גירסה 1.00 – 25.12.2002

<u>אלגוריתמים בתורת הגרפים – חלק שלישי</u>

מסמך זה הינו השלישי בסדרת מסמכים אודות תורת הגרפים, והוא חופף בחלקו לקורס "אלגוריתמים בתורת הגרפים" בטכניון (שאינו מועבר יותר).

ברצוני להודות תודה מיוחדת **לפרופסור שמעון אבן ז"ל**, על העזרה העצומה שעזר לי בהבנת החומר ובשאלות רבות עליהן ענה לי כאשר למדתי את הנושא. שמעון אבן היה מרצה שהערכתי מאוד ותרומתו למסמך זה היתה משמעותית.

מקורות:

- המסמך מבוסס במידה רבה על הרצאותיו של פרופסור שמעון אבן משנת 2002 וכן על שאלות רבות עליהן שמעון אבן ענה לי.
- ספרו של שמעון אבן מ-1979, Graph Algorithms
- פרקים מספרו החדש של שמעון אבן, שפורסמו בשנת 2002
- חלקים מהרצאות וידאו של **פרופסור ראובן בר יהודה** בנושא אלגוריתמים בתורת הגרפים.

המסמכים כוללים נקודות רבות ממקורות אלו, ובנוסף תובנות רבות, דוגמאות ורעיונות שאספתי במהלך לימוד הקורס.

ניר אדר

חיפוש לעומק – Depth First Search

<u>כללי</u>

DFS זוהי טכניקת סריקה של גרף לא מכוון וסופי. היא נחשבת טכניקה חזקה לפתרון בעיות שונות בתורת הגרפים.
מקור הטכניקה הוא במאה ה-19 בה שימשה להתמודדות מול בעיות מבוכים.

הבעיה:
נתון גרף סופי וקשיר G(V, E).
תהי צומת התחלה כלשהי, ממנה נתחיל ללכת על קשתות, מצומת לצומת, עד שנבקר את כל הצמתים.
אנו מחפשים אלגוריתם שיבטיח שאנו סורקים את כל הגרף, וכן שתהיה לנו אינדיקציה שסיימנו לסרוק את הגרף, מבלי לדעת כמה גדול הוא הגרף. (מבחינת כמות צמתים וקשתות).

הגבלות:
אנו פועלים ללא תכנון מוקדם, למשל למידת "מפת הדרכים" של הגרף לפני התחלת הסיור, אנו חייבים לקחת החלטה אחר החלטה, מכיוון שאנו מגלים את מבנה הגרף רק תוך כדי הסריקה.

נכנה בשם **מעברים** את החיבורים בין הקשתות לצמתים.
על מנת שנוכל למצוא אלגוריתם שיפתור בעיות זו, אנו צריכים להשאיר "**סימונים**" על המעברים כשאנו נעים, כדי שנוכל לזהות בעתיד שהגענו למקום בו כבר היינו.

<u>Tremaux's Algorithm</u>

<u>נתון</u>

גרף לא מכוון, קשיר וסופי G(V, E) עם צומת התחלה s.

נגדיר מספר סוגים של סימונים: מעבר יכול להיות לא מסומן, מעבר יכול להיות מסומן באות F, המסמן שדרך מעבר זה נעשתה הגישה הראשונה אל הצומת וכמו כן כל מעבר אחר בו נשתמש כדי לעזוב את הצומת יסומן ב-E.
כמו כן, איננו מוחקים או משנים סימונים שכבר סומנו.

<u>מטרה</u>

נרצה לצאת מצומת ההתחלה, ולסרוק את כל צמתי הגרף דרך הקשתות.
כמו כן, מותר להשאיר סימונים במעברים.

האלגוריתם

1) $v \leftarrow s$
2) כל עוד קיים מעבר בלתי מסומן ב-v או קיים מעבר מסומן ב-F בצע:
3) אם קיים מעבר לא מסומן $v \overset{e}{—} u$ אזי בצע:
4) סמן את המעבר של e בצומת v ב-E.
5) אם אין ב-u מעבר מסומן (צומת חדש) בצע
6) סמן את המעבר של e ב-u ב-F.
7) $v \leftarrow u$
8) אחרת - u צומת ישן - סמן את המעבר של e ב-u ב-E.
9) אחרת (אין איברים לא מסומנים) בצע
10) השתמש בקשת שהמעבר שלה בצומת v מסומן ב-F (לכל היותר 1).
11) $v \leftarrow u$

למה 1

באלגוריתם של Tremaux כל קשת נסרקת לכל היותר פעם אחת בכל כיוון.

הוכחה

אם בפעם הראשונה המעבר מ-v אל u ב-$u — v$ מסומן ב-E, לא תהיה יותר כניסה דרך מעבר זה.

נניח שקיימות קשת שנסרקות יותר מפעם אחת ממעבר מסומן F. תהיה $u — v$ קשת ראשונה שנסרקה פעמיים. מתקיים כי $s \neq v$ (כי s לא מסומן ב-F). מכאן: יצאנו מ-v מספר פעמים $d(v)+1$. כרגע אנו לא נמצאים ב-v, כי יצאנו ממנו וכמו כן גם לא התחלנו ב-v. $\Leftarrow$ ישנה צומת שדרכה נכנסנו פעמיים.

סתירה לכך שהקשת $u — v$ היא הראשונה שנסרקה פעמיים.

טענה

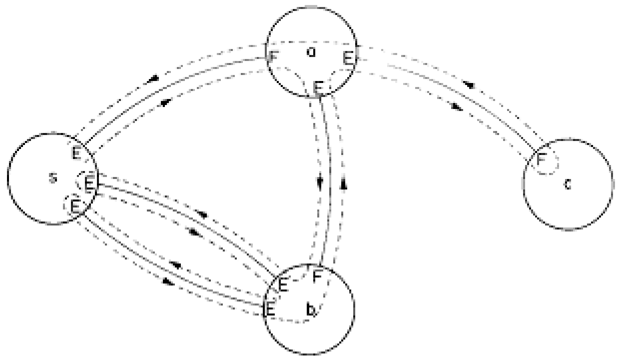
האלגוריתם של Tremaux תמיד מסתיים.

למה 2

עם סיום האלגוריתם של Tremaux כל קשת בגרף נסרקה פעם אחת בכל כיוון.

ניסוח נוסף: עם סיום האלגוריתם, עבור כל צומת מתקיים התנאי שכל קשת הפוגעת בו נסרקת פעם אחת בכל כיוון.

המחשה:



רקורסיה

ניתן גם לממש את האלגוריתם בצורה רקורסיבית, בצורה הבאה:

DFS(x)

סמן את x כצומת ישן.

עבור כל קשת $x — y$, אם y חדש אז DFS(y).

נשים לב לעובדה הבאה:

אם נשמור את השכנים כל פעם במחסנית (LIFO), הרי שמימשנו DFS.
לעומת זאת, אם נשתמש בתור (FIFO), הרי שמימשנו BFS.

הגדרה

במהלך סריקת הגרף - כאשר אנו מגיעים מצומת v אל צומת חדש u, נקרא לצומת v בשם **אב**, ואילו לצומת u נקרא **בן**.

האלגוריתם של Hopcraft & Tarjan - DFS

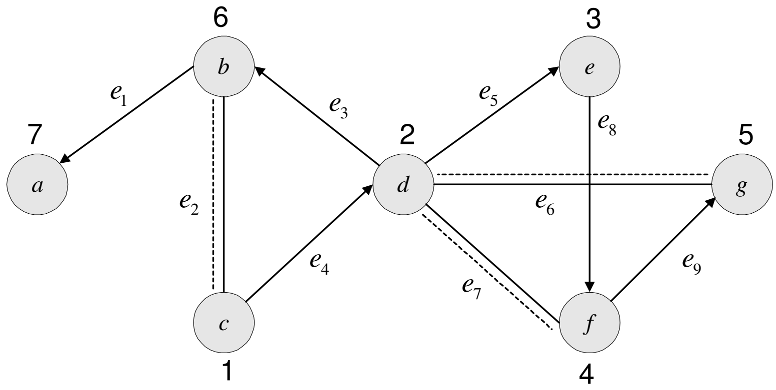
באלגוריתם זה אנו מסמנים גם קשתות וגם צמתים.
k(v) - מספור של כל צומת המתאים מספר לכל צומת. המספרים ("השמות") של הצמתים ינתנו לצמתים על פי סדר הגילוי.
בתחילת האלגוריתם, $\forall v \in \mathrm{V}, k(v) = 0$. במהלך ריצת האלגוריתם, אם $k(v) > 0$ אזי הצומת ידוע, כי בכל פעם שעוברים דרך צומת חדש מקבל הצומת סימון יחיד.
כמו כן, בתחילת האלגוריתם, $\forall v \in \mathrm{V}, f(v) = \mathrm{NIL}$ - לאף צומת אין אב.
אב - דרכו התגלתה הצומת.
בתחילת האלגוריתם, כל הקשתות הן "חדשות". כל קשת שאנו עוברים דרכה הופכת להיות "ישנה" ולכן לא נוכל להשתמש בה שוב כדי לעבור לשכן.

<u>האלגוריתם</u> $\mathrm{DFS}(\mathrm{G}, \mathrm{s}; k(v), f(v))$

1) לכל $e \in E$ סמן את e "חדשה".
2) לכל $u \in V$ בצע:
3) $k(u) \leftarrow 0$
4) $f(u) \leftarrow \mathrm{NIL}$
5) $v \leftarrow s$
6) $k(v) \leftarrow 1$
7) $i \leftarrow 2$
8) כל עוד יש ל-v קשת חדשה, או $f(v) \neq \mathrm{NIL}$ בצע:
9) אם ל-v יש קשת חדשה $v \overset{e}{-\!-\!-} u$ אזי בצע:
10) סמן את e כ-"ישנה".
11) אם $k(u) = 0$ (צומת חדש) בצע:
12) $f(u) \leftarrow v$
13) $k(u) \leftarrow i$
14) $i \leftarrow i + 1$
15) $v \leftarrow u$
16) אחרת: $v \leftarrow f(v)$.

שלבים 1-7 משמשים לאתחול.
האלגוריתם מסתיים בצומת ההתחלה, ולכן האלגוריתם עוצר.
סיבוכיות האלגוריתם: $O(E)$ - ליניארי לפי מספר הקשתות.

יצירת גרף מכוון בעזרת האלגוריתם של DFS

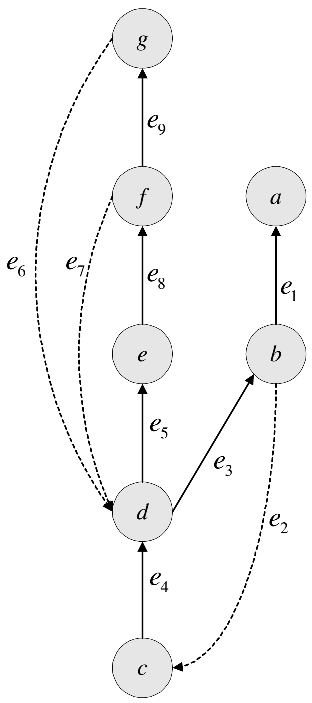


נתון $G(V,E)$ גרף סופי לא מכוון. $s \in V$ היא צומת ההתחלה. בדוגמא לעיל, $s=c$.
נכוון חלק מהקשתות בצורה הבאה: לכל $v \neq s$, $f(v) \to v$ - נכוון את הקשתות המובילות מאב לבן, כך שהקשת יוצאת מהאב אל הבן.
נסמן: E' - אוסף כל הקשתות המכוונות מאב לבן.
הגרף (V,E') הינו גרף מכוון, הפורש את צמתי הגרף המקורי.

נשים לב כי אם נריץ DFS מצומת אחרת, נקבל גרף מכוון אחר.

למה 3

הגרף המכוון (V,E') מהווה עץ מכוון פורש של הגרף המקורי $G(V,E)$.
וכן $d_{in}(s)=0$ $\forall v \neq s, d_{in}(v)=1$ (מכיוון שיש נה רק קשת מכוונת אחת מאב לבן).
s הינו השורש של העץ המכוון.



למשל, עבור הדוגמא מתקבל העץ הבא:

קשתות כגון e_2,e_6,e_7 המחברות בין בן לאב, נכד לסבא וכו', לעולם לא יהיו חלק מהעץ הנוצר, ללא קשר לגרף, לצומת ההתחלה או ל-DFS.

למה 4

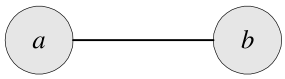
עבור כל גרף בלתי מכוון וקשיר $G(V,E)$, אם הריצו עליו DFS מצומת s, בעץ ה-DFS שהתקבל כל קשת שאינה קשת עץ מחברת בין צומת לאחד מצאצאיו.

כלומר: קשת $a—b \in E$ היא או קשת עץ DFS או קשת אחורית.

<u>הערות</u>

- מרכז הכובד עובר כאשר מתגלה <u>צומת חדש</u>. מרכז הכובד עובר מאב לבן בעת הגילוי.
- <u>נסיגה</u> - כאשר כל הקשתות מהצומת נבדקו, ואם יש לו אב מוגדר, מרכז הפעילות עובר בניגוד לכיוון הקשת.

<u>הוכחת הלמה</u>



מתקיים: $k(a) < k(b)$.

הקשת $a—b$ איננה קשת עץ. מרכז הכובד היה קודם ב-a כי $k(a)$ קטן מ $k(b)$.

הקשת $a—b$ אינה קיימת בעץ. מרכז הכובד הגיע מצומת a אל צומת b לא דרך הקשת הנ״ל, כי הרי מרכז הכובד עובר רק דרך קשתות עץ.
b איננו צאצא של a ולכן הייתה חייבת להיות נסיגה ואז מעבר אל b.

הייתה נסיגה רק לאחר שלא היו קשתות חדשות, אך הקשת $a—b$ לא נסרקה

עדיין, כלומר אנו מגלים את b בשלב זה, (b מכאן צומת חדש) - ואז הקשת $a—b$

הינה קשת-עץ, בסתירה להנחה שקשת זו איננה קשת עץ.

<u>הגדרה</u>

כל קשת שאיננה קשת עץ נקראת **קשת אחורית**. (מתחילה מהצאצא אל אב קדמון).
כיוון הקשת הנ״ל מוכתב להיות מהגבוה (צאצא) לנמוך (אב). הקשת נסרקה פעם אחת ויחידה כאשר היינו בצומת הגבוה.
כלומר: קשת אחורית מחברת צומת בעץ לאב קדמון שלו.
נעיר כי הכיוון הנ״ל תקף גם בקשתות עץ.

<u>טענה</u>

יהי x מרכז הפעילות. נסתכל על המסלול הבא: $x \to f(x) \to f(f(x)) \to ... \to s$.

נטען: מסלול זה הינו המסלול (היחיד) מ-s אל x.

האלגוריתם של Tarry לסריקת מבוך

בתחילת האלגוריתם כל המעברים אינם מסומנים.

1) $v \leftarrow s$
2) כל עוד יש ב-v מעבר בלתי מסומן או שיש ב-v מעבר מסומן ב-F, בצע:
3) אם יש מעבר בלתי מסומן לקשת v–u, בצע:
4) סמן את המעבר ב-E.
5) $v \leftarrow u$
6) אם אין מעברים מסומנים ב-u, סמן את הכניסה ל-u ב-F.
7) אחרת (קיים מעבר מסומן ב-F), בצע:
8) עבור דרך המעבר המסומן F לצומת השכן u.
9) $v \leftarrow u$

נעיר כי השוני בין Tremaux ו-Tarry הוא בעובדה שאם קיים מעבר לא מסומן, אנו נעבור בכל מקרה באלגוריתם של Tarry לצומת שבצד השני (בין אם הוא צומת חדש או לא), ואילו במקרה של Tremaux נעבור לצומת רק אם הוא חדש. הבדל נוסף: באלגוריתם של Tarry ייתכן מצב בו השתמשנו בקשת אך היא איננה מסומנת.

<u>טענה 1</u>

האלגוריתם של Tarry לא עובר על אותה קשת באותו כיוון יותר מפעם אחת.

הוכחה:
נניח בשלילה כי קיימת קשת e=(u,v) אשר עוברים עליה יותר מפעם אחת בכיוון מ-u ל-v ונסתכל על הפעם הראשונה שמאורע שכזה קורה. לא ייתכן כי המעבר שבין e ו-u מסומן על-ידי E, כי ברגע שנסמן מעבר זה ב-E, האלגוריתם לא ישתמש במעבר זה שוב (סימון של מעבר ב-E נעשה רק כאשר משתמשים בפעם הראשונה במעבר וגם האלגוריתם לא משתמש במעברים אשר מסומנים ב-E). מכאן נסיק כי מעבר זה מסומן ב-F. ל-s אין מעברים אשר מסומנים ב-F, ולכן $u \neq s$. נסיק כי השימוש במעבר שבין u ו–e הוא הפעם ה- d(u)+1 שיוצאים מהצומת u (אנו נשתמש במעבר המסומן ב-F רק אם כל שאר המעברים שנוגעים ב-u מסומנים ב-E, ודרך כל מעבר שכזה אנו יצאנו מ-u, ולכן סה״כ d(u)-1 יציאות עבור המעברים המסומנים ב-E ועוד שתי יציאות דרך המעבר שמסומן ב-F). מכיוון ש- $u \neq s$, אנו צריכים לפני כל יציאה מ-u גם להיכנס ל-u. ולכן היו קודם לכן d(u)+1 כניסות ל-u, וזאת סתירה לכך שבחרנו את הפעם הראשונה שבה עברנו על קשת כלשהי יותר מפעם אחת באותו כיוון.

<u>מסקנה</u>
האלגוריתם של Tarry עוצר בצומת s (כי לכל צומת שאינו s יש מעבר אשר נוגע בו ומסומן ב-F).

טענה 2

לאחר סיום ריצת האלגוריתם של Tarry, עבור כל צומת, עברנו על כל הקשתות שנוגעות בו בשני הכיוונים.

הוכחה

נסמן ב-S את קבוצת הצמתים אשר מקיימים את התכונה הנדרשת. ראשית נטען כי הקבוצה S לא ריקה. נראה כי $s \in S$: מכיוון שהאלגוריתם מסתיים רק ב-s, נסיק כי עברנו על כל המעברים שנוגעים ב-s, כלומר יצאנו מ-s בדיוק d(s) פעמים (לפי טענה 1). האלגוריתם מתחיל ב-s ומסתיים ב-s, ולכן גם נכנסו ל-s בדיוק d(s) פעמים. לפי טענה 1, נקבל כי עברנו על קשת שנוגעת ב-s בשני הכיוונים האפשריים.
נניח בשלילה כי $S \neq V$, אזי נתבונן בחתך: (S,V\S). חתך זה אינו ריק, כי $s \in S$, $S \neq V$ והגרף הוא קשיר. כמו-כן, כל קשת בחתך הזה עברנו עליה פעם אחת בכל כיוון. נסמן ב-e=(u,v) את הקשת שעברנו עליה ראשונה בחתך, כך ש-v נמצא ב-S ו-u נמצא ב- V\S. המעבר המשותף ל-u ו-e בהכרח מסומן ב-F. מכיוון שעברנו על הקשת e גם בכיוון ההפוך, יתקיים בהכרח כי כל המעברים אשר נוגעים ב-u מסומנים ב-E. לכן עברנו על כל הקשתות שנוגעות ב-u בכיוון החוצה מ-u (כלומר השתמשנו בכל המעברים המשותפים ל-u). מכיוון ש- $u \neq s$ (כי הנחנו ש-u נמצא ב-S\V), האלגוריתם לא התחיל מ-u ולא הסתיים ב-u, ולכן מספר הפעמים שנכנסנו ל-u שווה למספר הפעמים שיצאנו מ-u. מכאן, לפי טענה 1, נקבל כי עברנו על כל הקשתות שנוגעות ב- u בדיוק פעם אחת בכל כיוון. אבל במקרה זה היה צריך להתקיים: $u \in S$, וזאת סתירה.

טענה

אוסף הקשתות אשר מעבר אחד שלהן מסומן ב-E והמעבר השני ב-F ומכוונות מ-E ל-F, הן עץ פורש מכוון של G בעל שורש s.

תרגיל

תהי קשת e=u-v אשר אף אחד משני המעברים שלה אינו מסומן ב-F.
האם בהכרח u הוא צאצא של v או v הוא צאצא של u ? התשובה: לא.

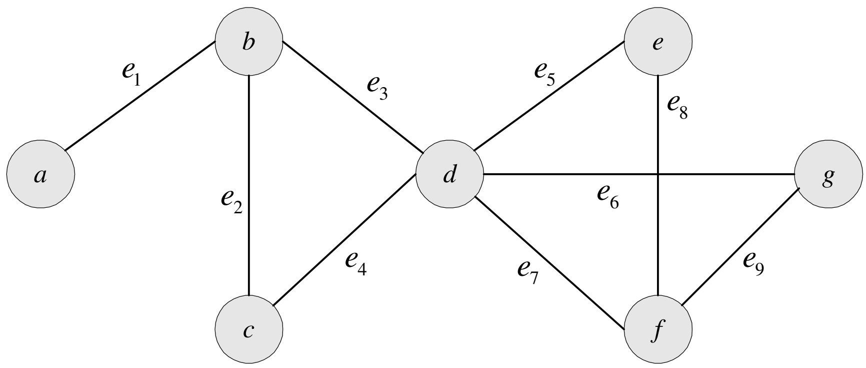
נתבונן למשל בגרף הבא:
V={s,a,b,c}, E={(s,a),(a,b),(s,b),(s,c),(b,c)}

אם נתחיל ונסייר על הצמתים בסדר הבא: s, אחר-כך a, אחר-כך b, אחר-כך שוב s ואחר כך c, אנו נגיע למצב שבו שני המעברים של הקשת (b,c) מסומנים ב-E. אף אחד מבין b ו-c אינו צאצא של השני. לכן הטענה אינה נכונה.

הערה: Tremaux הוא למעשה DFS, ולכן עבורו הטענה הייתה מתקיימת.
זהו אחד ההבדלים בין האלגוריתם של Tremaux וזה של Tarry.

שימוש ל-DFS - רכיבים בלתי פריקים וצמתי הפרדה

יהי $G(V,E)$ גרף לא מכוון, סופי וקשיר.
$v\in V$ נקרא **צומת הפרדה** (בין צמתים a,b) אם קיימים שני צמתים a,b שונים זה מזה ושונים מ-v שעבורם מתקיים כי כל מסלול המחבר בין a ל-b עובר דרך v.
דוגמא:



בדוגמא - d היא צומת הפרדה בין a ל-g.

נגדיר **מחלקות שקילות** בגרף. יהיו שני צמתים $a,b\in V$. נאמר שהצמתים a ו-b נמצאים באותה מחלקת שקילות, אם קיים מסלול ביניהם שאינו עובר דרך v.
בגרף בעל צומת הפרדה יהיו לפחות שתי מחלקות שקילות, כי V צומת הפרדה בין a,b מסויימים, כלומר כל המסלולים ביניהם עוברים בהכרח דרך v.
דוגמא: בגרף המוצג,
נחלק את $V-\{v\}$ על פי מחלקות שקילות.

גרף נקרא **גרף בלתי פריק** אם אין בו צמתי הפרדה. בהתאם, גרף הוא **פריק** אם יש לו צמתי הפרדה.
רכיב בלתי פריק - תת קבוצה $V'\subset V$ של צמתים מקסימלית - לא ניתן להוסיף לה צומת ולהשאיר אותה אי פריקה. תת הגרף של G המושרה על ידי $V'\subset V$ (נדרוש כי V' תהיה קשירה) כך ש-V' מקסימלית בהתייחס לקבוצת אי פריקות.
ניסוח נוסף (ידידותי יותר): **רכיב בלתי פריק** של גרף: תת גרף לא פריק מקסימלי (לא מקסימום!). $U\subseteq V$ רכיב בלתי פריק אם הגרף המושרה על ידי U אי פריק, וכן כל קבוצה U' המכילה את U היא פריקה.

דוגמאות לרכיבים בלתי פריקים:
1. $E'=\{e_5,e_8,e_9\}$ $\quad V'=\{d,e,f,g\}$
2. $E'=\{e_1\}$ $\quad V'=\{a,b\}$
3. $E'=\{e_2,e_3,e_4\}$ $\quad V'=\{b,c,d\}$

d,b הם שני צמתי ההפרדה היחידים בגרף המקורי.

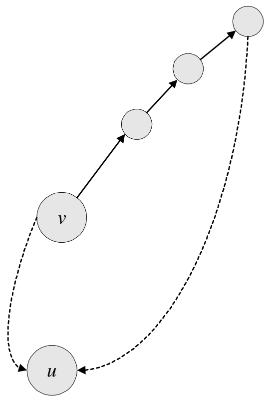
סימון והגדרה

נניח שנתון לנו גרף לו מספר רכיבים אי פריקים. נחליף כל רכיב אי פריק בצומת.
הגרף שנקבל הוא **מבנה על** של הגרף ביחס לרכיבים האי פריקים.
מבנה העל הוא עץ - חסר מעגלים וקשיר. הקשירות היא מכיוון שהגרף המקורי קשיר. נניח בשלילה שמבנה העל אינו עץ, אז ישנו מעגל לא מכוון בין הרכיבים האי פריקים השונים, אולם רכיבים השייכים למעגל פשוט שייכים לרכיב פריק אחד, ולכן הם אינם רכיבים פריקים שונים. מכאן - מבנה העל הינו עץ.
רכיב עלה הוא רכיב שבמבנה העל יש לו דרגה אחת - מחובר רק לצומת מעבר אחת.

הגדרה

נגדיר את ה-**lowpoint** של צומת v, שתסומן $L(v)$, בתור המספר הקטן ביותר, $k(u)$, של צומת u הנגישה מ-v בעזרת מסלול מכוון, המכיל לכל היותר קשת אחורית אחת. מתקיים באופן טריוויאלי כי $L(v) \le k(v)$, כי תמיד ניתן לבחור מסלול ריק - מהקשת לעצמה.

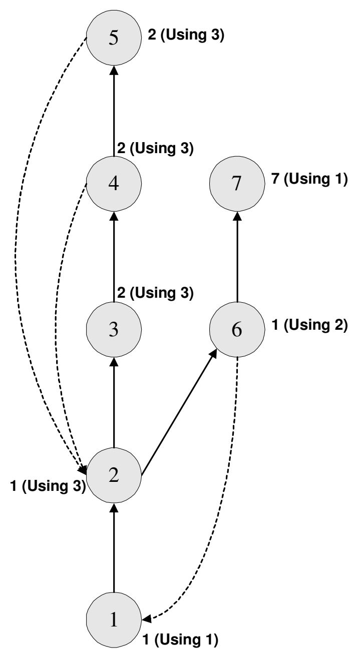
בחירת ה-lowpoint עבור צומת



ישנן שלוש אפשרויות לבחירת $L(v)$:

1. שימוש ב-$k(v)$ עצמו.
2. שימוש ב-$k(u)$ כאשר e קשת אחורנית מ-v אל u.
3. שימוש ב-$k(u)$ כאשר יש מסלול מכוון הבנוי מקשתות עץ וקשת אחורית אחת בסוף המסלול.

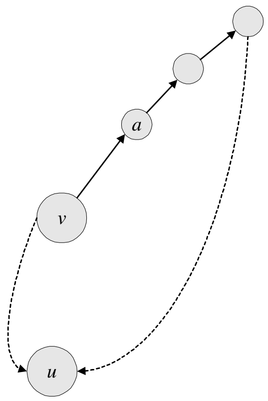
$L(v)$ יהיה האפשרות שתביא את הערך הנמוך ביותר מבין שלושת האפשרויות.



למשל, עבור העץ שראינו קודם :

בסריקת העץ (DFS) :
כאשר צומת v יתגלה בפעם הראשונה, $L(v)$ יאותחל ל-$k(v)$ (אפשרות 1).
כאשר אנו נתקלים בקשת אחורית (אנו נזהה קשת כזו על ידי העובדה ש-$k(u) \neq 0$), נשווה את $L(v)$ עם $k(u)$ ואם יתקיים כי $L(v) > k(u)$ נעדכן את $L(v)$ לערך $k(u)$.

טענה



כאשר ניסוג מצומת a אל צומת v אזי הערך $L(v)$ של צומת v יהיה הערך הסופי והנכון של $L(v)$.

למה 5

יהי G גרף שהרצנו עליו DFS (ותוך כדי כך חישבנו את $L(v)$ עבור כל הצמתים).
אם קשת עץ $v \longrightarrow u$ מקיימת כי $k(u) > 1$ (u איננה צומת ההתחלה) וגם $L(v) \geq k(u)$, אזי u היא צומת הפרדה.

הוכחה

תהא S קבוצת הצמתים הנמצאת במסלול מהשורש (כולל) עד לצומת u (לא כולל צומת זו). תהא T קבוצת הצמתים היוצאים מתחת העץ ששורשו v, כולל v.
לפי למה 4 אין קשת המקשרת בין צומת ב-T לפי צומת השייכת ל $V - \{S \cup \{u\} \cup T\}$.
בנוסף, אם קיימת קשת כלשהי המחברת בין צומת $t \in T$ אל צומת $s' \in S$, אז הקשת $t \to s'$ היא קשת אחורית ומתקיים כי $k(s') < k(u)$.
כעת $L(v) \leq k(s')$, מכיוון שניתן לקחת את קשתות העץ מ-v אל t על ידי הקשת $t \to s'$, ומכאן $L(v) < k(u)$, סתירה לטענה.
לכן: u מפרידה בין צמתי S לצמתי T, ולכן היא צומת הפרדה.

למה 6

יהי G גרף שהרצנו עליו DFS וחושב $L(v)$ לכל אחד מהצמתים. אם מתקיים עבור $u \in V$ כלשהו כי $k(u) > 1$ וגם u הוא צומת הפרדה, אזי קיים בן של u, שיסומן v, המקיים כי $L(v) \geq k(u)$.

למה 7

יהי $G(V,E)$ גרף שנסרק ב-DFS החל מצומת r $(k(r) = 1)$. r הוא צומת הפרדה אם ורק אם בעץ ה-DFS יש לו לפחות שני בנים.

הגדרה פורמלית למבנה על

יהיו $C_1, C_2, ..., C_m$ רכיבים בלתי פריקים של הגרף הקשיר $G(V,E)$. יהיו $s_1, s_2, ..., s_p$ צמתי ההפרד המתאימים. נגדיר את **מבנה העל** של $G(V,E)$ שיסומן ב- $\tilde{G}(\tilde{V},\tilde{E})$ בצורה הבאה:

$\tilde{V} = \{s_1, s_2, ..., s_p\} \cup (C_1, C_2, ..., C_m\}$,
$\tilde{E} = \{s_i - C_j \mid s_i$ is a vertex of C_j in G$\}$.

$\tilde{G}(\tilde{V},\tilde{E})$ הינו עץ.

גילוי רכיבים בלתי פריקים

נשתמש במחסנית - צומת חדש נכנס למחסנית. ברגע הנסיגה מ- v אל u (לאחר שראינו את כל הצמתים ברכיב פריק זה) על פי למה 7, u הוא השורש של עלה זה.
עלה זה הוא חלק בלתי פריק, כי אם יש לו שני בנים u הוא צומת הפרדם ואזש לא מתקיים שזהו רכיב בלתי פריק. $\Leftarrow$ ברגע שחזרנו מ-v אל u פריק, אין עוד בנים ל-u שלא ראינו, כלומר סרקנו את כל הרכיב.
כל הצמתים הנמצאים מעל v שייכים לרכיב אי פריק וקשיר ש-u שייך אליו.
אחרי ש-u הוכרז כצומת הפרדה - בעת הנסיגה פורקים מחסנית עד v, וברגע שמוצאים את u לא מוצאים אותו, ואז הצמתים שנפרקו בתוספת u הם צמתים של רכיב בלתי פריק.
על ידי פריקת מחסנית מורידים מהמבנה על עלים, וממשיכים כך עד שנגמרים העלים.
הסיבוכיות זהה לזו של DFS. הפעולות הנוספות הן בסיבוכיות קבועה.
יוצא מהכלל: כאשר מוציאים צמתים מהמחסנית, כלומר במקרה שנסוגים בקשת ומוצאים צומת הפרדה - פריקת מחסנית.
כל צומת נכנס למחסנית פעם אחת ויוצא פעם אחת $\Leftarrow$ מספר הפריקות זהה למספר הצמתים, אך סך כל מספר הפריקות חסום במספר הצמתים החסמום בעצמיו על ידי מספר הקשתות ועוד 1 (מבנה על הוא עץ).
מכאן: סיבוכיות האלגוריתם היא ליניארית.

האלגוריתם למציאת רכיבים בלתי פריקים:

S זוהי המחסנית. v מרכז הפעילות.

Procedure NONSEPARABLE(G(V,E), s; Set of separating vertices, List of nonseparable components)

1. for every $e \in E$ mark e "new"
2. for every $u \in V$ do
3. $k(u) \leftarrow 0$
4. $f(u) \leftarrow \text{NIL}$
5. $v \leftarrow s$
6. $k(s) \leftarrow 1$
7. $i \leftarrow 2$
8. vacate S
9. push s into S
10. while v has a new incident edge or $f(v) \neq \text{NIL}$
11. if v has new incident edge $v \overset{e}{\text{——}} u$ then do
12. mark e "old"
13. if $k(u) = 0$ (u is a new vertex) then do
14. push u into S
15. $f(u) \leftarrow v$
16. $k(u) \leftarrow i$
17. $L(u) \leftarrow i$
18. $i \leftarrow i+1$
19. $v \leftarrow u$
20. else (u is old) do
21. $L(v) \leftarrow \min\{L(v), k(u)\}$
22. else ($f(v)$ is defined)
23. if $L(v) \geq k(f(v))$ then do
24. if $f(v) \neq s$ or s has a new incident edge then do
25. add $f(v)$ to the set of separating vertices
26. pop vertices from S down and including v
27. the set of popped vertices, with $f(v)$, is an element of of the set of nonseparable components
28. else ($L(v) < k(f(v))$) then do
29. $L(f(v)) \leftarrow \min\{l(f(v)), L(v)\}$
30. $v \leftarrow f(v)$

ההבדל מאלגוריתם DFS הינו השורות: 8, 9, 14, 17, 20, 21, 23-27, 28, 29.

אלגוריתם DFS לגרפים מכוונים

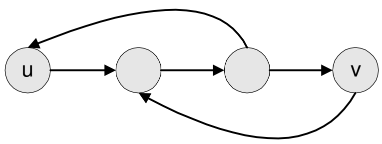
קלט: גרף מכוון וסופי G(V, E).

מטרה: חישוב $k(v), f(v)$ לכל צומת, בדומה ל-DFS עבור גרף לא מכוון.

<u>האלגוריתם</u>

1) לכל $e \in E$ סמן את e "חדשה".
2) לכל $u \in V$ בצע:
3) $k(u) \leftarrow 0$
4) $f(u) \leftarrow \text{NIL}$
5) $v \leftarrow s$
6) $k(v) \leftarrow 1$
7) $i \leftarrow 2$
8) כל עוד קיים צומת חדש u $\left(k(u)=0\right)$ בצע:
9) אם ל-v יש קשת חדשה $v \xrightarrow{e} w$ אזי בצע:
10) סמן את e כ-"ישנה".
11) אם $k(w)=0$ (צומת חדש) בצע:
12) $f(w) \leftarrow v$
13) $k(w) \leftarrow i$
14) $i \leftarrow i+1$
15) $v \leftarrow w$
16) אחרת: $v \leftarrow f(v)$.

רכיבים קשירים היטב של גרף מכוון



נגדיר יחס בין צמתים: uNv אם קיים מסלול מכוון מ-u אל v וגם מסלול מכוון מ-v אל u.
היחס משרה חלוקה על הגרף. כל קבוצה הנוצרת הינה **רכיב קשיר היטב**.

<u>טענה</u>

היחס N הוא יחס שקילות.

הוכחה:
- vNv (קיים מסלול ריק)
- אם $a \sim b$ אזי bNa.
- אם $a \sim b$ וגם bNc אזי aNc. (מסלול מ-a ל-c יעבור דרך b ולהפך על ידי סימטריות).

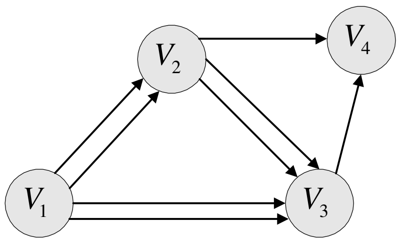
<u>מסקנה</u>

מתקיים: $V = V_1 \cup V_2 \cup ... \cup V_m$, כאשר V_i הינו רכיב קשיר היטב בגרף.

<u>הגדרה</u>

רכיב קשיר היטב שאין ממנו קשתות יוצאות יקרא **בור**.

דוגמא:
נביט במבנה העל הבא:



בתוך כל V_i יש קשתות על פי הגדרת היחס.
V_4 הוא בור, לפי ההגדרה.

<u>אלגוריתם Kosaraju-Sharir למציאת רכיבים קשירים היטב בגרף מכוון</u>

יהי G(V,E) גרף, ונניח כי מבנה העל שלו הוא גרף מכוון אציקלי, אשר לו רכבי מקור אחד ורכיב בור אחד.

<u>האלגוריתם</u>

1. בצע וריאציה של DFS על G(V, E) לחישוב הסימנים $h(v)$. הסימון הוא מספור של הצמתים לפי סדר הנסיגה מהם. הוריאציה תסומן DFSB.
2. בנה את הגרף G^R מתוך G על ידי הפיכת כל כיווני הקשתות. (נשים לב שרכיבים קשירים היטב לא נפגעו).
3. בצע וריאציה נוספת של DFS על הגרף - כל פעם נבחר צומת "חדש" עם סימון h הגדול ביותר. הצמתים המתגלים בסיור DFS ממנו מהווים רכיב קשיר היטב. נסמן את כולם "ישנים" ונחזור על התהליך עד שנמצא את כל הרכיבים. הוריאציה תסומן DFSA.

<u>סיבוכיות האלגוריתם</u>

כל אחד מהצעדים מתבצע ב-$O(|E|)$, ולכן סיבוכיות האלגוריתם היא $O(|E|)$.

<u>טענה</u>

אם ב-DFS מתגלה צומת a וממנו יש מסלול מכוון לצומת $b(\neq a)$ שכל הצמתים בו, כולל b, עדיין לא התגלו, אזי בעת הנסיגה מ-a, צומת b יהיה גלוי וגם הנסיגה ממנו תחול קודם לנסיגה מצומת a.

<u>הגדרה</u>

יהי גרף $G(V,E)$, ויהי T העץ המתקבל מהפעלת DFS על הגרף.
נסמן ב- T_a את תת העץ של T, המכיל את a ואת כל צאצאיו.

<u>מסקנה</u>

ביער המכוון המתקבל לאחר שלב 1 של האלגוריתם, $\forall v$ מתקיים כל צומת v מתגלה לפני כל צמתי T_v (אם יש כאלו), ולכן מתקיים: $h(v) = \max_{u \in T_v} \{h(u)\}$

<u>טענה</u>

יהי C רכיב קשיר היטב של G, ויהי v הצומת ברכיב זה שעבורו $h(v)$ הגבוה ביותר.
בתנאים אלה, v הוא הצומת הראשון שהתגלה ב-C בריצת DFSB.

<u>טענה</u>

יהי v הצומת שעבורו בריצת DFSB מתקבל $h(v)$ הגבוה ביותר. לרכיב הקשיר היטב C_1 שבו v נמצא, אין קשתות נכנסות מצמתים אחרים. כלומר, C_1 הוא רכיב מקור.

הכללה: תהי $A = \{C_1, C_2, ..., C_k\}$ תת קבוצה של רכיבים קשירים היטב בגרף G.
הצומת v שעבורו $h(v)$ הגבוה ביותר מבין הצמתים $\bigcup_{i=1}^{k} C_i$ נמצא ברכיב הקשיר היטב C_j של G שאליו אין קשתות מרכיבים אחרים מ-A.

<u>טענה</u>

הצומת v שעבורו $h(v)$ הגבוה ביותר נמצא ברכיב קשיר היטב של G^R שממנו אין קשתות יוצאות אל רכיבים אחרים. כלומר ב- G^R רכיב זה הוא רכיב בור.

<u>טענה</u>

סיור ה-DFSA מצומת v שעבורו $h(v)$ הגבוה ביותר יגלה בדיוק את אברי הרכיב הקשיר שבו נמצא v.