1. לכל צור
   2. לכל :
      1. אם צור
      2. אם צור
2. כעת נבצע ב- מ- כאשר לכל צומת נשמור מה הקשת שהובילה אליו על מנת שנוכל לשחזר מסלולים לצומת .
3. אם בסוף ריצת ה- מתקיים אז אין מסלול מתאים המבקר ב- 2 פעמים בלבד ונדפיס שגיאה מתאימה.
4. אחרת נשחזר אחורה את המסלול שהוביל ל- מ- כאשר אנו מורידים את סימוני התגים של הצמתים וכך נקבל את המסלול המזערי המבקר בקודקודים המועדפים בדיוק פעמיים.

## ניתוח זמן ריצה:

האלגוריתם בונה גרף חדש, מבצע BFS עליו ובונה את התוצאה הסופית. ולכן:

## הוכחת נכונות:

האלגוריתם בהכרח מסיים לרוץ, כי הבנייה היא ישירה, אלגוריתם BFS בהכרח יסיים את עבודתו, ולכן האלגוריתם מסתיים ואינו אינסופי עבור קלט מסוים.

נשתמש בטענת עזר: מסלול בגרף בין לבין קצר ביותר שעובר פעמיים דרך קודקודים מועדפים אם"ם מסלול בגרף שעובר בין ל- קצר ביותר.

אם אין מסלול העונה על השאלה, אז לפי טענת העזר הוא גם לא קיים בגרף ולכן נקבל מהאלגוריתם שלא קיים מסלול.

אם קיים מסלול בגרף העונה על השאלה, אז לפי טענת העזר קיים מסלול מתאים בגרף . על בסיס הוכחת נכונות של , אנחנו נקבל את ה- שהוא באורך מינימלי (ובעזרת ההרחבה של שמירת המסלול) נקבל את המסלול המינימלי, ועל ידי הורדת התיוג נקבל את המסלול המתאים בגרף .

מש"ל

## הוכחת טענת העזר:

יהי מסלול בגרף העונה על השאלה. נסמן ב- (לא בהכרח שונים) את הקודקודים המועדפים דרכם עובר .  
ולכן אפשר לסמן כי .  
נשים לב כי (כלומר מסלול מתאים להגדרה הזאת). כמו כן נשים לב כי כל הצמתים שייכים לקבוצה אחרת המסלול לא מתאים לשאלה.

כל זה מתקיים אם ורק אם לכל (הכל לפי הבנייה):

1. קיימת קשת אם"ם
2. קיימת קשת אם"ם
3. כאשר לכל קשת אם"ם
4. כאשר לכל קשת אם"ם
5. כאשר לכל קשת אם"ם
6. קיימת קשת אם"ם
7. קיימת קשת אם"ם
8. קיימת קשת אם"ם
9. קיימת קשת אם"ם
10. קיימת קשת אם"ם
11. קיימת קשת אם"ם
12. קיימת קשת אם"ם

סה"כ עברנו על כל האופציות ולכן מתקיים אם"ם קיים בגרף כך ש:

מש"ל