2.6/11

21

נושא 5 - DFS (Depth First Search)

א) **הגדרה:** DFS זהו אלגוריתם המקבל גרף G לא ממושקל וצומת כלשהי בגרף זה. האלגוריתם מתחיל בצומת זו וממנו עובר על כל צמתי הגרף, שיטת המעבר היא סריקה לעומק, כלומר מתקדם לאורך הגרף עד שהוא נתקע, ואז חוזר אחורה ובוחר צומת שטרם היה בה.* במהלך הסריקה פועל "שעון", שבכל פעם שאנו מגלים צומת חדשה או נתקעים, השעון קופץ ב-1. בתחילת הסריקה כל הצמתים הם לבנים, כאשר אנו מגלים צומת לראשונה נצבע צומת זו באפור, וכאשר נתקענו בצומת, כלומר ממנה אין לאן להתקדם, נצבע אותה בשחור.

* אם הגרף אינו קשיר, נבחר צומת כלשהי ברכיב הקשירות הנוסף ונמשיך ממנה עד שנעבור על כל צמתי הגרף.

מטרת האלגוריתם להחזיר את שלושת המערכים הבאים:

d[v] - זמן הגילוי של כל צומת (מעבר מלבן לאפור). כל צומת מיוצגת בתא נפרד במערך.

f[v] - זמן הסיום של כל צומת (מעבר מאפור לשחור).

π[v] - עבור כל צומת u נשמור מיהי הצומת שבאמצעותה גילינו את u. בתחילה כולם NULL.

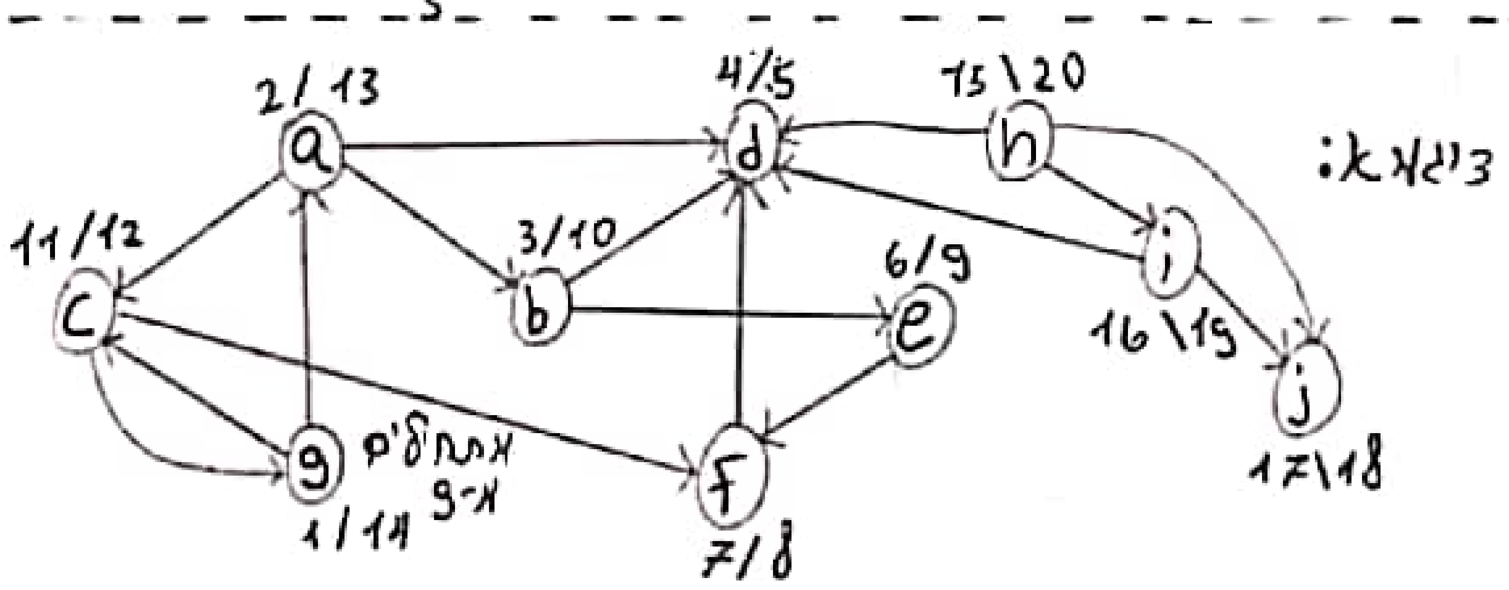
באמצעות שלושת מערכים אלו נוכל למצוא מידע שימושי מאוד על הגרף שעליו הפעלנו את DFS.

ב) **פסאודו קוד:**

```
DFS (G, u) {
    time = 0
    For (Node u : V) {
        color[u] = WHITE
        π[v] = NULL
    }
    For (Node u : V) {
        IF (color[u] = WHITE)
            DFS_Visit (u)
    }
}
```

```
DFS_Visit (u) {
    color[u] = GRAY
    time ++
    d[u] = time
    For (Node w : adj[u]) {      // adj[u] הם כל השכנים של u
        IF (color[w] = WHITE)
            π[w] = u
            DFS_Visit (w)
    }
    color [u] = BLACK
    time ++
    F[u] = time
}
```

דוגמא:

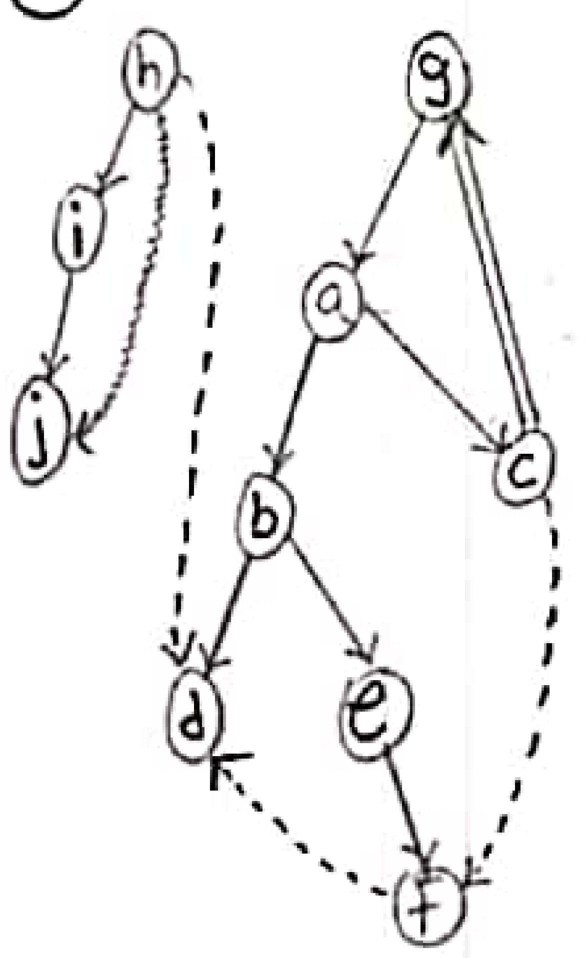


d = {2, 3, 11, 4, 6, 7, 1, 15, 16, 17}

f = {13, 10, 12, 5, 9, 8, 14, 20, 19, 18}

π = {g, a, a, b, b, e, NULL, NULL, h, i}

ג) **זמן ריצה:** באלגוריתם זה אנו עוברים על כל צמתי הגרף וכל הקשתות בו, ולכן זמן הריצה הוא $\Theta(V+E)$.



ג) **יער עומק**: זהו יער המתאר את המסלול שבו הלכנו באלגוריתם DFS. אם הגרף הוא קשיר אזי יער העומק יתואר כעץ 1 ששורשו בצומת שהתחלנו. ואם הגרף אינו קשיר יתואר כיער. יער העומק מכיל את כל קשתות הגרף עם סימונים מיוחדים לסיווג הקשתות כמו שמוצג בהמשך. בציור משמאל ניתן לראות את היער עומק של הדוגמא מעמוד קודם.

סיווג קשתות: ניתן לסווג כל קשת בגרף G ל-4 סוגי קשתות לפי זמן הגילוי של צומת היעד וצומת המקור בקשת:

(1) קשת עץ - זוהי קשת מצומת שהתגלתה לצומת שלא התגלתה. מאפור ללבן. ביער עומק ניתן לראות זאת כקשת מאבא לבן. מסומנת בקו רגיל ←.

(2) קשת אחורה - זוהי קשת מצומת שהתגלתה לצומת שהתגלתה אך עדיין לא הגענו לזמן הסיום שלה. מאפור לאפור. ביער עומק ניתן לראות זאת כקשת מצאצא לאב קדמון. מסומנת ⇐.

(3) קשת קדימה - זוהי קשת מצומת שהתגלתה אל צומת שהגיע לזמן הסיום שלה. מאפור לשחור. אלא שצומת המקור התגלתה לפני צומת היעד. ביער עומק ניתן לראות זאת כקשת מאב קדמון אל צאצא שלו. מסומנת בקו מזוגזג ⟿.

(4) קשת חוצה - זוהי קשת מצומת שהתגלתה אל צומת שהגיע לזמן הסיום שלה. מאפור לשחור. אלא שצומת המקור התגלתה אחרי צומת היעד. ביער עומק ניתן לראות זאת כקשת בין שני צמתים לא קשורים.* מסומנת בקו מקווקו ←----.

* גם באותו תת-עץ

עבור קשת (u,v) מאפור לשחור נוכל להבדיל בין קשת קדימה לקשת חוצה כך:
אם $d[u] < d[v]$ אז זוהי קשת קדימה. ואם $d[u] > f[v] > d[v]$ זוהי קשת חוצה.

ה) **משפטי עזר**:

(1) **משפט הסוגריים**: דרך נוספת לייצג את סריקת DFS היא באמצעות סוגריים. בכל פעם שנגלה צומת נפתח סוגריים ובכל פעם שנסיים צומת נסגור את הסוגריים. לדוגמא, עבור הסריקה לדוגמא מעמוד קודם נקבל:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(	(	(	(	)	(	(	)	)	)	(	)	)	)	(	(	(	)	)	)
g	a	b	d	d	e	f	f	e	b	c	c	a	g	h	i	j	j	i	h

נשים לב כי הביטוי שהתקבל מייצג סוגריים מאוזנות, וכי אם הסוגריים של שני צמתים זרים ()() אזי ביער עומק יהיו בשני תתי עצים נפרדים, ואם מקוננים (()) אזי יהיו באותו תת עץ. מכאן נגיע לשני מסקנות:

① v צאצא של u אם"ם הוא בתוך הסוגריים שלו, כלומר שמתקיים: $d[u]<d[v]<f[v]<f[u]$.
אחרת הם מוכרחים להיות בסוגריים מקוננים. לא יכול להיות $d[u]<d[v]<f[u]<f[v]$. ✓

שתי מסקנות אלו לא נוכיח אך נשתמש.

② משפט המסלול הלבן - v צאצא של u אם ורק אם בזמן הגילוי של u ($d[u]$) יש מסלול מ-u ל-v שכולו צמתים לבנים, כלומר שנוכל לפתוח סוגריים עד שנגיע ל-v בלי שנצטרך לסגור את u.

(2) <u>סיווג קשתות בגרף לא מכוון:</u> בהרצת DFS על גרף לא מכוון נקבל רק קשתות מסוג קשת עץ או קשת אחורה, לא נקבל קשתות מסוג קשת קדימה או קשת חוצה.
הוכחה: תהי קשת $(u,v)\in E$. בלי הגבלת הכלליות נניח כי u התגלה לפני v, $d[u]<d[v]$. אזי מכיוון ש-u ו-v באותו תת עץ, לפי משפט הסוגריים v ח"ב להתגלות ולהסתיים לפני ש-u מסתיים, $d[u]<d[v]<f[v]<f[u]$. במהלך סריקת DFS על הגרף נעבור בקשת (u,v), יכולים להיות רק שתי אפשרויות: אם נעבור בכיוון $u\to v$, כלומר v היה לבן עד שעברנו בקשת זו, אזי (u,v) היא קשת עץ. ואם נעבור בכיוון $v\to u$ אזי נהיה מעבר מאפור לאפור, ולכן (u,v) היא קשת אחורה. אין עוד אפשרויות, ולכן אלו שתי סוגי הקשתות היחידים שנקבל בגרף לא מכוון.

(3) <u>זיהוי גרף חסר מעגלים:</u> גרף מכוון G יהיה חסר מעגלים אם ורק אם בסריקת DFS אין שום קשת אחורה. גרף חסר מעגלים מכונה DAG (Directed Acyclic Graph).
<u>הוכחה: כיוון 1</u> - נניח שב-G אין מעגל, וצ"ל שאין קשת אחורה. נניח בשלילה שיש קשת אחורה (u,v). לפי הגדרת קשת אחורה אזי בהכרח u הוא צאצא של v, כלומר יש מסלול כלשהו מ-v ל-u, וכן יש קשת מ-u ל-v. קיבלנו מעגל $v\to u\rightsquigarrow v$. סתירה!
<u>כיוון 2</u> - נניח כי אין קשת אחורה, וצ"ל שאין מעגל. נניח בשלילה שב-G יש מעגל c המתחיל ומסתיים בצומת v, וכי הקשת האחרונה במעגל היא (u,v). בתחילת המעגל, בזמן הגילוי של v, ישנו מסלול של צמתים לבנים מ-v ל-u. שהרי לפי האלגוריתם אם צומת לא לבן לא ניכנס אליו ונחזור אחורה. לכן אם הגענו ל-u ישנו מסלול של צמתים לבנים מ-v ל-u. אזי לפי משפט המסלול הלבן u הוא צאצא של v, ולכן הקשת (u,v) היא קשת מצאצא לאב קדמון, שלפי הגדרתנו היא קשת אחורה. סתירה!

מיון טופולוגי

כאן עד פה 1) צריך אלו בעיות ניתן לפתור באמצעות מידע שקיבלנו ב-DFS.

לעיתים נרצה להציג קבוצת איברים שמוגדר עליהם סדר חלקי אך סדר מלא. סדר חלקי מקיים טרנזיטיביות (לדוגמא אם $a>b$ ו-$b>c$ אזי $a>c$), אך לא ניתן להשוות את היחס על כל שני איברים (לדוגמא אם $a>b$ אזי $b\ngtr a$). לעומת זאת סדר מלא אומר שכל

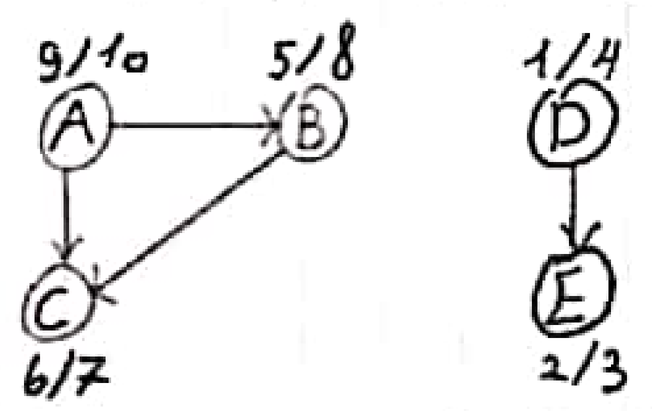
3/12 בס"ד

(24)

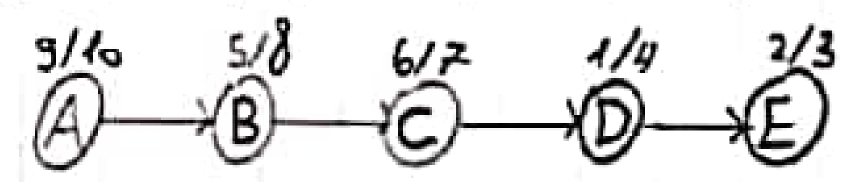
כל שני איברים ניתן להשוות את היחס. ניתן לייצג סדר בגרף. סדר חלקי יוצג בגרף לא קשיר, ואילו סדר מלא יוצג בגרף קשיר. באמור לעיתים נרצה להמיר קבוצת איברים המוגדר עליהם סדר חלקי אל סדר מלא. לדוגמא, עבור סדר פעולות מסוים שבו פעולה גוררת פעולה, אך יש שני תהליכים מקבילים שלא תלויים זה בזה ומייצגים סדר חלקי, נרצה להמיר לסדר מלא כדי שיהיה לנו סדר פעולות אחד שמבצע את שניהם. נוכל לעשות זאת באמצעות מיון טופולוגי.

אלגוריתם מיון טופולוגי - אלגוריתם זה פועל על גרף DAG, כלומר חסר מעגלים. מטרת האלגוריתם היא להחזיר רשימה מקושרת של כל צמתי הגרף, כך שעבור כל קשת $(u,v)\in E$ צומת u תופיע לפני v ברשימה המקושרת.

האלגוריתם יקרא לאלגוריתם DFS כדי לחשב את זמני הסיום של כל צומת $F[v]$. בכל פעם שנסיים עם צומת נכניס אותה לראש הרשימה המקושרת. לבסוף נחזיר את הרשימה המקושרת. 3 פעולות אלו יבצעו את מטרת האלגוריתם כמו שנוכיח בהמשך.



לדוגמא: עבור הגרף משמאל שהרצנו עליו DFS מצומת D נקבל את הרשימה המקושרת:



הוכחת האלגוריתם - צריך להוכיח שעבור כל קשת בגרף (u,v) צומת u תופיע לפני v ברשימה המקושרת, זה יקרה כאשר $F[v]<F[u]$.

תהי קשת כלשהי בגרף $(u,v)\in E$. בזמן הסריקה של קשת זו צומת u בהכרח אפור אך מה יהיה הצבע של v? נבדוק את כל שלושת האפשרויות:

v אפור - לא יכול להיות כי אז (v,u) הייתה קשת אחורה בסתירה לכך שהגרף הוא DAG, לפי טענת עזר (3).

v לבן - אזי v הוא צאצא של u, ולפי משפט הסוגריים מתקיים: $d[u]<d[v]<F[v]<F[u]$

v שחור - אזי v כבר יסתיים אך u לא יסתיים כי עדיין חוקרים אותו, לכן $F[v]<F[u]$.

רכיבי קשירות חזקה ט

הגדרה: גרף G מוגדר כקשיר חזק אם עבור כל זוג צמתים u,v יש מסלול מ-u ל-v וכן יש מסלול מ-v ל-u. אחרת נאמר שהגרף לא קשיר חזק (אך יכול להיות קשירות רגילה).

גרף שאינו קשיר חזק ניתן לחלק לרכיבי הקשירות החזקים. - SCC*. כל רכיבי קשירות חזק הוא קבוצה Strongly connected component

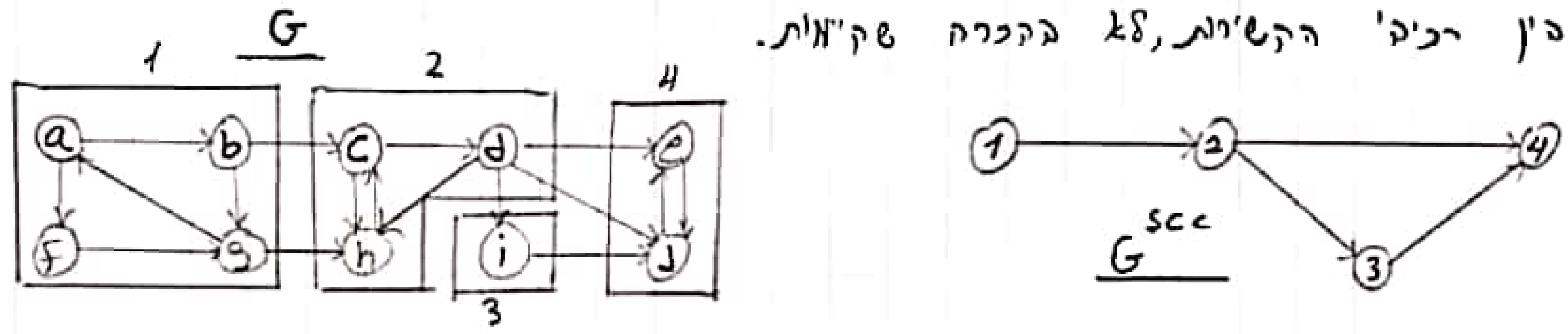
מקסימלית של קודקודים C, כך ש- $C\subseteq V$, ולכל שני צמתים ב-C u,v יש מסלול מ-u ל-v ומסלול מ-v ל-u

17/12

(29)

בס"ד

גרף רכיבים - G^{scc}: גרף רכיבים המוגדר $G^{scc}=(V^{scc},E^{scc})$ הוא גרף שבו כל צומת מייצגת רכיב קשירות חזקה בגרף G המקורי. כל קשת בגרף G^{scc} הם רק קשתות שיש בין רכיבי הקשירות, לא בהכרח שקיימות.



גרף G^{scc} הוא בהכרח חסר מעגלים (DAG). ננסח את המשפט בצורה רשמית:

משפט: יהיו C ו-C' שני רכיבי קשירות חזקים בגרף G. יהיו $u,v\in C$ ו-$u',v'\in C'$.
אזי אם יש מסלול $u\rightsquigarrow u'$ לא יכול להיות מסלול $v'\rightsquigarrow v$.

הוכחה: נניח בשלילה כי יש מסלול $v'\rightsquigarrow v$. מכיוון ש-u ו-v באותו רכיב קשירות חזקה בהכרח יש מסלול $v\rightsquigarrow u$, ובהתאם יש מסלול $u'\rightsquigarrow v'$. לכן נקבל את המסלול הבא $u\rightsquigarrow u'\rightsquigarrow v'\rightsquigarrow v\rightsquigarrow u$, כלומר יש מסלול מ-$u$ ל-v' וכן יש מסלול מ-v' ל-u, לכן הם צריכים להיות באותו רכיב קשירות. סתירה!

שחלוף גרף מכוון - G^T: השחלוף של גרף מכוון G הוא גרף G^T שבו כל הקשתות מחליפות כיוון. כלומר אם גרף $G=(V,E)$ אזי גרף $G^T=(V,E^T)$ כאשר $E^T=\{(u,v):(v,u)\in E\}$.
ניתן לייצר גרף G^T בזמן ריצה $\Theta(V+E)$ אם משתמשים ברשימת סמיכויות ולא במטריצת סמיכויות. ל-G ו-G^T יש אותם רכיבי קשירות חזקה!

אלגוריתם למציאת רכיבי קשירות חזקה - כדי לחלק את קבוצת הצמתים V לפי רכיבי הקשירות החזקה שבגרף נבצע את האלגוריתם הבא:

זמן ריצת כל שלב:

(1) נפעיל אלגוריתם DFS על גרף G כדי לחשב את זמן הסיום של כל צומת $f[u]$. — $\Theta(V+E)$

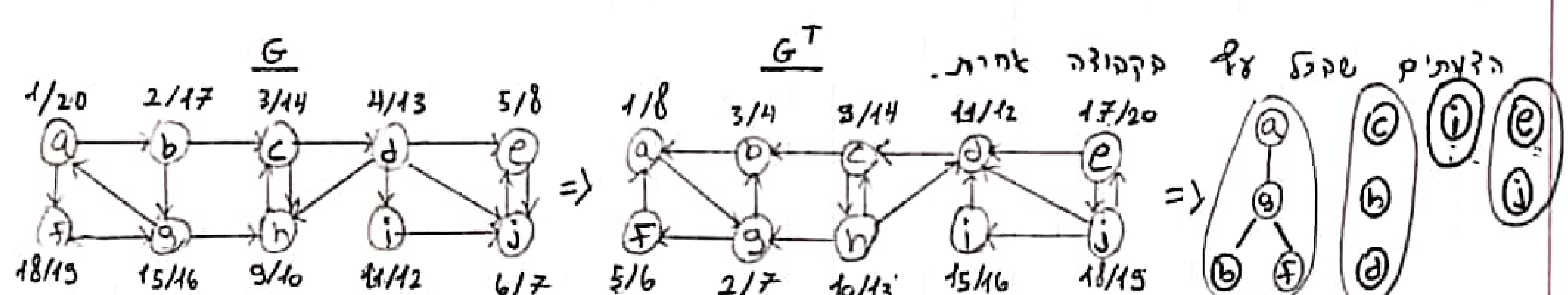
(2) נבנה את G^T שהוא השחלוף של גרף G. — $\Theta(V+E)$

(3) נפעיל שוב אלגוריתם DFS אך הפעם על G^T. בנוסף סדר בחירת הצמתים יהיה לפי סדר יורד של $f[u]$ מה-DFS הראשון. כלומר נבחר קודם לצמתים שזמן הסיום שלהם הכי גבוה. — $\Theta(V+E)$

(4) ביער עומק שקיבלנו ב-DFS השני על G^T, נתייחס רק לקשתות העץ ולא לסוגי קשתות אחרות. כל עץ ביער עומק מייצג רכיב קשירות חזקה אחר. נחזיר את הצמתים שבכל עץ בקבוצה אחרת. — $\Theta(V)$

סה"כ: $\Theta(V+E)$

דוגמא:



<u>טענות עזר</u> - נוכיח שתי טענות עזר לצורך הוכחת האלגוריתם:

(1) טענה: יהיו c,c' שני רכיבי קשירות חזקה שונים ב-G. נניח שישנה קשת $(u,v)\in E$ כך ש-$u\in C$ ו-$v\in C'$, אזי בסריקה <u>על G</u> זמן הסיום של C כולו <u>גדול</u> מזמן הסיום של c'. $f(c)>f(c')$.

הוכחה: נבחן את זמן הגילוי של c ו-c', יכולים להיות שני מקרים. אם $d(c)<d(c')$, כלומר גילינו את c לפני c', אזי בזמן הגילוי של הצומת הראשונה ב-c, נקרא לה x, כל שאר הצמתים ב-c וב-c' היו לבנים. מכיוון שיש מסלול מ-x לכל הצמתים ב-c וב-c', שהרי שניהם רכיבי קשירות חזקה ויש קשת מ-c ל-c', לפי משפט המסלול הלבן כל הצמתים ב-c וב-c' הם צאצאים של x, לכן זמן הסיום של x גדול משל כולם. ומכאן $f(c)>f(c')$ שהרי $x\in c$.

האפשרות השנייה ש- $d(c)>d(c')$, כלומר גילו את c' לפני c. אזי בזמן הגילוי של הצומת הראשונה ב-c', נסמנה y, כל הצמתים ב-c ו-c' היו לבנים. מכיוון שיש מסלול מ-y לכל הצמתים ב-c', לפי משפט המסלול הלבן זמן הסיום של y גדול מכל הצמתים ב-c'. אמנם בזמן זה כל הצמתים ב-c עדיין לבנים, כי אין דרך להגיע מ-c' ל-c, שהרי אם כן הם היו באותו רכיב קשירות, כי יש קשת מ-c ל-c', ולכן זמן הסיום של כל צומת ב-c גדול מ-c', והטענה מתקיימת.

(2) טענה: יהיו c,c' שני רכיבי קשירות חזקה שונים ב-G. נניח שישנה קשת $(u,v)\in E^T$ כך ש-$u\in C$ ו-$v\in C'$. אזי בסריקה על G זמן הסיום של c כולו <u>קטן</u> מזמן הסיום של c'. $f(c)<f(c')$.

הוכחה: מכיוון ש-$(u,v)\in E^T$ אזי $(v,u)\in E$. ואז לפי טענה קודמת נקבל $f(c')>f(c)$.

המסקנה מטענה זו היא שאם $f(c)>f(c')$ אזי אין ב-G^T קשת מ-c ל-c'.

<u>הוכחת נכונות האלגוריתם</u> - בהפעלת ה-DFS השני על G^T נתחיל מהצומת שזמן הסיום שלה ב-DFS הראשון מקסימלי, נסמן צומת זו ב-x. נניח x שייכת לרכיב קשירות חזקה C. ב-DFS השני מכיוון ש-$f(c)$ מקסימלי אזי לפי מסקנת הטענת עזר השנייה אין קשת מ-c לרכיב אחר, לכן מ-x נמשיך לכל הצמתים ב-c, והם יהיו עץ נפרד ביער עומק של ה-DFS השני. לאחר שנסיים את c נעבור לצומת הבאה שזמן הסיום שלה מקסימלי מבין הצמתים שנותרו, השייכת לרכיב קשירות חזקה c'. לפי אותה מסקנה מטענת עזר השנייה יכולה להיות קשת רק מ-c' ל-c אך מכיוון שכבר סיימנו עם כל הצמתים ב-c, נסרוק רק את הצמתים ב-c', כך שביער עומק גם c' יוצג בתור עץ נפרד. כך נמשיך בכל צמתי G^T, כך שכל רכיב קשירות חזקה יוצג עץ נפרד ביער עומק. מ.ש.ל.