לירון שלי

313576209

אלגוריתמים 20417

04/12/19

**ממ"ן 12**

**שאלה 1**

סעיף א' –

יהי מסלול, ויהיה סדר הצלעות במסלול , וסדר הקודקודים , כך שהצלע המחברת בין  *היא , בין היא , וכו', עד לצלע האחרונה בין היא .*

נוכיח את הטענה כי אם כל הצלעות ב שימושיות, אז מסלול מזערי, באינדוקציה על כמות הצלעות במסלול.

בסיס האינדוקציה: עבור מסלול בעל צלע אחת, אם מדובר בצלע שימושית, לפי הגדרה, היא צלע אחרונה (ויחידה) במסלול מזערי , כלומר הטענה מתקיימת.

נניח כי הטענה מתקיימת עבור n צלעות, נראה כי הטענה תתקיים גם עבור n+1 צלעות:

כל הצלעות במסלול שימושיות. המסלול לפי הנחת האינדוקציה הוא מסלול מזערי, מאחר והצלע המחברת בין הקודקוד לקודקוד היא שימושית, אז לפי הגדרה היא צלע אחרונה במסלול מזערי. כלומר המסלול הוא מסלול מזערי. מש"ל.

סעיף ב' –

הטענה - אם קיימת צלע לא שימושית במסלול אז  *אינו מסלול מזערי, שקולה לטענה – אם מסלול מזערי, אז כל הצלעות ב שימושיות.*

יהי מסלול מזערי, ויהיה סדר הצלעות במסלול , וסדר הקודקודים , כך שהצלע המחברת בין  *היא , בין היא , וכו', עד לצלע האחרונה בין היא .*

שימושית, מאחר ו- הוא מסלול מזערי.

כעת נסתכל על תת מסלול , מאחר והוא תת מסלול מזערי, הוא בעצמו מסלול מזערי, לכן שימושית, מאחר ו- הוא מסלול מזערי.

וכך נמשיך ונסתכל על כל תת מסלולים לכל 1<i<n , כאשר כל אחד מתתי המסלולים הוא מסלול מזערי בעצמו ולכן הצלע האחרונה בכל תת מסלול היא שימושית.

כלומר, כל הן שימושיות, ומאחר והן כל הצלעות הקיימות במסלול , הרי שכל הצלעות במסלול המזערי הן שימושיות.

סעיף ג' –

אם הוא מסלול כמעט מזערי, אז לפי סעיף ב' קיימת בו לפחות צלע לא שימושית אחת, נראה כי לא יתכן שבמסלול זה יש יותר מצלע לא שימושית אחת.

*צלע היא לא שימושית אם היא צלע אחרונה במסלול שאינו מזערי.*

*נסמן את המסלול המזערי ב ואת המסלול שאינו מזערי ב .*

תהי הצלע הלא שימושית הראשונה במסלול*, אז הוא מסלול שאינו מזערי, אך*

*הוא אכן מסלול מזערי (מתקיים באופן ריק אם 𝛼=1) . לכן:*

*יהי המקיים :*

*קיים כזה מאחר ו-*  צלע לא שימושית*, ולכן*  *.*

*נניח בשלילה כי ישנן צלעות לא שימושיות נוספות במסלול.*

תהי הצלע הלא שימושית הבאה במסלול*, כלומר המסלול הוא מסלול שבו כל הצלעות שימושיות, כלומר מזערי, ולכן:*

*אך הוא מסלול שאינו מזערי, אז קיים המקיים :*

*קיים כזה מאחר ו-*  צלע לא שימושית*,*  *.*

*נקבל אם כך:*

*נגדיר מסלול העובר דרך מסלול מזערי (לא עובר דרך צלע ) ומסתיים בצלע עבורו יתקיים:*

*מאחר ו בהכרח:*

*קיבלנו כי קיים מסלול שאינו מזערי אך הוא קל יותר מהמסלול בו יש שתי צלעות לא שימושיות, לכן המסלול לא מסלול כמעט מזערי, ולכן גם המסלול (בין אם מכיל צלעות לא שימושיות נוספות או לא) אינו מסלול כמעט מזערי*

*().*

*הגענו לסתירה ונסיך מכך כי לא יתכן שבמסלול כמעט מזערי תהיה יותר מצלע לא שימושית אחת.*

*לסיכום, במסלול כמעט מזערי תהיה בדיוק צלע לא שימושית אחת.*

*סעיף ד –*

*המסלול הוא כמעט מזערי כאשר היא הצלע הלא שימושית היחידה בו, לכן כל שאר הצלעות של מסלול זה הן שימושיות:*

*כלומר הוא מסלול המכיל צלעות שימושיות בלבד – ולכן לפי סעיף א' מזערי.*

*וכן הוא מסלול המכיר צלעות שימושיות בלבד – ולכן מאותה סיבה מזערי.*

*סעיף ה' –*

*נובע מסעיפים קודמים שמסלול כמעט מזערי, הוא מסלול שמלבד צלע אחת שאינה שימושית, כל שאר הצלעות שבו שימושיות. כלומר, נבדוק את כל המסלולים מs לt כשכל צלע בהם שימושית למעט אחת.*

*המינימלי מביו מסלולים אלו, הוא המסלול הכמעט מזערי אותו אנו מחפשים.*

*נשתמש באלגוריתם של דייקסטרה למציאת מסלולים קלים ביותר, כדי לבדוק מהו הערך המינימלי של כל מסלול מs לt שיש בו צלע לא שימושית בדיוק אחת.*

***האלגוריתם:***

*נדאג במימוש שלכל קודקוד u בגרף יהיו התכונות ds(u), dt(u).*

*נפעיל את אלגוריתם דייקסטרה מהקודקוד s ועד ל t, ונשמור את ערכי המשקלים המינימליים של כל קודקוד u בתכונה ds(u).*

*נפעיל את אלגוריתם דייקסטרה בכיוון ההפוך מהקודקוד t ועד ל s, ונשמור את ערכי המשקלים המינימליים של כל קודקוד u בתכונה dt(u).*

*נגדיר משתנה ונגדיר מצביע לא מאותחל .*

***לכל*** *צלע בגרף*

***אם***

***אם*** *אינו מאותחל*

*הודעה "אין מסלול כמעט מזערי בגרף"*

***אחרת****, נחזיר את השרשור .*

***הוכחת נכונות:***

*נוכיח כי עבור כל גרף קשיר G האלגוריתם יחזיר את מסלול כמעט מזערי בו מ-s ל-t.*

*יהי G גרף קשיר, s קודקוד ההתחלה וt קודקוד הסיום במסלול. הרצת האלגוריתם של דייקסטרה מs לt תשמור לכל קודקוד את המרחק המינימלי שלו מs. הרצת דייקסטרה בכיוון ההפוך תשמור לכל קודקוד את המרחק המינימלי שלו מt.*

*כלומר, הפעלת דייקסטרה בשני הכיוונים תשמור לכל קודקוד את אורך המסלול המזערי עד אליו ואורך המסלול המזערי ממנו והלאה.*

*כעת נעבור להסתכל על הצלעות, כל צלע מחברת בין שני קודקודים, וידוע כי במסלול כמעט מזערי, כפי שהראנו בסעיפים קודמים, קיימת בדיוק צלע אחת שאינה שימושית, נסיק מכך שני דברים:*

1. *הצלע היא לא חלק במסלול המזערי, משקל המסלול כולו יהיה בהכרח גדול (ממש) ממשקל המסלול המזערי*
2. *הביטוי מבטא את משקל המסלול שבו רק הצלע אינה מזערית.*

*ממסקנות אלה נובע:*

*במידה ולאף צלע בגרף הביטוי לא יתקיים, הרי שאין צלע לא שימושית יחידה במסלול מs לt בכל הגרף, ולכן אין מסלול כמעט מזערי, והאלגוריתם יחזיר הודעה בהתאם.*

*במידה וקיימת צלע לא שימושית , האלגוריתם יבחר את הצלע שהביטוי הוא מינימלי עבורה, ולכן היא צלע לא שימושית דרכה עובר מסלול כמעט מזערי בגרף.*

*האלגוריתם שומר את הצלע עבורה הביטוי הוא מינימלי ומחזיר את שרשור המסלול הקצר ביותר מs עד הצלע, אותה, ואת המסלול הקצר ביותר ממנה עד ל t, קרי, מסלול כמעט מזערי בגרף, כנדרש.*

***זמן ריצה:***

*האלגוריתם של דייקסטרה פועל בזמן , מופעל בדיוק פעמיים באלגוריתם.*

*מתבצעות מספר קבוע של בדיקות על כל הצלעות בגרף, לכן .*

*החזרת מסלול כמעט מזערי מs ועד לצלע הלא שימושית שמצאנו וממנהעד לt . סה"כ זמן ריצת האלגוריתם , כנדרש.*

**שאלה 2**

*נבנה אלגוריתם שבודק בעץ T לאחר הסרת צלע e\* מהם שני רכיבי הקשירות הקיימים ומוצא את העלות המינימלית לקישור שלהם.*

***האלגוריתם:***

*נסיר מהעץ את הצלע e\* ונשמור מצביעים לשני הקודקודים u1,u2 שe\* חיברה בינהם.*

*נבצע סריקה BFS של העץ מקודקוד u1, ונקבל את הקודקודים ברכיב הקשירות של u1- נוסיף להם שדה צבע: כחול.*

*שאר הקודקודים ברכיב הקשירות של u2. נוסיף להם שדה צבע: אדום.*

*נגדיר משקל צלע מינימלי - ∞, ומצביע לא מאותחל.*

***כל עוד*** *יש בגרף G' צלע שלא עברנו עליה,*

***אם*** *הצלע מקשרת בין שני קודקודים השייכים לקבוצות שונות (באמצעות בדיקה של שדות הצבעים שלהם).*

***אם*** *משקל הצלע קטן מהמינימלי שנמצא עד לרגע הבדיקה,*

*נשנה את המינימלי להיות גודל הצלע הנוכחית, ונשמור את המצביע אליה.*

*נוסיף את הצלע שאליה מצביע המצביע לעץ T'.*

***הוכחת נכונות:***

*(\*) עץ הוא גרף קשיר ללא מעגלים, לכן ברגע שנוציא ממנו צלע, בהכרח נקבל גרף המורכב משני רכיבי קשירות שונים שאין בינהם צלע מחברת – נצבע קבוצה אחת בכחול ושניה באדום כדי לבצע "בדיקת שייכות" קלה בהמשך.*

*נסתכל על קודקוד u1, סריקת הBFS תחזיר את כל רכיבי הקשירות של u1, ונסמן קבוצה זו ב-S.*

*V-S היא קבוצת הקודקודים שנמצאים ברכיב הקשירות של u2.*

*בגרף G' מאחר והוא קשיר, קיימת לפחות צלע אחת בין איבר כלשהו לבין איבר כלשהו .*

*כל אחת מהקבוצות S, V-Sמכילה קודקודים שונים (אם היה קודקוד אחד לפחות המופיע בשתי הקבוצות אז הן קשירות וזו סתירה ל(\*)), אינה ריקה ואינה שווה לשניה בשלמותה (קודקוד אחד בלבד מהקודקודים u1,u2 מופיע בכל קבוצה).*

*מטענה 4.17 בספר עמ' 157 מתקיימים התנאים לכך שקיימת e=(v,w) בעלת משקל מינימלי, שכל עץ פורש מינימלי מכיל אותה, לכן כשנכניס ל את הצלע e נקבל עץ פורש מינימלי T'.*

*מציאת הצלע e מתבצעת באמצעות מעבר על כל הצלעות ובדיקה של המינימלי מבין הצלעות המחברות בין בין שני קודקודים מקבוצות שונות.*

***זמן ריצה:***

*נשים לב שבגרף קשיר , לכן:*

*הסרת צלע e\*=(u1,u2) מעץ T – .*

*הכנסת צלע לעץ T' מתבצעת ב .*

*זמן ריצת BFS - .*

*זמן צביעת כל הקודקודים -*

*הלולאה עוברת על כל הצלעות ב G', מבצעת מספר קבוע של בדיקות בכל אינטרציה - .*

*סה"כ האלגוריתם רץ בסיבוכיות , כנדרש.*

***שאלה 3***

*נבחר נוסחה שבה הליטרלים מופיעים יותר פעמים מהליטרלים אך יהיה הכרח לכך שלפחות אחד מהליטרלים יהיה אמת.*

*הנוסחה:*

*עבור הנוסחה האלגוריתם יחזיר לכל :*

*מופיע 3 פעמים xi ופעמיים ¬xi, לכן יבצע השמה .*

*אך הפסוקית אינה מסופקת ולכן הנוסחה לפי השמת האלגוריתם אינה ספיקה.*

*אך קיימת ההשמה:*  עבורה:

*בכל פסוקית קיים לפחות ליטרל אחד (מסומן בצהוב) שמחזיר ערך אמת, ולכן כל פסוקית מסופקת ולכן עבור השמה זו הנוסחה ספיקה, כנדרש.*

***שאלה 4***

*יהיה T עץ בינארי לחלוטין.*

*נגדיר סדרת שכיחויות לכל עלה של T, , כאשר d הוא עומק אותו עלה בT.*

*לכן מתקיים , כלומר, סדרת השכיחויות היא סדרה עולה.*

*בכל אינטרציה של אלגוריתם הופמן על עץ הופמן T' , שני העלים בעלי השכיחות הנמוכה ביותר מוסרים מהעץ, לאחר שסכום השכיחויות שלהם מושם כשכיחות ההורה שלהם (שלאחר הסרתם הופך לעלה).*

*הפעולה תחזור על עצמה שוב ושוב עד שנגיע לשורש, שסכום שכיחויות בניו הם 1.*

*באותו אופן, בכל עץ מושרש בינרי לחלוטין T, אם נסיר כל שני עלים אחים שקיימים בו, בהסרתם יהפוך ההורה שלהם להיות עלה, ומאחר ודרגתו קטנה ב-1 מדרגתם, נקבל כי שכיחותו היא למעשה סכום השכיחויות שלהם .*

*פעולה זו מקבילה לחלוטין לפעולה המתבצעת בכל אינטרציה של אלגוריתם הופמן.*

*ואכן, גם כאן נוכל להמשיך באופן רקורסיבי ולהסיר כל שני עלים אחים עד שנגיע לשורש, ששכיחות בניו (שדרגתם 1) היא .*

*מאחר ו- T ו-T' שניהם עצים בינאריים לחלוטין, בעלי אותה כמות עלים, מתבצעת עבורם אותה פעולה של מחיקת עלים והשמת סכום שכיחותם בהורה, והתוצאה זהה (נותרנו עם שורש בלבד ששכיחותו 1), הרי שבהכרח ישנו עץ הופמן של הסדרה שהוא זהה לעץ T.*