ממן 12 אלגוריתמים

# שאלה 1

נגדיר להיות המשקל של מסלול מזערי מs לv בגרף. נגדיר ש הוא תת מסלול של אם כל הקשתות ב מופעיות באותו סדר איפשהו במסלול .

## סעיף א

יהי מסלול בגרף שכל הצלעות בו הן שימושיות. נוכיח ש.

נסמן , כלומר . נוכיח באינדוקציה על i ש

הוא מסלול מזערי מ ל לכל .

בסיס: : הקשת שימושית כלומר היא על מסלול מזערי כלשהו מ ל ולכן . המסלול הוא מסלול מ ל כלומר . לכן כלומר הוא מסלול מזערי.

צעד: נניח מסלול מזערי ונראה מסלול מזערי. הקשת היא אחרונה במסלול מזערי כלשהו מ ל. ולכן . המסלול הוא מזערי לכן . כלומר . המסלול הוא מסלול מ ל כלומר . לכן כלומר מזערי.

הוכחנו שלכל הוא מסלול מזערי מ ל ובפרט הוא מסלול מזערי מ ל, כלומר מs לv. לכן הוא מסלול מזערי והוכחנו את הטענה.

## סעיף ב

נגדיר שוב . נניח שקיימת קשת שאינה שימושית. אז היא אינה על אף מסלול מזערי כלומר קיים מסלול מs ל שמשקלו קטן מכל מסלול מs שעובר ב ובפרט מ. כלומר, . לכן, אם נחליף את תת המסלול במסלול ונסמן את התוצאה , נקבל כי:

כלומר אינו מזערי.

## סעיף ג

אם הוא מסלול כמעט מזערי אז בפרט אינו מזערי לכן לפי סעיף ב יש לו לפחות צלע לא שימושית אחת. נראה שיש בדיוק אחת, כלומר נניח בשלילה שיש לפחות שתי צלעות לא שימושיות ב והוא עדיין כמעט מזערי ונגיע לסתירה.

נסמן את שתי הצלעות הלא שימושיות . בלי הגבלת הכלליות נניח . אם נסתכל על בתור v, אז לפי סעיף ב קיים מסלול המקיים . כעט, נוסיף ל את הקשתות כך שהוא יהיה מסלול מs לv. יש במסלול זה קשת שאינה שימושית והיא כלומר לפי סעיף ב קיים מסלול כך ש.

סה"ך, קיבלנו שקיימים מסלולים כך ש

כלומר אינו כמעט מזערי בסתירה.

לכן לא יכול להיות שיש 2 קשתות לא שימושיות או יותר. כלומר סה"כ הוכחנו שאם  *כמעט מזערי יש בו קשת אחת בדיוק שאינה שימושית.*

## סעיף ד

אם היא הקשת היחידה שאינה שימושית ב אז ברישא של  *מs ל כל הקשתות שימושיות, אז לפי סעיף א' הרישא היא מסלול מזערי. הסיפא של מ לv גם היא מורכבת רק מקשתות שימושיות אז לפי סעיף א' הסיפא היא מסלול מזערי.*

## סעיף ה

האלגוריתם:

1. נמצא על ידי אלגוריתם דייקסטרא לכל צומת את אורך המסלול המזערי ממנה אל v, על ידי הרצת האלגוריתם עם v כמקור וקשתות הפוכות עם משקלים זהים. נסמן
2. נמצא גם על ידי אלגוריתם דייקסטרא על הגרף המקורי את אורך המסלול המזערי מs לכל צומת. נסמן .
3. לכל צומת u, שממנה יוצאות הקשתות , נתעלם מכל הקשתות השימושיות, כלומר אלו שמקיימות , ומתוך הנותרות נבחר את הקשת כך שהסכום הוא השני הכי קטן. נסמן מספר זה ב.
4. נעבור על כל הצמתים בגרף ונחפש u כך ש הוא ה המינימלי מכל הצמתים. המסלול הכמעט מזערי בגרף הוא המסלול המזערי מs לu, הקשת שדרכה מתקבל , והמסלול המזערי מהקודקוד שאליו הגיעה לv.

*הוכחת נכונות:*

*לפי סעיפים ג' וד', המסלול הכמעט מזערי מs לv חייב להיות מהצורה כך ש מזעריים ו(u,w) לא שימושית. שלבים 3 ו4 הם למעשה סריקה של כל המסלולים מהצורה הזו- עוברים לכל צומת u על כל צומת w שהיא קשורה אליה בקשת לא שימושית. לכל צומת בוחרים את הw השכנה שלה כך שהמסלול מu לv יהיה מינימלי דרך w, ובסוף בוחרים את הu שעבורה מתקבל אורך המסלול המינימלי.*

*כלומר, מתקבל בסוף האלגוריתם המסלול מהצורה הנ"ל בעל האורך המינימלי נסמנו . מכיוון שהמסלול הכמעט מזערי חייב להיות מהצורה הזו, אם נסמן המסלול הכמעט מזערי, . כמו כן, מכיוון שהקשת אינה שימושית, לפי סעיף ב, המסלול אינו מזערי. כלומר . לפי הגדרת המסלול הכמעט מזערי, לא יכול להיות ש*

*כי המשקל של הכמעט מזערי הוא השני הכי קטן. לכן, האפשרות היחידה היא ש*

*כלומר המסלול שבחר האלגוריתם הוא כמעט מזערי.*

*זמן ריצה:*

*שלב 1 מתבצע ב, מכיוון שהוא פשוט אלגוריתם דייקסטרא, ההוכחה של זמן הריצה שלו מוצגת בספר. נבחין כי הפיכת כיוון הקשתות לוקחת לכן זניחה ביחס לזמן ההרצה של דייקסטרא.*

*כנל לגבי שלב 2.*

*בשלב 3 עוברים על כל קשת בדיוק פעם אחת ומבצעים עליה פעולות אלמנטריות (הגדלים שמשתמשים בהם בחישובים חושבו כבר לכל צומת בשלבים 1 ו2). לכן זמן הריצה הוא .*

*בשלב 4 עוברים על כל צומת בדיוק פעם אחת ובמצעים עליה פעולות אלמנטריות, לכן זמן הריצה הוא . נתון שהגרף קשיר, לכן יש לו לפחות קשתות (\*), ובכל מקרה כלומר זמן הריצה הוא .*

*סה"כ זמן הריצה של האלגוריתם הוא עם כן:*

*(\*) טענת עזר- אם G גרף קשיר יש בו לפחות קשתות. נוכיח באינדוקציה על גודל הגרף.*

*בסיס: |V|=1, אז אכן יש לפחות 0 קשתות.*

*צעד: נניח שלכל גרף קשיר עם n קודקודים יש n-1 קשתות לפחות ונראה שלכל גרף קשיר עם n+1 קודקודים יש n קשתות.*

*יהי גרף קשיר כלשהו עם n+1 קודקודים. נבחר קודקוד כלשהו ונמחק אותו ואת כל הקשתות שנוגעות בו. אם הגרף שנותר הוא גרף קשיר, אז הוא באורך n לכן יש בו n-1 קשתות, ואז בגרף המקורי יש לפחות n קשתות כי הוא קשיר אז הקודקוד שנמחק חייב להיות מחובר לגרף החדש בלפחות קשת אחת.*

*אם הגרף שנותר אינו קשיר, אז הוא מכיל k רכיבים קשירים. נסמן כמות הקודקודים בכל אחד מהם. אז בכל אחד מהם יש קשתות לפחות. הצומת שמחקנו חייבת להיות מחוברת לכל אחד מהרכיבים הקשירים, אחרת הם לא היו קשירים בגרף המקורי. כל קשת כזו נמחקה לכן נמחקו לפחות קשתות. סה"כ, היו בגרף המקורי*

*N קשתות.*

*והוכחנו את הטענה.*

# שאלה 2

האלגוריתם יקבל את G’ ואת T ויחזיר T’ שהוא עץ פורש מזערי עבור G’.

1. נבחר צומת כלשהי בT ונתקדם ממנה על כל הקשתות של T. לכל קשת נבדוק אם היא קיימת בG'. אם קשת כלשהי חסרה (ויכולה להיות רק אחת בהגדרת הבעיה), נעבור לשלב 2. אם כל הקשתות עדיין קיימות נחזיר את T.
2. נמצא את הרכיב הקשיר של v ושל u בחיפוש לעומק. מכיוון שT עץ, אין בו מעגלים, לכן כאשר מנתקים בו קשת מתנתק חלק שלם של העץ מחלק אחר כי אין אף מסלול אחר ממנו חזרה, לכן בוודאות יהיו שני רכיבים קשירים שאינם מחוברים.
3. נעבור על כל הקשתות בE. אם E היא מצומת ברכיב של u לצומת ברכיב של v, נזכור אותה.
4. מתוך הצמתים שמצאנו ב3, נבחר את זו בעלת האורך המינימלי. נוסיף אותה לT ונקבל T’ שאותו נחזיר.

הוכחת נכונות:

אם ההוכחה הסתיימה בשלב 1 הנכונות טריוויאלית: כל עץ פורש בG’ הוא גם עץ פורש בG (כל הקשתות של G’ נמצאות בG () וקבוצת הקודקודים זהה ()). לכן אם היה עץ פורש מינימלי שאינו T בG’ הוא גם היה עץ פורש מינימלי בG וזו סתירה כי T עץ פורש מינימלי של G.

לכן נניח ש. שנוצר על ידי האלגוריתם הוא עץ המכיל את אותן קשתות כמו בT, בלי ועם קשת נוספת בין הרכיב הקשיר של v לרכיב הקשיר u, שהיא הקשת המינימלית בין הרכיבים האלו, נסמנה .

נוסיף או נוריד מכל קשת בגרף מספר קטן מאוד כלשהו כך שמשקלי כל הקשתות יהיו שונים. בעמודים 161,162 יש הסבר מדוע אם נמצא עץ פורש מינימלי לאחר השינוי הזה, הוא יהיה גם עץ פורש מינימלי בגרף המקורי.

כעט, יש קשת שהיא המינימלית (בגרף שבו הוסרה) בין הרכיב הקשיר של v למשלים שלו (הרכיב הקשיר של u). עלויותיהן של כל הקשתות שונות זו מזו לכן לפי משפט 4.17 e מופיעה בכל עץ פורש מינימלי של G’.

לבסוף, לכל קשת , , קשת זו מחברת בין הרכיב הקשיר של w1 בT’ ובין הרכיב הקשיר של w2 בT’. כלומר, היא הקשת המינימלית שמקשרת בין קבוצת הצמתים של הרכיב הקשיר של w1 לשאר הגרף (אם הייתה קשת אחרת יותר קטנה, אז לפי 4.17 היא הייתה מופיעה בעץ הפורש ולכן דרכה ואז דרך e’ היה נוצר מעגל, ובעץ פורש לא יכולים להיות מעגלים כי הוא עץ). שוב, ממשפט 4.17, נובע שקשת זו נמצאת בכל עץ פורש מינימלי של G’.

סה"כ הוכחנו שכל קשת ב נמצאת בגרף הפורש המינימלי של G’, ושe נמצאת בגרף הפורש המינימלי של G’ כלומר T’ מוכלת בעץ הפורש המינימלי של G’. T’ הוא עץ פורש של G’ לכן נובע שT’ הוא עץ פורש מינימלי של G’ כלומר האלגוריתם עובד.

זמן ריצה:

שלב 1 עובר על כל קשת פעם אחת לכל היותר לכן פועל ב.

שלב 2 עובר על כל צומת פעם אחת בדיוק. כמו בשאלה 1, גם כאן יש עץ פורש של הגרף לכן

כלומר שלב זה פועל ב.

שלב 3 עובר על כל קשת פעם אחת בדיוק לכן פועל ב.

שלב 4 עובר על תת קבוצה של קבוצת הקשתות, פעם אחת בדיוק על כל קשת, לכן פועל ב.

סה"כ, זמן הריצה הוא ובפרט .

# שאלה 3

האלגוריתם יבחר לכל אחד מהמשתנים השמות שיספקו כמה שיותר פסוקיות חדשות. כלומר, ל יבחר ערך T כדי לספק את הפסוקיות הראשונה והשנייה. ל יבחר ערך T בשביל השלישית והרביעית, ול ערך T בשביל החמישית והשישית. כלומר הערך שיוצב בפסוקית האחרונה הוא כלומר F כלומר הביטוי כולו לא יסופק.

נבחין כי אכן קיימת הצבה מספקת: . כל הפסוקיות יסופקו- 1-6 על ידי המשתנים הממוספרים על ידי שתי ספרות, שמקבלים ערך אמת, והפסוקית האחרונה על ידי ערך השקר של המשתנים האחרים.

כלומר האלגוריתם החמדני לא עובד.

# שאלה 4

יהיה T עץ מושרש בינארי לחלוטין. נראה שקיימת סדרת שכיחויות שאחד מעצי הופמן שלה הוא T. כלומר נראה שקיים T’ המתקבל מעץ הופמן של סדרת שכיחות כלשהי עבורו .

נסמן (המרחק המקסימלי מהשורש של עלה כלשהו). לכל עלה , נוסיף את , כאשר הוא המרחק של מהשורש.

נראה שאלגוריתם הופמן בונה את העץ T על הקלט . נסמן את העץ שנוצר על ידי האלגוריתם T’ ונראה ש.

נבחין כי בעץ בינארי לחלוטין בכל עומק יש מספר זוגי של צמתים (כי כאשר יוצרים צומת בעומק כלשהו, חייבים ליצור לו גם אח באותו עומק).

לכן, בגלל שכמות המופעים של כל שכיחות זוגית, תמיד כשמחברים שתי צמתים באלגוריתם הם יהיו בעלי אותו מספר- מתחילים בלחבר את כל הצמתים עם שכיחות "1". כאשר מחברים אותם מקבלים מספר כלשהו של צמתים המסומנים "2". זו בדיוק הכמות של צמתים בעומק שיש להם בנים. כלומר אם מוסיפים את הצמתים שהיו מלכתחילה מסומנים "2" מקבלים (עלים בעומק ) מקבלים את כל הצמתים שהיו בT בעומק וזה מספר זוגי. לכן, עכשיו כשהשכיחות המינימלית היא "2", ויש מספר זוגי של כאלה, באינדוקציה מגיעים לכך שתמיד כמות הצמתים עם השכיחות המינימלית הוא זוגי ולכן תמיד מחברים שני צמתים באותו גודל.

מכך נובע, שכל פעם שמחברים צומת כלשהו בשלב כלשהו של האלגוריתם, מכפילים את הערך שלו.

לבסוף, הסימון על הצומת הסופי שיתקבל לאחר איחוד כל הצמתים יהיה פעמיים גודל השורש כלומר . לכן, עבור צומת בעומק d כלשהי, אם מחברים אותה m פעמים, אז

כלומר כלומר כלומר כל צומת מאוחד סה"כ d פעמים, כלומר העומק שלו בעץ שיווצר T’ הוא d.

הראנו שכמות הצמתים בעומק d כלשהו בT שווה לכמות הצמתים בעומק d בT'. מכיוון שאין לצמתים שם