לירון שלי

313576209

אלגוריתמים 20417

14/11/19

**ממ"ן 11**

שאלה 1

נסמן את קבוצת הספינות

וקבוצת הנמלים

נוכיח באינדוקציה כי תמיד אפשר למצוא קבוצת לוחות זמנים כזו.

עבור n=1 כמובן שהנדרש מתקיים, מגיעה ל ונשארת בו עד סוף החודש.

נניח כי הטענה מתקיימת עבור n=k כלשהו.

נוכיח כי הטענה מתקיימת עבור n=k+1.

נניח כי אחת הספינות של חברת הספנות, ספינה היא הספינה האחרונה מגיעה לנמל (כלשהו) בחודש (לאחר מכן לא יכנסו לנמל זה ספינות נוספות עד החודש הבא).

לכן, אם נשאיר את בנמל לצרכי תחזוקה, נהיה בטוחים כי לא צפויות להגיע ספינות נוספות לנמל, והספינה לא תגיע לנמלים נוספים.

לפי הנחת האינדוקציה, קיים לוח זמנים מקוצץ מתאים עבור שאר k הספינות, וk הנמלים.

במידה והמסלול של הספינה (שאינה חלק מk הספינות) נכנס לנמל (שהוא כן חלק מk הנמלים) ואמורה להיות בנמל זה ספינה אחרת מתוך k הספינות, נשלח את במקום לנמל אל הנמל (שאינו חלק מk הנמלים) וכך נהיה בטוחים שאין שתי ספינות הנכנסות בו זמנית אל אותו נמל, כנדרש.

בכך הוכחנו את צעד האינדוקציה, כלומר הטענה מתקיימת לכל n.

**אלגוריתם:**

מבוסס על אלגוריתם השידוך היציב.

כל הספינות וכל הנמלים לא מזווגים (חופשיים).

הספינות מופיעות ברשימת העדפת הנמלים לפי סדר הגעתן מהסוף – כלומר, הספינה שמגיעה אחרונה אל נמל, נמצאת עבור הנמל בעדיפות הראשונה.

הנמלים מופיעים ברשימת ההעדפות של הספינות באופן הפוך – כלומר, סדר ההעדפה של הספינות הוא מהנמל אליו אמורות להגיע קודם עד לנמל האחרון.

**כל עוד** ספינה חופשיה ואין נמל בו היא אמורה לבצע תחזוקה נבצע באופן רקורסיבי את הפעולות הבאות:

בוחרים נמל, ומזווגים לו את הספינה האחרונה בחודש שאמורה להגיע אליו.

**אם** הספינה לא זווגה לנמל אחר קודם,   
אז יוצרים את הזיווג בין הנמל והספינה.

**אחרת**, הספינה זווגה כבר לנמל אחר,  
אז הספינה "תבחר" מבחינת לוח הזמנים את הנמל הבא ברשימת ההעדפות שלה, ונבצע זיווג בין שניהם, כשהנמל הנותר יחזור להיות חופשי.

**הוכחת נכונות:**

בכל אינטרציה נמל כלשהו "מבקש" לזווג לו ספינה, שאחריה לא אמורה להגיע אף ספינה נוספת אל אותו נמל (כך נבחר סדר ההעדפות שלו).

במידה ואותו נמל מקבל "סירוב", אז לספינה אותה ביקש לזווג לעצמו, יש נמל אחר אליו היא מגיעה קודם ובו היא מבצעת את התחזוקה, ולכן בכל מקרה לא תגיע אל אותו נמל מאחר והוא מקוצץ מלוח הזמנים שלה.

וכך נשמרת התכונה הראשונה בכל איטרציה, שאף ספינה לא אמורה להגיע אחרי אף ספינה שעומדת בראש סדר ההעדפות הנוכחי של כל נמל.

לבסוף נקבל קבוצת לוחות זמנים מקוצצים בהן כל ספינה מגיעה אל נמל בו היא מבצעת תחזוקה עד סוף החודש, ולכל נמל לא מגיעה אף ספינה מהיום בו ספינה החלה בו תחזוקה.

**זמן ריצה:**

האלגוריתם יוצר רשימת העדפות עבור n הנמלים – לכל נמל דירוג העדפה של n ספינות, ועבור n הספינות -דירוג העדפה של n נמלים – סה"כ .

האלגוריתם "משדך" לכל n הנמלים ספינה, ובמקרה הגרוע הספינה "תפוסה" ע"י נמל אחר, ובמקרה כזה לאחר ה"שידוך" עדיין יהיה נמל חופשי. ספינה יכולה להיות "תפוסה" עד n-1 פעמים, לכן זמן הריצה הוא .

שאלה 2

נשים לב שניתן לכוון גרף במידה וכל חלק קשיר בגרף מכיל מעגל, כך שדרגת הכניסה של כל קודקוד תהיה גדולה מ-0 (דבר המבטיח לפחות n צלעות על n קודקודים באותו חלק קשיר של הגרף, הכרחי מאחר ויש כיוון יחיד לכל צלע, לא אפשרי אם יש יותר קודקודים מצלעות – היפוך טענה 3.2 בעמ' 84).

**אלגוריתם:**

נסתכל בנפרד על כל חלק קשיר בגרף, אם החלק הקשיר הוא גרף פשוט שקיים בו מעגל (או מעגלים).

נבחר לכל זוג קודקודים כיוון שממנו נמשיך הלאה עד להקפת המעגל, לכל מעגל בגרף בן n קודקודים נבחר את הכיוון כך:

(e1,e2) , (e2,e3) , … , (e(n-1),en) , (en,e1)}}.

במידה ויש קודקודים נוספים שאינם חלק ממעגל, מאחר והם חלק קשיר למעגל, הכיוון יהיה מקודקודים שהם חלק במעגל (שדרגת הכניסה שלהם גדולה מאפס) החוצה והלאה עד שנגיע לקודקוד שלא יוצאת ממנו צלע אל קודקוד נוסף.

במידה ויש צלעות נוספות, אין חשיבות לכיוון שנבחר ולכן פשוט נבחר בכיוון כלשהו, מאחר וכבר דאגנו לכך שצלע אחת תכנס לכל קודקוד.

לסיום נקבל גרף מכוון שלכל צלע בו יש לפחות דרגת כניסה אחת, כנדרש.

**הוכחת נכונות:**

במעגל המכוון לכל n הקודקודים יש דרגת כניסה גדולה מ0, ולכל קודקוד נוסף הקשיר למעגל, הצלע המקשרת בינו לבין שאר הגרף, מוכוונת אליו, כלומר דרגת הכניסה שלו לפחות 1, כנדרש.

חסר ניתוח זמן ריצה

שאלה 3

ניצור גרף G מכוון בהתאם ל y :

ניצור n2 קודקודים בהתאם לn הליטרלים הקיימים בנוסחא:

קודקוד אחד עבור ליטרל xi וקודקוד שני עבור ליטרל xi¬.

בהתאם לכל פסוקית בעלת 2 ליטרלים: ניצור  *וגם , כדי לוודא שנקבל פסוקית מסופקת.*

**אלגוריתם:**

**כל עוד** קיים קודקוד שלא קיבל ערך אמת או שקר,

נפעיל סריקה לעומק מהקודקוד x הנבדק

**אם** לא הופיע בסריקה

**אם** ישנו קודקוד אחר בסריקה שגם הנגדי שלו בסריקה – מסירים אותו ואת הנגדי שלו מהסריקה.

**אחרת,** נגדיר ערך אמת לכל הקודקודים שמופיעים בסריקה, ובהתאם, שקר בהתאם לכל הקודקודים הנגדיים שלהם.

**אחרת,** נפעיל סריקה מהקודקוד

**אם** לא הופיע בסריקה

**אם** ישנו קודקוד אחר בסריקה שגם הנגדי שלו בסריקה – מסירים אותו ואת הנגדי שלו מהסריקה.

**אחרת,** נגדיר ערך אמת לכל הקודקודים שמופיעים בסריקה, ובהתאם, שקר בהתאם לכל הקודקודים הנגדיים שלהם.

**אחרת,** נדווח שהנוסחה אינה ספיקה.

(נגיע לסתירה מאחר וx תלוי ב, ולעולם שניהם לא יהיו ספיקים.)

נחזיר את ההשמה הנדרשת בהתאם לליטרלים - אמת/שקר.

**הוכחת נכונות:**

הלולאה תסתיים רק כאשר עברנו על כל הקודקודים בגרף (כל הליטרלים בנוסחה) , ולכל קודקוד נמצא כי הוא לא עומד בסתירה אל מול הקודקוד הנגדי שלו.

בסריקה שנבצע עבור כל קודקוד, נקבל את כל רכיבי הקשירות שלו, ואם התנאי הנגדי שלו מתקיים במקביל אליו התנאי הנגדי יופיע כרכיב קשירות שלו.

במקרה כזה, עלינו לבדוק אם אותו קודקוד גם מופיע באחד מרכיבי הקשירות בסריקה של הקודקוד הנגדי לו.

במידה וכן, אז תנאי x אמת מחייב (בהתאם לשאר הרכיבים התלויים בו) שתנאי x¬ שקר, ותנאי x¬ אמת (בהתאם לשאר הרכיבים התלויים) דורש שתנאי x שקר, אז הגענו למצב בו יהיו שני רכיבים תלויים בפסוקית שלעולם לא יהיו אמת, מכאן שהפסוקית לא תהיה מסופקת ולכן הנוסחה אינה ספיקה.

במידה ולפחות אחד מהתנאים אינו תלוי בתנאי הנגדי שלו, אז נוכל לדעת בוודאות כי ניתן להגדיר את כל רכיבי הקשירות במסלול להיות ספיקים – "אמת" לכל הליטרלים המופיעים ברכיב הקשירות ו"שקר" לליטרלים הנגדיים להם.

במידה וישנם רכיבי קשירות המנוגדים זה לזה, הם יוגדרו בהתאם בעת ביצוע הפעולה עבור אותם קודקודים "בעייתיים", כך שאם הוגדרו קודם כאמת/שקר ההגדרה שלהם תשמר, ואם לא הוגדרו, יוגדרו בהמשך (לא עוברים כל שינוי).

במידה ולא התגלה כי הנוסחה אינה ספיקה, האלגוריתם יסתיים לאחר שבוצעה השמה על כל הקודקודים לפיה לא קיימות סתירות, כלומר – הנוסחה ספיקה.

**זמן ריצה:**

האלגוריתם בונה לכל משתנה שני קודקודים – 2n , בונה לכל פסוקית שתי קשתות – 2m, ולכן בניית גרף בעל 2n קודקודים ו2m קשתות הוא *.*

*הלולאה החיצונית תתבצע לכל היותר 2n פעמים, כאשר בכל פעם היא תבצע השמה על קודקוד אחד בלבד (על עצמה).*

*או, שהלולאה תתבצע פחות מ 2n פעמים, כאשר תבצע סריקה על d קודקודים, ולאחר מכן השמה על אותם קודקודים (אליהם לא תצטרך לחזור), ולכן לאחר השמה זו נותרו 2n-d קודקודים לבדיקה של הלולאה, כלומר, תוחלת זמן ביצוע הלולאה החיצונית הוא O(n).*

*מכאן, שזמן ביצוע האלגוריתם כולו הוא .*

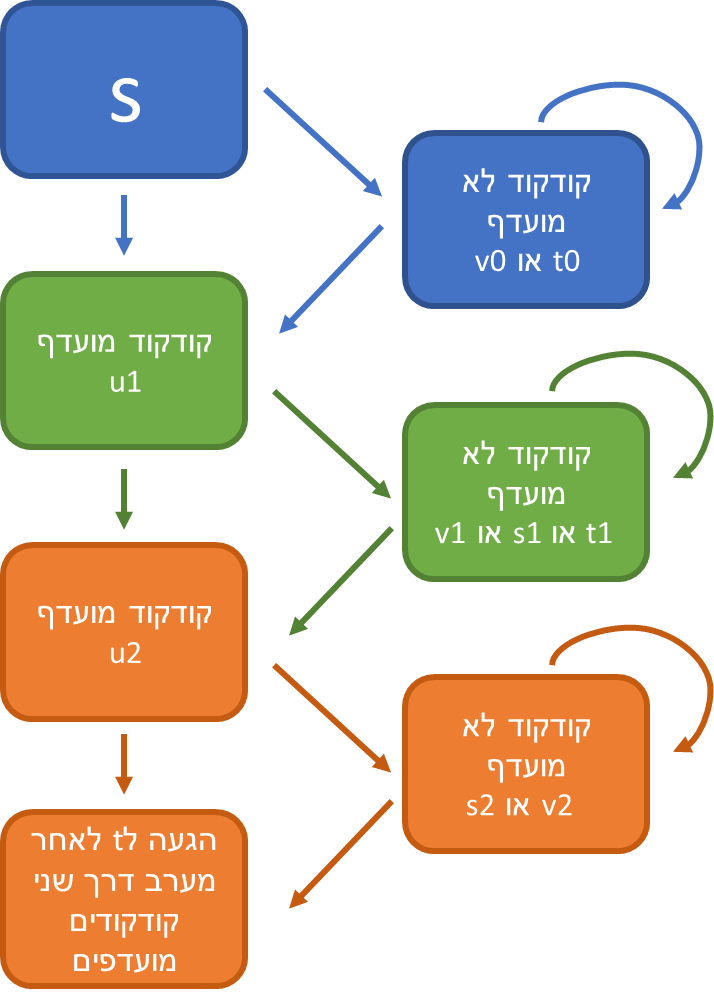
*שאלה 4*

*נבנה גרף מכוון G' המורכב מהצמתים s0,s1,s2, t0,t1,t2,*

*לכל u בG יהיו u1,u2 ב G'.*

*לכל v בG יהיו v0,v1,v2 ב G'.*

*כל מסלול שניצור ב G' , נוודא שעברנו דרך בדיוק 2 קודקודים מועדפים בדרך הבאה:*

**

*תחילה s יוביל אותנו אל t0 או v0 (קודקוד שאינו מועדף) או אל u1.*

*קודקוד לא מועדף מוביל אל t0 או v0 (קודקוד לא מועדף), או אל קודקוד מועדף u1.*

*קודקוד מועדף מוביל אל t1 או v1 או s1 (קודקוד לא מועדף), או אל קודקוד מועדף u2.*

*קודקוד לא מועדף מוביל אל t1 או v1 (קודקוד לא מועדף), או אל קודקוד מועדף u2.*

*קודקוד מועדף מוביל אל v2 או s2(קודקוד לא מועדף), או אל t2 (כלומר קודקוד t לאחר שביקר בדיוק פעמיים בקודקודים מועדפים).*

***אלגוריתם***

*נקח את הגרף G ולפיו נבנה את G':*

*נבנה קודקודים s0, s1, s2, t0, t1, t2 בגרף G'.*

*לכל קודקוד מועדף u בגרף G ניצור שני קודקודים u1, u2 בגרף G'.*

*לכל קודקוד v שאינו מועדף, אינו קודקוד s ואינו קודקוד t ניצור שלושה קודקודים v1, v2, v3 בגרף G'.*

*כדי להימנע מכתיבה חוזרת, נסמן: x={s,t,v}, y={s,t,v} כאשר x ו-y אינם אותו קודקוד בG .*

*לכל קשת (x→y) בG ניצור בG' את הקשתות (x0→y0), (x1→y1), (x2→y2).*

*לכל קשת (x→u) בG ניצור בG' את הקשתות (x0→u1), (x1→u2).*

*לכל קשת (u→x) בG ניצור בG' את הקשתות (u1→x1), (u2→x2).*

*נריץ סריקה BFS בG' על קודקוד s0, בתוצאת הסריקה נראה מתי מופיע t2, המסלול בין s0 לt2 הוא המסלול הקצר ביותר בגרף G.*

*במידה ולא נמצא t2 בסריקה אז לא קיים מסלול כנדרש.*

**הוכחת נכונות**

*כל מסלול מתחיל בs בG ובs0 בG' .*

*כל מסלול מסתיים ב t בG לאחר שעבר דרך 2קודקודים מועדפים (או דרך אותו קודקוד פעמיים) ומסלול מסתיים בt2 בG' , לאחר שעבר דרך u1 וu2 (יתכן של אותו קודקוד מועדף בG או של שני קודקודים מועדפים שונים).*

*כל קודקוד שאינו קודקוד מועדף x0 יכול להוביל לקודקוד שאינו מועדף אחר (לא לחזור על עצמו, כי אז אנחנו מאריכים את המסלול ללא תועלת), או לקודקוד מועדף.*

*ברגע שהגענו לקודקוד מועדף, "עלינו שלב", ולכן הוא מוביל לקודקוד x1 שאינו מועדף, או לקודקוד מועדף נוסף. לא ניתן להגיע לשלב זה מבלי לעבור דרך קודקוד מועדף.*

*במידה והגענו לקודקוד שאינו מועדף, נמשיך עד למציאת קודקוד מועדף שני.*

*האפשרות השניה שיש צלע מכוונת ישירה לקודקוד המועדף השני מהראשון.*

*בשני המקרים אנחנו מקדמים "שלב" נוסף.*

*גם לשלב זה לא ניתן להגיע לפני שעברנו דרך בדיוק שני קודקודים מועדפים.*

*כעת לא ניתן להגיע שוב אל קודקודים מועדפים (כי אז יהיו 3 כאלה במסלול וזה בניגוד לדרישה), ולכן ישנה אפשרות של מעבר בין קודקודים לא מועדפים בלבד, על להגעה לקודקוד המסיים את המסלול – t2.*

*האלגוריתם BFS מבטיח את החזרת המסלול הקצר ביותר בין s0 לt2, מאחר והוא סורק את הגרף לפי שכבות.*

*הגרף G' נבנה על סמך הגרף G, ניתן לשחזר את גרף G דרך G' באמצעות איחוד הצמתים x0,x1,x2 ואיחוד u1,u2, ושחזור הקשתות בהתאם. לכן כל מסלול בG' בוודאות קיים בגרף G.*

*לכן, אותו מסלול קצר ביותר בG' הוא המסלול הקצר ביותר שקיים בG מs לt.*

***זמן ריצה***

*מבצעים סריקה של הגרף ובניה של גרף חדש על סמך אותה סריקה, כשגודל הגרף החדש הוא עד פי 3 קודקודים ועד פי 3 קשתות מהגרף המקורי, לכן .*

*לאחר מכן מפעילים את האלגוריתם BFS שזמן הריצה שלו .*

*מכאן שזמן הריצה הכולל הוא .*