**תשובות ממ"ן 14 - אלגוריתמים**

**שאלה 1 –**

*אלגוריתם:*

*נבחר מהגרף צומת כלשהי ונסתכל על העץ המושרש בצומת . יהי צומת ב-, ותהי – קבוצת בניו של ב-. נגדיר , ו-: - גודל הזיווג המקסימלי בתת-העץ המושרש בצומת שב-. - גודל הזיווג המקסימלי בתת-העץ המושרש בצומת שב-, כך ש- אינו נמצא באף זיווג. נעבור על כל הצמתים ב- לפי למספור שלהם, ועבור כל צומת נציב: אם הוא עלה, אזי , . אחרת (אם יש לו בנים) נציב . בסופו של דבר נחזיר את - גודל הזיווג המקסימלי בעץ המושרש בצומת .*

*נכונות: לצורך הוכחת נכונות ניעזר בטענות העזר הבאות:*

*טענה 1: לכל צומת ב- מתקיים .*

*הוכחה: לפי ההגדרה של מתקיים . אם הזיווג המקסימלי בתת-העץ של לא מכיל בתוכו את אז לפי ההגדרה מתקיים . אחרת, נסיר מהזיווג המקסימלי את הקשת שמכילה בתוכה את ונקבל זיווג שלא מכיל את שגודלו הוא , ולפי ההגדרה נקבל . ובכך כיסינו את כל המקרים האפשריים ונשארנו עם התנאי .*

*טענה 2: האלגוריתם מחשב נכון את הערכים ו-עבור כל צומת ב-.*

*הוכחה: נוכיח באינדוקציה על מספר הצמתים בתת-העץ של . ניתן לראות בקלות שעבור צומת אחת ברור שמתקיים . כעת, נניח שהטענה נכונה עבור כל תת-עץ עם פחות מ- צמתים. נסתכל על עץ בן צמתים המושרש ב-. אם אינו נכלל בזיווג המקסימלי אז אין כל הגבלה על הקשתות שיכולות להיבחר בתת-העצים של הבנים שלו, ולכן הזיווג המקסימלי הוא סכום הזיווגים המקסימליים של תתי-העצים של בניו (מכאן ). אחרת (אם כן נכלל בזיווג המקסימלי) הערך תלוי בבן שאליו מחוברת הקשת שמכילה את , אותו נסמן ב-. בתת-העצים של הבנים האחרים ניתן לבחור את הזיווג המקסימלי, אבל עבור תת-העץ של לא ניתן לבחור קשת שמכילה אותו (כי הוא כבר מוכל דרך הקשת שהכילה את ), ולכן יש לבחור בזיווג שנותן את הערך . כלומר, יהיה מורכב מסכום ערכי של כל הבנים שלו, חוץ מאחד מהם שייתן את ערך ה-שלו. לפי טענה 1, ערך זה יכול להיות קטן לכל היותר ב-1 מ-. אם קיים בן שעבורו ערך ה-שווה לערך ה-, אז ניתן לבחור זיווג עם קשת שמחברת את לבן הזה, וסכום הזיווגים של העצים של בניו יהיה . אחרת, כאמור, הסכום יהיה קטן מערך זה בדיוק ב-1. ולכל זה מוסיפים 1 בעקבות הקשת שמכילה את (מכאן ). כל הערכים של תת-העצים של הבנים של חושבו נכון בשל הנחת האינדוקציה עבור תתי-עצי עם פחות מ-צמתים. האלגוריתם בוחר באופציה שנותנת את הערך המקסימלי מבין השניים, ומכיוון שאלו שתי האופציות היחידות האפשריות – האלגוריתם נותן את האפשרות שמתאימה לזיווג המקסימלי בתת-העץ של . לבסוף האלגוריתם מחזיר את גודל הזיווג המקסימלי בעץ המושרש ב-, אבל עץ זה הוא העץ כולו שניתן בקלט ולכן אנחנו מקבלים את הפלט המבוקש. נשים לב שאם הגרף הוא יער שמורכב ממספר עצים אז ניתן להפעיל את האלגוריתם הנ"ל על כל אחד מהעצים בנפרד ולסכום את כל הערכים שיתקבלו. כמובן, שבדומה לתרגילים בספר, בכדי לספק את הזיווגים עצמם כל מה שנשאר לעשות בסוף זה "ללכת אחורנית" ובהתאם לערכי ה-*-ים לבדוק לאיזו אופציה מהאופציות שנבדקו ה- שווה ומכאן ניתן יהיה להסיק עבור כל צומת האם היא נכללת בקבוצת הזיווגים ובהתאם לקבל את הזיווג עצמו.

*סיבוכיות: הסריקה על העץ על כל הצמתים לפי המספור שלהם לוקחת . בנוסף, אנחנו עוברים על כל צומת פעמיים – פעם אחת כאב ופעם אחת כבן. בכל מעבר אנחנו מבצעים פעולה בזמן קבוע. מכאן שזמן הריצה הוא .*

**שאלה 2 –**

*אלגוריתם:*

1. נבצע BFS ובכך נחלק את הגרף לרכיבי קשירות ונקבל את גודלם.
2. נסמן: – מספר רכיבי הקשירות.
3. נריץ (עמוד 289 בספר).
   1. אם האלגוריתם החזיר – פלוט "כן".
   2. אחרת – פלוט "לא".

*נכונות: ראשית נוכיח כי שני צמתים שנמצאים באותו רכיב קשירות לא שייכים לקבוצות שונות בחלוקה של הגרף. נניח בשלילה שהם כן שייכים לקבוצות שונות - צומת שייך לקבוצה ואילו צומת שייך לקבוצה () ושניהם נמצאים באותו רכיב קשירות. כיוון שהם נמצאים באותו רכיב קשירות קיים מסלול*  המחבר ביניהם. קצוות המסלול שייכים לקבוצות שונות, אזי בהכרח קיימת לפחות קשת אחת ב- כך שקצה אחד שלה ב- והשני ב-, אך זה לא יכול להתקיים לפי התנאי השני בשאלה ומכאן שקיבלנו סתירה. לכן *הוכחנו ששני צמתים שנמצאים באותו רכיב קשירות נמצאים באותה קבוצה*. מכאן שכל אחת מהקבוצות מוכרחה להיות מורכבת מאיחוד של (אחד או יותר) רכיבי קשירות, כך שכל רכיב קשירות בגרף נמצא בקבוצה אחת בדיוק. ולכן אם קיימת קבוצה של רכיבים שסכום צמתיה הוא חצי מסך הצמתים אז גודלה של הקבוצה השנייה (המכילה את כל הקודקודים שנשארו בגרף) הוא זהה, וחלוקה זו לשתי קבוצות היא פתרון מתאים, ולהיפך. נוכל לסכם ולהגיד שלפי אלגוריתם , קבוצה כזאת של רכיבים (שבה סכום הצמתים הוא חצי מסך הצמתים בגרף) קיימת אם ורק אם הרצת תחזיר .

*סיבוכיות: נסמן:*  – מספר הקשתות בגרף . ביצוע סריקת BFS לוקח . ביצוע לוקח (כי מספר רכיבי הקשירות הוא במקסימום ). ולכן סה"כ זמן הריצה הוא – פולינומיאלי כפי שהתבקשנו.

**שאלה 3 –**

*אלגוריתם:*

ראשית נמצא לכל את אורך תת-הסדרה המונוטונית העולה הארוכה ביותר שמסתיימת ב-. נסמן ערך זה ב-. מכיוון שתת-סדרה מונוטונית עולה הארוכה ביותר של הסדרה מסתיימת באיזשהו , הערך שאנחנו מחפשים, במונחי הוא , אשר קל לחישוב בהינתן ערכי כל ה-*-ים. כעת, לכל נגדיר את הקבוצה – קבוצת האינדקסים של כל האיברים אשר יכולים להופיע לפני בתת-סדרה מונוטונית עולה. כלומר - . מכאן שמתקיימת נוסחת הנסיגה הבאה: .*

*נכונות: כל תת-סדרה מונוטונית ארוכה ביותר המסתיימת באיבר היא מאורך 1 פלוס מספר האיברים שלפניו. האיבר שלפני האיבר בתת-סדרה זו (אם קיים) מוכל ב-וזאת מהגדרת . מכיוון שאנחנו ממקסמים את אורך תת-הסדרה שקודמת ל-, מובטח שתת-הסדרה המתקבלת היא מונוטונית ארוכה ביותר.*

*סיבוכיות: אנו רוצים לחשב את לכל (ואז, כאמור, לקחת מקסימום). את הערך של ניתן לחשב בהינתן הערך של לכל , וזאת ב-. אם כך, הסיבוכיות של האלגוריתם היא (התוספת היא לחישוב המקסימום), כלומר - .*

**שאלה 4 –**

*כיוון ראשון – אין בגרף מעגל עם משקל שלילי .*

*הוכחה –הערך המוחזר מאלגוריתם פלויד-וורשאל מייצג את המרחק של צומת לעצמו, ולא יתכן שהמרחק שלו מעצמו הוא שלילי כי אז זה אומר שקיים מסלול של צומת לעצמו שהוא שלילי ומסלול כזה הוא למעשה מעגל, אך נתון שאין מעגל עם משקל שלילי.*

*כיוון שני – אין בגרף מעגל עם משקל שלילי.*

*הוכחה – נניח בשלילה שקיים בגרף לפחות מעגל אחד עם משקל שלילי. נסתכל על מעגל שהאינדקס הכי גבוה של צומת בו הוא המינימלי מבין כל המעגלים השליליים, ונסמן אינדקס זה ב-.*

*האלגוריתם עדיין מחשב נכון את כל הערכים עבור כל האיטרציות בהן , בהתאם לנוסחה 6.4 ממדריך הלמידה. כלומר האלגוריתם עדיין מחשב נכון את המרחקים כאשר לא נלקחים בחשבון מסלולים שקיים בהם צומת פנימי שאינו קטן מ-X.*

*החל מהאיטרציה ה- ניתן לעבור כבר דרך הצומת (כי לפי ההגדרה הוא שייך לתת-הקבוצה ׂ) ולכן ניתן לראשונה ליצור מסלול שהוא מעגל שלילי, ואז נוסחה 6.4 לא מתאימה, כי במסלול לא מספיק לחשב רק את ערך של הסיפא מצומת , אלא גם יש לקחת בחשבון את מסלולי הביניים מ-* לעצמו בין הרישא והסיפא, כיוון שכעת מסלולים אלה יכולים להיות גם שליליים.

באיטרציה ה- של הלולאה החיצונית באלגוריתם נסתכל על צומת כלשהו על המעגל . זה סכום המשקלים של שני חלקים משלימים של ולכן משקלם הוא משקלו של , ומשקל זה הוא שלילי (לפי ההנחה שממנה יצאנו). כיוון שהאלגוריתם מכניס ל- את המינימלי מבין שני הערכים הנבדקים, ערכו יהיה כעת שלילי. היות וגם באיטרציות העוקבות יבחר הערך המינימלי, ואחד המועמדים הוא הערך השלילי הקודם, הערך יישאר שלילי לאחר כל איטרציה. ובסיום ריצת האלגוריתם יתקבל בסתירה לנתון. ולכן ההנחה הייתה שגויה ולא קיים בגרף מעגל שלילי.

**שאלה 5 –**

תתי-הבעיות ייצגו את הדרך האופטימלית לבצע הזמנות תוך הישארות עם מלאי של משאיות במחסן לאחר החודש . יהי – ערך הפתרון האופטימלי של תת הבעיה. ברור שהבעיה שאנחנו מעוניינים לפתור היא , וזאת מפני שאנחנו לא צריכים שיישאר בסוף מלאי במחסן. כמו כן, מספר תתי-הבעיות הוא *, כי יכולים להיות משאיות שיישארו במחסן בסוף כל פרק זמן. ולכן, נבנה טבלה בגודל שבה מייצג את העלות המינימלית שיש לשלם כדי שלאחר החודש יישארו במחסן משאיות לאחסון עבור החודש הבא. לצורך החישוב נשים לב שבכל חודש כדי לכסות את מספר המשאיות הרצויות אפשר להשתמש במשאיות שנשארו מהחודש הקודם, או לבצע הזמנה של משאיות בעלות* . לכן, הנוסחה שלפיה נמלא כל תא בטבלה היא:

*(הערה: ברור שהאופציה של לספק את המשאיות הנדרשות לחודש זה באמצעות חלק מתוך המלאי שנשאר מהחודש הקודם וחלק מתוך ההזמנה הוא לא רלוונטי מפני שבאופציה זו אנחנו משלמים עלויות גם על אחסנה וגם על הזמנה, וזה מן הסתם לא ייתן מצב אופטימלי)*

*בנוסף, עבור החודש הראשון חייבים להזמין משאיות בכל מקרה (מפני שאנחנו מתחילים ללא משאיות כלל) ולכן העלות היא*  לכל . העלות המינימלית עבור חודשים היא .

לאחר שחישבנו את העלות המינימלית נמצא באמצעות הטבלה את הכמות להזמנה (0 אם לא נדרש לבצע הזמנה) בכל חודש: עבור חודש מסוים ומספר משאיות שנשארו בסוף החודש לאחסון, , נבדוק האם בחודש זה הזמנו משאיות, כלומר האם . *אם כן, בחודש זה ההזמנה היא של משאיות. אחרת, לא התבצעה הזמנה בחודש זה.*