**אלגוריתמים 20417 – ממ"ן 1**

**שאלה מס' 1**

הרעיון המרכזי:

יהי קבוצת הספינות ויהי קבוצת הנמלים כאשר כל אחת מהקבוצות בגודל .

נשתמש באלגוריתם הזיווג היציב ונתאים ספינה אחת לנמל כך שכאשר היא תגיע לנמל היא תישאר בו עד סוף החודש לצורך תחזוקה. אסור ש-2 ספינות יגיעו לאותו נמל באותו היום.

נבנה רשימת העדפות לספינות ולנמלים.

**רשימת העדפות לספינות** - ספינה תעדיף נמל על פני נמל אם נמל נמצא בלוח הזמנים של לפני נמל . כך שהנמל הראשון יהיה בלוח הזמנים של ספינה מסוימת בעדיפות הראשונה שלה, הנמל השני בעדיפות השנייה.....

**רשימת העדפות לנמלים** – נמל יעדיף את ספינה על ספינה אם ספינה מגיעה לנמל אחרי ספינה s`. כך שהספינה שאמורה להגיע אחרונה לנמל תהיה בעדיפות ראשונה שלו, הספינה שאמורה להגיע לפני אחרונה תהיה בעדיפות שנייה.....

פירוט:

את רשימת ההעדפות של הספינות נבנה ע"י ריצה על כל הספינות. לכל ספינה נסרוק את לוח הזמנים שלה ובכל פעם שנתקל בנמל כלשהו נכניס אותו לסוף רשימת ההעדפות של הספינה.

את רשימת ההעדפות של הנמלים נבנה ע"י לולאה על היום ה- בחודש מסוף החודש עד לתחילתו ( עד 1). בתוך הלולאה תהיה לולאה פנימית שתרוץ על הספינה ה-, החל מ- עד 1. בגוף הלולאה הפנימית נבדוק האם הספינה ה- עוגנת בנמל כלשהו ביום ה- בחודש. אם כן, נקרא לנמל זה ונכניס את הספינה ה-לסוף רשימת ההעדפות של הנמל .

לאחר מכן נריץ את אלגוריתם הזיווג היציב.

נכונות:

נראה שהתנאים לאלגוריתם הזיווג היציב מתקיימים:

כל ספינה עוגנת בכל נמל פעם אחת בלבד, כלומר בלוח הזמנים של הספינה מופיע כל נמל פעם אחת בלבד. חישבנו את רשימת ההעדפה מכל נמל בלוח הזמנים ולכן ברשימת ההעדפה נמצא כל נמל פעם אחת בלבד.

כפי שאמרנו כל ספינה עוגנת בכל נמל פעם אחת בלבד, לכן בכל נמל עוגנות על הספינות אך בימים שונים לפי לוח הזמנים שלהן. רשימת ההעדפה של כל נמל מורכבת מכל הספינות שעוגנות לפי סדר העגינה לכן רשימת ההעדפה של הנמל מורכב מכל הספינות.

עפ"י אלגוריתם הזיווג היציב, כל ספינה שייכת לנמל בקשר חד חד ערכי ועל, כל ספינה עובדת לפי לוח הזמנים שלה וכאשר היא מגיעה לנמל אליו היא מקושרת היא עוצרת בו לתחזוקה עד לסוף החודש. נראה שכאשר ספינה עוצרת בנמל כלשהו לא עוגנת ספינה נוספת באותו נמל. כלומר נראה שאם ספינה מקושרת לנמל והיא מגיעה אליו ביום אז לא קיים שעוגן בנמל ביום

נניח בשלילה שקיים כזה, אם קיים אז בלוח הזמנים של הספינה היא אמורה לעגון בנמל בזמן מאוחר יותר מאשר הספינה . אם כך הדירוג של הוא גבוה יותר משך ברשימת ההעדפות של הנמל , עפ"י הגדרת רשימת ההעדפות של הנמלים. לפיכך כאשר הנמל צריך לבחור בין ספינה לספינה , הוא יעדיף את מאחר והוא מדורג יותר גבוה. אך ידוע לנו ש-ו- מקושרים וזאת בסתירה להנחה.

לכן לא קיים שעוגן בנמל ביום .

זמן ריצה:

רשימת ההעדפות של הספינות מחושבת ע"י ריצה על n ספינות, כאשר בגוף הלולאה אנו סורקים את ימי החודש. שאר הפעולות רצות בזמן קבוע. לכן רשימת ההעדפות רצה ב-

רשימת ההעדפות של הנמלים מחושבת ע"י ריצה על m ימי החודש, כאשר בגוף הלולאה אנו רצים בלולאה פנימית על n הספינות, ובכל פעם מבצעים פעולות בזמן קבוע. לכן רשימת ההעדפות רצה ב-

זמן הריצה של אלגוריתם הזיווג היציב הוא כפי שהוכח בספר בעמודים 50-52.

נתון ש-  *לכן ניתן להסיק שזמן הריצה של האלגוריתם הוא:*

**שאלה מס' 2**

הרעיון המרכזי:

כדי לכוון את הגרף עלינו שיהיה בו לפחות מעגל אחד, אם יש מעגל נמצא קודקוד במעגל נראה אותו כמעין שורש ונכוון בסריקת *BFS*.

פירוט:

נניח שהגרף מכוון, ניקח קודקוד  *כלשהו ונריץ עליו סריקת DFS נקבל עץ . אם כל הקשתות ב- הן קשתות עץ אז האלגוריתם יחזיר "לא", אחרת, קיימת קשת חוזרת . נכוון אותה כ-. לאחר מכן נפנה לכל הקשתות שיוצאות מ- שאינן קשתות מכוונות ונכוון אותן אל הקשתות הבאות בתור, וכך הלאה עד שלא נמצא קשתות לא מכוונות. נדפיס למסך כל קשת שכוונה ונחזיר "כן".*

נכונות:

ב- יש 2 סוגי קשתות: קשתות עץ וקשתות חוזרות. אם כל הקשתות הן קשתות עץ מתקיים ET=EG, Vr-VG לכן גם הוא עץ. אם ב- יש n צמתים אז יש ב- קשתות (לפי משפט 3.1). אם ננסה להגדיר גרף מכוון אז בהכרח קיים קודקוד כך שדרגת הכניסה שלו היא 0 (לפי משפט 3.19), אם בגרף מכוון אין מעגלים יש בו צומת שלא נכנסת אליו אף קשת. במצב הזה נחזיק תשובה שלילית. אחרת, קיימת קשת חוזרת. האלגוריתם מכוון את כל הקשתות כך שלכל צומת ברמה מצביע לצומת ברמה . מתקבל שדרגת הכניסה של כל צומת היא לפחות 1 והקודקוד עצמו גם דרגתו לפחות 1 שכן קיימת קשת חוזרת.

זמן ריצה:

את רכיבי הקשירות נמצא בעזרת *DFS* בזמן של כמו גם סריקת עץ *BFS* לכן זמן הריצה הסופי הוא

**שאלה מס' 3**

הרעיון המרכזי:

נתרגם את הפסוקית לקשרי גרירה על מנת לתרגם את הפסוקיות לגרף ולבצע חיפוש מסלול בגרף.

פירוט:

נבנה גרף מכוון כך:

1. לכל ליטרל ניצור את הקודקודים ו-
2. לכל פסוקית בנוסחה ניצור 2 קשתות
3. על עוד קיים קודקוד שלא עבר השמה:
4. נבחר קודקוד כזה
5. נריץ סריקת *DFS* ממנו.

אם לא נמצא בקבוצה שמצאנו בסריקה נבצע השמה, נגדיר (אמת) לכל קודקוד בקבוצה שנמצאה ו- (שקר) לכל הקודקודים הנגדיים שלהם.

אחרת, נריץ סריקת *DFS* על במקום.

אם לא נמצא בקבוצה שמצאנו בסריקה נבצע השמה, נגדיר (אמת) לכל קודקוד בקבוצה שנמצאה ו- (שקר) לכל הקודקודים הנגדיים שלהם.

אחרת, נחזיר כישלון.

בסיום נחזיר הצלחה.

נכונות:

מתקיים לכל בוליאניים:

מריצים *DFS* מקודקוד ומקבלים את רכיבי הקשירות שלו, אם נמצא בין צאצאי את לא ניתן להציב "אמת" ב- כי נקבל סתירה במסלול הבוליאני. נצטרך לראות האם בסריקת *DFS* החל מ- נגיע ל-. אם כן, אז ו-נגישים הדדית ולכן שום הצבה לא תגרום לנוסחה להיות ספיקה. אחרת, קיים מסלול כלשהו שלא גורם, כלומר נוכל להגדיר "אמת" לכל הליטרלים בו.

המסלולים בגרף שקולים לתנאים המקוריים של הנוסחה ולכן ערכי הליטרלים מספקים את הנוסחה המקורית.

זמן ריצה:

בניית הגרף רצה ב-אנו רצים בלולאה על קודקודים שעוד לא ביצעו להם השמה ובכל פעם מריצים במקרה הגרוע, סריקת *DFS* כפולה על קודקודים, ובמהלך סריקה זו אנו מבצעים השמה ל-. אם נסמן ב-את מספרהקודקודים שעשינו להם השמה בסריקה הכפולה , נקבל ש-  כמובן שגם עברנו על  קשתות ולכן גוף הלולאה כולו רץ בעלות לינארית. סה"כ זמן ריצה 

**שאלה מס' 4**

הרעיון המרכזי:

נמצא מסלול מזערי דרך קודקודים מועדפים מ- ל- שמבקר בדיוק פעמיים ב- ושאורכו מזערי מבין כל המסלולים האלו.

פירוט:

נשתמש ברדוקציה לגרף , נשכפל את הגרף G ל-3 גרפים ולכל קודקוד נוסיף משתנה שיציין כמה פעמים בין 0-2 עבר מסלול מ- ל- כאשר *. נאתחל את הגרפים וכל צומת v תסומן בהתאמה*

*עבור כל קשת שמחברת בין קודקוד ל-:*

*אם נבנה את הקשת , ביקרנו ב-*

*אם נבנה את הקשת , לא ביקרנו ב-*

*נקבל גרף עליו נצבע סריקת BFS למציאת המסלול הקצר ביותר מ-* ל-

נכונות:

שלושת הקודקודים מקבילים לקודקוד

בניית הגרף אינה משנה את המסלולים הקיימים ב- אלא רק מוסיפה ערך המייצג את

נסתכל על הגרפים האלו כמעין שכבות, כאשר יש קודקוד שלא שייך לקבוצת המועדפים אנחנו נשארים באותה השכבה, אחרת, אנחנו עוברים לשכבה הבאה ע"י הוספת קשתות.

ע"י שכפול הגרף ל-3 גרפים אנחנו מייצגים את מס' הפעמים שהמסלול עובר מ- ל- השייך לקבוצת המועדפים ובכך נבטיח שהמסלול מבקר ב- בדיוק פעמיים.

לכן ב- *יהיה מסלול מ-* ל- אא"מ קיים מסלול ב- בין ל- כאשר

*סריקת ה-BFS מבטיחה לנו שנקבל את המסלול הקצר ביותר.*

זמן ריצה:

מעבר על כל צומת וקשת – זמן ליניארי

סריקת *BFS* -

סה"כ זמן ריצה