**תשובות ממ"ן 13 - אלגוריתמים**

**שאלה 1 –**

1. עלינו להוכיח שאם מסלול מ- ל- מכיל רק קשתות שימושיות אז הוא מסלול קצר ביותר מ- ל-. נוכיח זאת באינדוקציה על אורך המסלול. בסיס האינדוקציה: אורכו של המסלול הוא 1. מכאן שישנה רק קשת אחת במסלול והיא קשת שימושית. ולפי ההגדרה של קשת שימושית הרי שקיים מסלול קצר ביותר מ- ל- שהיא הקשת האחרונה בו. הנחת האינדוקציה: נניח שאם מסלול באורך מ- ל- מכיל רק קשתות שימושיות אז הוא מסלול קצר ביותר מ- ל-. הוכחה: כעת נוכיח למסלול באורך . אם מ- ל- מכיל רק קשתות שימושיות, אז ניתן לחלק אותו ל-2 מסלולים ו- כך ש: – הרישא של המסלול אשר מכיל את הקשתות הראשונות. – הסיפא של המסלול אשר מכיל את הקשת האחרונה. ברור לפי הנחת האינדוקציה ש- מכיל רק קשתות שימושיות ולכן הוא מסלול קצר ביותר מ-ל- (הקודקוד שקודם ל-במסלול ). כמו כן, מכיל רק קשתות שימושיות ולכן הוא מסלול קצר ביותר מ- ל-. לכן ברור שהמסלול מ- ל- יהיה קצר ביותר, מפני שהוא מכיל את המסלול הקצר ביותר עד ל- ומ- ל-. מ.ש.ל.
2. עלינו להוכיח שאם מסלול מ- ל- מכיל קשת לא שימושית אז הוא אינו מסלול קצר ביותר מ- ל-. יהי מסלול מ- ל- המכיל קשת לא שימושית . נניח בשלילה שהמסלול הוא מסלול קצר ביותר. נחלק את המסלול לשני תתי-מסלולים ו- כך ש: – הרישא של המסלול מהקודקוד לקודקוד . – הסיפא של המסלול מהקודקוד לקודקוד . ברור שמתקיים . ידוע כי אינה קשת שימושית, ולכן לפי ההגדרה לא קיים מסלול קצר ביותר מ- ל- כך שהקשת האחרונה בו היא , ומכאן שקיים מסלול אחר מ- ל- כך שהקשת האחרונה בו שימושית. נסמן את המסלול , ומתקיים , כלומר , בסתירה לנתון ש-הוא מסלול קצר ביותר.
3. עלינו להוכיח שבמסלול שני קצר ביותר קיימת בדיוק קשת לא שימושית אחת. יהי מסלול מ- ל- המכיל קשתות שאינן שימושיות

. נניח בשלילה כי מסלול שני קצר ביותר. נבחר בקשת – הקשת הראשונה שמופיעה במסלול , ונסמן . נחלק את המסלול לשני תתי-מסלולים ו- כך ש: –המסלול מהקודקוד לקודקוד . –המסלול מהקודקוד לקודקוד . ברור שמתקיים . מסעיף ב' נובע כי המסלול אינו הקצר ביותר, ולכן נוכל להחליפו במסלול אחר מ- ל- קצר ביותר, ולכן מורכב רק מקשתות שימושיות. קיבלנו מסלול , ומתקיים , אולם נשארו עדיין קשתות שאינן שימושיות במסלול , ולכן מסעיף ב' קיים מסלול קצר ביותר מ- ל- שאורכו , ומתקיים *. כלומר, לא יתכן כי מסלול שני קצר ביותר, בסתירה להנחה. כמו כן, אם נניח ש-מסלול שני קצר ביותר שאינו מכיל אף קשת לא שימושית, הרי שמסעיף א' נקבל סתירה גם כן מפני שהוא יהיה בעצם המסלול הקצר ביותר. לכן נשארנו עם האופציה היחידה שהמסלול השני הקצר ביותר מכיל קשת לא שימושית אחת בלבד.*

1. ראשית כל נוכיח מספר טענות עזר שיעזרו בהרכבת האלגוריתם:

טענה 1: כל הקשתות המוחזרות מהרצת אלגוריתם דייקסטרה על צומת הן קשתות שימושיות. הוכחה: אלגוריתם דייקסטרה מחזיר את כל המסלולים הקצרים ביותר מצומת לכל שאר הצמתים, ומסעיף א' נובע שכל מסלול כזה מורכב מקשתות שימושיות בלבד.

טענה 2: יהיה גרף מכוון. ניצור גרף מכוון  *המתקבל מהיפוך הכיוונים של הקשתות בגרף . הרצת אלגוריתם דייקסטרה על צומת ב-תחזיר את כל המסלולים הקצרים ביותר מכל הצמתים אל צומת בגרף . הוכחה: נניח בשלילה שהרצת דייקסטרה על צומת ב-החזירה מסלול שאינו קצר ביותר מצומת ל- ב-. מההנחה ש- אינו המסלול הקצר ביותר, נקבל כי קיים מסלול קצר ביותר ב-, אולם מסלול זה יהיה המסלול ב-*, והראנו כי דייקסטרה מצא מסלול שאינו הקצר ביותר ב-, בסתירה.

*האלגוריתם:*

1. הרץ דייקסטרה מצומת וסמן את כל הקשתות שדייקסטרה עבר בהן (שהן שימושיות לפי טענה 1).
2. הרץ דייקסטרה מצומת עבור קשתות הפוכות (מה שייתן מכל צומת את מרחקה ל-, לפי טענה 2).
3. .
4. עבור כל צומת שאינה שימושית:
   1. (אשר מתקבלות משורות 1 ו-2).
5. החזר את .

*נכונות: שורות 1 ו-2 מוצאות את המסלולים הקצרים ביותר מ- לכל צומת ומכל צומת אל . מהטענות של סעיפים א' ו-ב' – מסלולים אלו מורכבים מקשתות שימושיות בלבד. לפי סעיף ג', המסלול השני הקצר ביותר יהיה מורכב מקשת שאינה שימושית אחת, ולכן נמצא את הקשת הלא שימושית אשר נותנת את הסכום המינימלי, וזה יהיה המסלול השני הקצר ביותר.*

*סיבוכיות: האלגוריתם מריץ דייקסטרה פעמיים בסיבוכיות של , הופך פעם אחת את כל הקשתות ורץ על כל קשת לא שימושית בזמן . הפיכת הקשתות וריצה על כל הקשתות הלא שימושיות לוקחת , שזה זמן זניח יחסית להרצת דייקסטרה, ומכאן שהאלגוריתם סה"כ רץ בזמן ריצה של דייקסטרה – כלומר .*

**שאלה 2 –**

ידוע כי עבור פולינום מדרגה מתקיים - . מכאן ש- לכל *. החישוב של ידרוש וקטור תשובה בגודל , ולכן בחישוב זה . נציב את בפונקציה ונקבל כי: . קיבלנו כי בתחום השורשים חייבים להיות זהים, נראה מה קורה בתחום : ברור כי ולכן נוכל לרשום את התחום הנ"ל כך:* :

אולם הוספה של בחזקה היא סיבוב של 360 מעלות בדיוק, ולכן נקבל כי גם בתחום אנו מקבלים: *. ומכאן, שלכל התחום : .*

**שאלה 3 –**

1. *עלינו להוכיח כי אם ניתן לחשב את , בהינתן מטריצה , בעזרת 5 פעולות כפל של מספרים ממשיים. ראשית נשים לב לצורת :*

*5 פעולות הכפל שבאמצעותן נוכל לחשב את הם:*

*כעת נראה כי בעזרתן אכן ניתן לבנות את כל ארבעת המחוברים מעל הממשיים:*

*ומכאן שכל מטריצה בגודל ניתן להעלות בריבוע בעזרת 5 פעולות כפל של מספרים ממשיים.*

1. *האלגוריתם המוצע אינו פותר את הבעיה.*

*האלגוריתם מנסה לפתור את הבעיה באופן רקורסיבי לפי סעיף א':*

*כלומר 5 תתי הבעיות הן:*

*אולם, הבעיה היא שבעולם המטריצות כפל אינה פעולה חילופית ולכן:*

*,*

ומכאן שהאלגוריתם לא יכול להשתמש ב-5 המכפלות הללו, ולכן הוא שגוי.

כמו כן, האלגוריתם מכפיל מטריצות זהות זו לזו (כלומר מעלה בריבוע) ואילו הקריאות הרקורסיביות מתבקשות להכפיל שתי מטריצות שאינן בהכרח זהות.

**שאלה 4 –**

ראשית כל נוכיח מספר טענות עזר שיעזרו בהרכבת האלגוריתם:

טענה 1: לכל מטריצה יש מינימום מקומי. הוכחה: נבחר את הצומת עם התווית הקטנה ביותר במטריצה. מכאן שהוא קטן מכל שאר התוויות במטריצה ובפרט מכל התוויות , עבור כל הצמתים המחוברים אליו בקשת.

טענה 2: בכל מטריצה , המינימום המקומי ימצא או על שפת המטריצה או בפנים המטריצה. הוכחה: מטענה 1 נובע שלכל מטריצה יש מינימום מקומי. אם הוא אינו על השפה, אזי נוכל "לחתוך" מהמטריצה את שפתה ולקבל מטריצה חדשה, , קטנה יותר, אשר גם עליה חלה טענה 1.

טענה 3: יהיו תתי-המטריצות המתקבלות מחלוקת המטריצה לארבעה חלקים. כמו כן, תהי קבוצת כל האיברים שעל שפת , ונסמן , כלומר האיבר הקטן ביותר על כל השפות . אזי מתקיים: אם הפינה (של אחד מהחלקים), אז הוא המינימום. אחרת, מספיקה השוואה אחת בלבד כדי לדעת האם הוא מינימום מקומי או לא. הוכחה: נמצא על שפה של תת-מטריצה כלשהי. ברור כי לכל תא בשפה ישנם שני שכנים באותה השפה, וזאת מפני שהשפה היא סגורה. כעת נבחין שאם נמצא בפינה אזי יכול להיות: א. שהפינה היא הפינה של המטריצה הראשית , ואז בוודאות הוא המינימום המקומי (כי שני התאים הנוספים שיש להשוות מולם נמצאים על השפה, ובפרט גדולים ממנו). ב. ש- צמוד לשפות של שתי תתי-מטריצות, ובפרט קטן יותר מכל האיברים על שפותיהן, וגם כאן הוא המינימום המקומי. *ג.* ש- צמוד לשפות של שלוש תתי-מטריצות, ובפרט קטן יותר מכל האיברים על שפותיהן, וגם כאן הוא המינימום המקומי. *מכאן שבהכרח -* אם הפינה (של אחד מהחלקים), אז הוא המינימום. *כעת נבחן את המקרה ש- הוא לא הפינה. לכל צלע בשפה – או שהצלע היא גם השפה של המקורית, או שהצלע צמודה לצלע שפה של אחר (). במקרה הראשון, השפה היא השפה של המטריצה הראשית, ולכן לא ניתן להשוות לאיבר שמחוץ למטריצה, ומתוך 4 האיברים להשוואה נותר איבר בודד. במקרה השני, הוגדר להיות המינימום על כל השפות, ולכן קטן מאיבר זה, וגם כאן מתוך 4 האיברים להשוואה הראנו כי תמיד קטן מ-3 איברים, ולכן נשאר להשוות רק עוד איבר נוסף בודד.*

*האלגוריתם:*

1. *חלק את המטריצה ל-4 חלקים.*
2. *מצא את המינימום משפות כל ארבעת המטריצות שהתקבלו.*
3. *אם המינימום נמצא בפינה אז מצאנו מינימום מקומי – סיים את האלגוריתם. (לפי טענה 3)*
4. *אחרת – בדוק את ערך התא שעוד לא נבדק, השכן של המינימום. (לפי טענה 3)*
   1. *אם ערך התא גדול מהמינימום אז מצאנו מינימום מקומי - סיים את האלגוריתם. (לפי הגדרת המינימום המקומי)*
   2. *אחרת – קרא לאלגוריתם ברקורסיה על תת המטריצה המכילה את המינימום.*

*נכונות:*

1. *ניתן לראות שהאלגוריתם בהכרח יסתיים וזאת מפני שבכל קריאה רקורסיבית גודל המטריצה יקטן, ולכן במקרה הגרוע ביותר נגיע למטריצה , בה כל התאים הם פינות (ומטענה 3 המינימום יהיה אחד מהם).*
2. *התשובה שנקבל תהיה נכונה מפני שאם האלגוריתם יסתיים בשורה 3 אז הוא נכון לפי טענה 3, ואם הוא יסתיים בשורה 4.1 אז הוא נכון לפי טענה 3 והגדרת המינימום המקומי.*
3. *כמו כן, הקריאה הרקורסיבית עושה את הדבר הנכון מפני שמטענה 2 – או שהמינימום על השפה (שורה 3) או שהמינימום בפנים המטריצה ואז אנחנו קוראים ברקורסיה לפנים המטריצה (לחלק הרלוונטי שבו).*

*סיבוכיות: נסתכל על החלק של מציאת המינימום - במטריצה בגודל שחולקה ל-4 מטריצות בגודל , יש איברים, ומציאת מינימום ב- איברים לוקחת . הבדיקה האם המינימום הוא מינימום מקומי לוקחת . ולכן קיבלנו את נוסחת הנסיגה , ולכן אנחנו מקבלים סה"כ , כנדרש מאיתנו.*