מבני נתונים ואלגוריתמים – תרגול 8-9#

מציאת מסלולים קצרים

<u>הבעיה</u>: נתון גרף. ממשוקל. רוצים למצוא את המסלול הקצר בין זוג קודקודים.

עיקרון ***הרלקסציה*** של קשת: בדיקה האם ניתן לשפר מסלול מ-*s* ל-*v* ע"י מעבר דרך קודקוד *u*:

$$d[v] > d[u] + w(u,v)$$

?

<u>מציאת מסלול מקודקוד **מקור יחיד** לכל שאר הקודקודים:</u>

<u>האלגוריתם של ***Dijkstra*** (משקלים חיוביים):</u>

קלט: גרף נתון גרף מכוון/לא מכוון $G = (V,E)$, פונקציית משקל על הקשתות $w: E \to \mathbb{R}^+$ (כלומר $\forall\ e \in E, \boldsymbol{w(e)} > 0$) וקודקוד מקור *s*.

פלט: לכל $v \in V$ - אורך המסלול (=משקל המסלול) הקצר ביותר מ-*s* ל-*v*.

הרעיון: תחילה מגדירים את המרחק מ-s לכל קודקוד להיות משקל הקשת בינו לבין s אם קיים, אחרת אינסוף, במערך D. אחר כך בלולאה מחפשים את הקודקוד בגרף שהמרחק שלו מ-s הוא המינימלי. בודקים עבורו האם קיים מסלול קל יותר מ-s אליו שעובר דרך קודקוד שאינו ב-S (קודקודים שכבר מצאנו עבורם את המסלול הקצר ביותר).

<u>*Dijkstra (G, W, s):*</u>

S ={s}

D(s) = 0

for each $v \in V/S$

$$D(v) = \begin{cases} w(s,v), & (s,v) \in E \\ \infty, & (s,v) \notin E \end{cases}$$

while $S \neq V$

find $v \in V/S$ *so that D(v) = min(D)*

$S = S \cup \{v\}$

for each $u \in V/S$

D(u) = min(D(u), D(v) + w(v,u)) *(relaxation)*

סיבוכיות:

- $O(|V|^2)$.
- ניתן לשפר עם ערימת פיבונצ'י – $O(|V|log|V| + E)$.

<u>האלגוריתם של ***Bellman-Ford*** (גם משקלים שליליים):</u>

קלט: גרף נתון גרף **מכוון** $G = (V,E)$, פונקציית משקל על הקשתות $w: E \to \mathbb{R}$ (כלומר ייתכנו משקלים שליליים), וקודקוד מקור *s*.

פלט: לכל $v \in V$ – אורך המסלול (=משקל המסלול) הקצר ביותר מ-*s* ל-*v*.

הרעיון: לעומת דייקסטרה, האלגוריתם הזה לא חמדני. עוברים על כל הקשתות |V|-1 פעמים, ועושים "הקלה" עבור כל קשת אם אפשר. אם אחרי כל האיטרציות, עדיין ניתן לעשות רלקסציה לקת מסוימת, סימן שיש מעגל שלילי, והאלגוריתם לא יחזיר מרחקים קצרים אלא false.

<u>Bellman-Ford(G, W, s):</u>

```
for each v ∈ V
        d(v) = ∞
        Π(v) = NULL
d(s) = 0
for i=1 to |V|-1
        for each e = (u,v) ∈ E
                if d(v) > d(u) + w(u,v)
                        d(v) = d(u) + w(u,v)
                        Π(v) = u
for each e = (u,v) ∈ E // search for negative cycle
        if d(v) > d(u) + w(u,v)
                return false
return true
```

סיבוכיות: $O(|V|*|E|)$

<u>מציאת מסלול **מכל קודקוד** לכל קודקוד:</u>

<u>האלגוריתם של ***Floyd-Warshall***:</u>

קלט: גרף נתון גרף מכוון/לא מכוון $G = (V, E)$, פונקציית משקל על הקשתות $w: E \to \mathbb{R}^+$ (כלומר $\forall\, e \in E, \boldsymbol{w(e)} > 0$).
פלט: אורך המסלול הקצר ביותר מכל קודקוד לכל קודקוד.

הרעיון: האלגוריתם מתבסס על האבחנה הבאה – אם ממספרים את הקודקודים 1,2,..., n ומסמנים ב-$\mathrm{d_k(x,y)}$ את משקל המסלול הקצר ביותר מ-x ל-y שעובר אך ורק בקודקודים מתוך הקבוצה {1,2,..., k}, אז מתקיים : $\mathrm{d_k(x,y) = \min\{d_{k-1}(x,y), d_{k-1}(x,k) + d_{k-1}(k,y)\}}$. כלומר תמיד מתקיים אחד משניים:

- *אם המסלול הקצר ביותר מ-x ל-y שעובר דרך קודקודים שמספרם לכל היותר k ולא עובר ב-k עצמו, אז $\mathrm{d_k(x,y) = d_{k-1}(x,k)}$.*
- *אחרת, אם המסלול כן עובר ב-k עצמו, הוא עובר בו רק פעם אחת (כי אין מעגלים שליליים בגרף) ואז אפשר לפרק את המסלול ל-2 : מסלול מ-x ל-k ומסלול מ-k ל-y. מסלולים אלו לא עוברים דרך קודקודים שמספרם גדול מ-k-1, לכן משקל המסלול הכולל יהיה $\mathrm{d_{k-1}(x,k) + d_{k-1}(k,y)}$.*

כעת נשאר לבדוק איזה מסלול קצר יותר עבור על קודקוד.

<u>Floyd-Warshall (G, W):</u>

for k=1 to n
        for i=1 to n
                for j=1 to n

$$d_{ij}^k = min\{\underbrace{d_{ij}^{k-1}}_{\text{המסלול לא עובר ב-}k}, \underbrace{d_{ik}^{k-1} + d_{kj}^{k-1}}_{\text{המסלול עובר ב-}k}\}$$

d_{ij}^k - נמרחק המינימלי בין *i* ל-*j* כאשר המסלול בין הקודקודים הממוספרים לכל היותר *k*.
סיבוכיות: $O(|V|^3)$

תרגיל:

נתונים n סוגי מטבעות. שערי החליפין בין כל זוג מטבעות נתונים במטריצה A_{nxn} כך ש-A_{ij} הוא שער החליפין בין מטבע i ל-j (כלומר עבור מטבע אחד מסוג i ניתן לקנות A_{ij} מטבעות מסוג j).
הציעו אלגוריתם המכריע האם קיימת סדרת החלפת מטבעות, המתחילה ומסתיימת באותו מטבע כך שבסיומה נרוויח כסף.

פיתרון:

נמיר את הבעיה למציאת מעגל שלילי בגרף.
הגדרת הגרף:
- $G = (V, E)$ על n צמתים.
- בין כל זוג צמתים תהיה קשת $w_{ij} = -\ln(A_{ij})$
  רוצים לקבוע משקלים כך שמעגל שלילי בגרף יהיה שקול לסדרת החלפת מטבעות.

מעגל שלילי – אפשר להגיע מצומת לעצמו במסלול שמחירו קטן מ-0.

האלגוריתם:
- בנה גרף G כנ"ל.
- הרץ בלמן-פורד מקודקוד מקור שרירותי
- אם קיים מעגל שלילי – החזר אותו.

- מכיוון שהגרף קשיר, חיפוש מעגל שלילי לא תלוי בקודקוד המקור.

סיבוכיות: $|V| = n,\ |E| = n^2 \leftarrow O(n^3)$.

הוכחת נכונות:

אם $v_1, v_2, \ldots, v_k, v_1$ הוא מעגל שלילי, אז מתקיים:

$$-\ln(A_{12}) - \ln(A_{23}) - \cdots - \ln(A_{k1}) < 0$$

$$-\ln(A_{12} \cdot A_{23} \cdot \ldots \cdot A_{k1}) < 0$$

$$A_{12} \cdot A_{23} \cdot \ldots \cdot A_{k1} > 1$$

כלומר ממטבע יחיד מסוג 1 נקבל יותר ממטבע מסוג 1 בסוף הסדרה, לכן אפשר להרוויח כסף ע"י המרת v_1 בסדר הבא:

$$v_1 \rightarrow \ldots \rightarrow v_k \rightarrow v_1$$

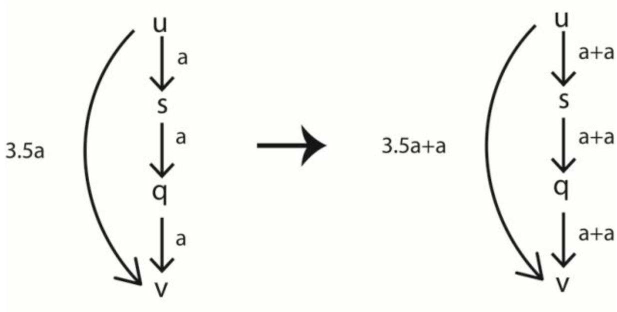
תרגיל:

גרף נתון גרף קשיר ולא מכוון $G = (V, E)$ ופונקציית משקל על הקשתות $w: E \rightarrow \mathbb{R}$.

הוכיחו/הפריכו את הטענות הבאות:
1) נגדיר פונקציית משקל חדשה $c: E \rightarrow \mathbb{R}$ כך ש- $c(e) = w(e) + a$ (a אי-שלילי). אם P הוא המסלול הקצר ביותר בין זוג קודקודים לפי w, הוא גם הקצר ביותר לפי c.
2) כנ"ל עם פונקציית המשקל $c(e) = w(e) * a$.

פיתרון:

1) לא נכון. דוגמא נגדית:



מוסיפים a לכל הקשתות, לכן זה משנה כאשר מספר הקשתות של המסלולים השונים האפשריים משתנה.

2) נכון. הוכחה (בשלילה):

נניח שקיים מסלול Q כך ש- $w(P) = w(Q)$, אך $c(P) > c(Q)$ (כלומר Q קצר לפי w אך לא לפי c).
מקבלים:

$$c(Q) = \sum_{e\in Q} a\cdot w(e) = \sum_{e\in Q} c(e) < \sum_{e\in P} c(e) = \sum_{e\in P} a\cdot w(e) = c(P)$$

$$a\cdot w(Q) = a\sum_{e\in Q} w(e) < a\sum_{e\in P} w(e) = a\cdot w(P)$$

$$w(Q) < w(P)$$

בסתירה להנחה ש-$w(P) = w(Q)$.

<u>תרגיל:</u>

גרף נתון גרף קשיר ומכוון $G = (V,E)$ ופונקציית משקל על הקשתות $w: E \to \mathbb{N}$.
הציעו אלגוריתם שמוצא את המסלול הקל-קצר ביותר מ-s לכל $v \in V$.

- כלומר מבין כל המסלולים הקצרים נעדיף את אלו עם מספר הקשתות הנמוך ביותר.

<u>פיתרון:</u>

נרצה לתת עדיפות למסלולים עם מספר הקשתות הנמוך ביותר:

$$w'(u,v) = |V|\cdot w(u,v) + 1$$

<u>האלגוריתם:</u>

- חשב את w'
- הרץ את דייקסטרה מ-s לפי w'.

<u>הוכחת נכונות:</u> P המסלול הקל-קצר ביותר $\Leftrightarrow$ P המסלול קצר ביותר לפי w'.

ניקח 2 מסלולים פשוטים (ללא מעגלים) P,Q כך ש- $w(Q) > w(P)$. אזי:

$$w'(Q) - w'(P) = \sum_{(u,v)\in Q}(|V|\cdot w(u,v)+1) - \sum_{(u,v)\in P}(|V|\cdot w(u,v)+1)$$

$$= (|V|\cdot w(Q) + |Q|) - (|V|\cdot w(P) + |P|)$$

$$= |V|\cdot\big(w(Q) - w(P)\big) + |Q| - |P|$$

* מכיוון ש-P מסלול פשוט ללא חזרה על קודקודים מתקיים $|P| \leq |V|-1$
* $w(Q) - w(P) \geq 1$ כי המשקלים שלמים.

$$\Rightarrow w'(Q) - w'(P) \geq |V| - |V| + 1 > 0$$

$$\Rightarrow w'(Q) > w'(P)$$

כלומר אם P הוא המסלול הקל-קצר לפי w הוא גם המסלול הקצר ביותר לפי w'.

בנוסף נבחין כי אם $w(P) = w(Q)$ אז $w'(P) > w'(Q)$ אם"ם $|P|<|Q|$.

זרימה

<u>רשת זרימה:</u> גרף מכוון ומכוון $G = (V, E)$ עם פונקציית **קיבולת** $C: V \times V \rightarrow \mathbb{R}$ כך ש-

- לכל $(u, v) \in E$ מתקיים ש- $c(u, v) > 0$
- לכל $(u, v) \notin E$ מתקיים ש- $c(u, v) = 0$

בנוסף יש קודקוד מקור s וקודקוד יעד t.

<u>זרימה:</u> נתונה רשת זרימה G כנ"ל. **זרימה** ב-G היא פונקציה ממשית $f: V \times V \rightarrow \mathbb{R}$ המקיימת 3 תכונות:

1) אילוצי קיבולת – לכל $u, v \in V$ מתקיים $f(u, v) \leq c(u, v)$.
2) סימטריה נגדית – לכל $u, v \in V$ מתקיים $f(u, v) = -f(v, u)$ (זרימה מ-u ל-u שווה ל-0)
3) שימור הזרימה – לכל $u \in V/\{s, t\}$ דורשים ש- $\sum_{v \in V} \mathrm{f(u, v)} = 0$

<u>ערך הזרימה</u> שיוצאת מהמקור – $|f| = \sum_{v \in V/\{s\}} f(s, v)$ (הזרימה מהמקור לקודקודים שיש קשת מהמקור אליהם).

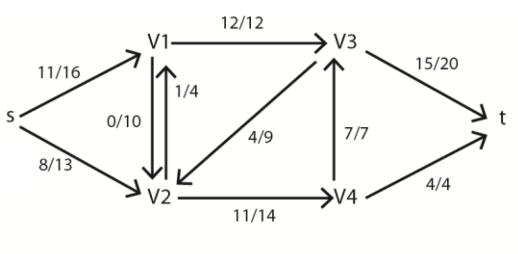
<u>רשת שיורית:</u> בהינתן רשת זרימה וזרימה, הרשת השיורית מורכבת מקשתות שיכולות להכיל זרימה נוספת.

<u>קיבולת שיורית של *(u,v)*</u>: $c_f(u, v) = c(u, v) - f(u, v)$

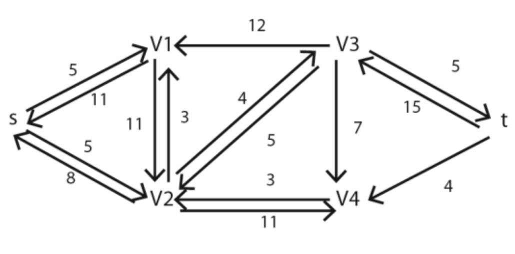
גרף השיורית של רשת זרימה G וזרימה f הוא גרף $G_f = (V, E_f)$ כאשר

$$E_f = \{(u, v) \in V \times V: c_f(u, v) > 0$$

דוגמא:



רשת זרימה G עם זרימה f=19

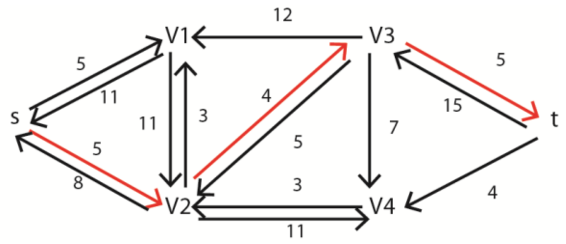


גרף השיורית של G עם זרימה f=19

הערה: (u,v)יכולה להיות קשת ב-G_f גם אם היא לא קשת ב-G כי כאשר $f(u,v)<0$ אז מקבלים $c_f(u,v) > 0$. לכן $|E_f| \leq 2|E|$.

<u>מסלול שיפור:</u> בהינתן רשת זרימה G וזרימה f, מסלול שיפור הוא מסלול פשוט מ-s ל-t ב-G_f.

לדוגמא:



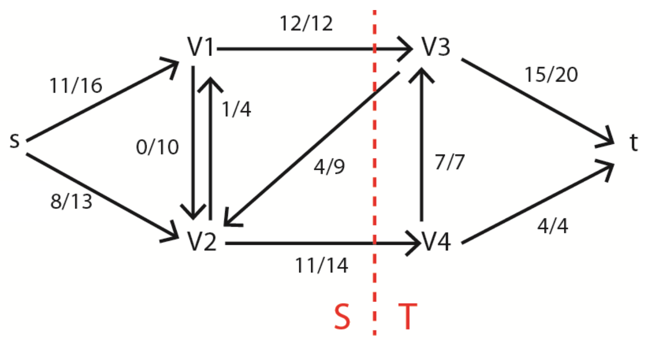
P (באדום) הוא מסלל שיפור עם קיבולת $c_f(P) = 4$.

<u>חתך (S,T)</u> של רשת זרימה היא חלוקת V ל-2 קבוצות S ו- $T=V/S$ כך ש- $s \in S$ ו-$t \in T$.

אם f זרימה, אז:

- $f(S,T)$ היא **הזרימה על החתך** ומוגדרת להיות $f(S,T) = \sum_{u\in S}\sum_{v\in T} f(u,v)$.
- $c(S,T)$ היא **הקיבולת על החתך** ומוגדרת להיות $c(S,T) = \sum_{u\in S}\sum_{v\in T} c(u,v)$.

לדוגמא:



הזרימה על החתך:

$$f(S,T) = f(v1,v3) + f(v2,v3) + f(v2,v4) = 12 + (-4) + 11 = 19$$

- $f(S,T)$ עשויה להכיל זרימות שליליות כי זה זרימה מ-S ל-T.

הקיבולת של החתך:

$$c(S,T) = c(v1,v3) + c(v2,v4) = 12 + 14 = 26$$

- $c(S,T)$ מכילה רק ערכים חיוביים.

<u>חתך מינימלי:</u> חתך שהקיבולת שלו היא המינימלית מבין כל החתכים.

משפט *Min-Cut Max-Flow*:

תהי f זרימה ברשת זרימה G. התנאים הבאים שקולים:

1) f זרימה מירבית ב-G
2) ברשת השיורית G_f אין מסלול שיפור מ-s ל-t.
3) קיים חתך (S,T) כך ש-$|f| = c(S,T)$ (כלומר (S,T) חתך מינימלי).

← ברשת זרימה מתקיים שהזרימה המקסימלית שווה לקיבולת של החתך המינימלי.

אלגוריתם *Ford-Fulkerson* למציאת זרימה מקסימלית:

<u>*Ford-Fulkerson(G, s, t):*</u>

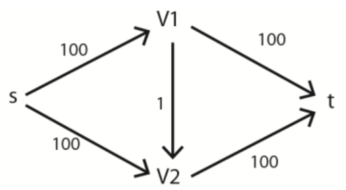
for each $(u,v) \in E$
  $f(u,v) = 0$
  $f(v,u) = 0$ — אתחול הזרימה ל-0
$G_f = G$
while there exists a path P from s to t in G_f — חיפוש מסלול באמצעות DFS
  $c_f(P) = \min\{c_f(u,v) \mid (u,v) \in P\}$ — הקיבולת של P זה הקיבולת של הקשת עם הקיבולת הכי נמוכה
  for each $(u,v) \in P$
    $f(u,v) = f(u,v)+c_f(P)$
    $f(v,u) = -f(u,v)$ — עדכון P ב-G
    $c_f(u,v) = c(u,v)-f(u,v)$
    $c_f(v,u) = c(v,u)-f(v,u)$ — עדכון P ב-G_f

סיבוכיות $O(|E|*f^*)$

שיפור: **האלגוריתם של *Edmonds-Karp*:**
הרעיון הכללי: האלגוריתם זהה ל-*FF* אך במציאת מסלול משפר, מחפשים את המסלול הקצר ביותר.
סיבוכיות: $O(|V|*|E|^2)$

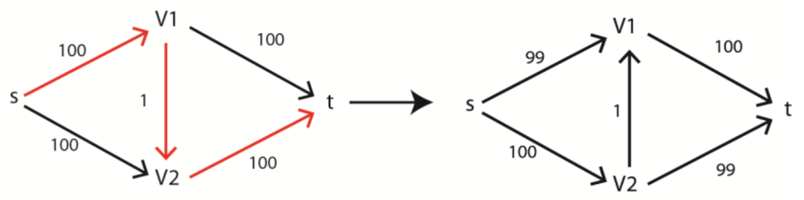
<u>תרגיל:</u>

נתונה רשת זרימה G. הוכח/הפרך: אם באיטרציה ה-i בהרצת פורד-פלקרסון הקשת e נעלמת מ-G_f, אזי e א תופיעה ב-G_f באיטרציות הבאות.



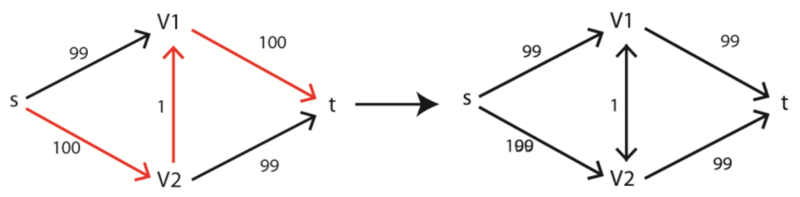
<u>פיתרון:</u>

לא נכון. דוגמא נגדית:



ניקח את המסלול $P=\{s,v1,v2,t\}$ עם קיבולת 1.

הקשת $(v1,v2)$ נעלמה ובמקומה הופיעה $(v2,v1)$



כעת ניקח מסלול שיפור $P=\{s,v2,v1,t\}$ (שוב עם קיבולת 1).

הקשת $(v1,v2)$ שוב מופיעה.

<u>הגדרה</u>: **רשת זרימה שלמה** היא רשת שבה כל הקיבולות הן מספרים שלמים. **זרימה שלמה** היא זרימה f בה מתקיים לכל $u,v \in V$ $\quad f(u,v) \in \mathbb{Z}$.

<u>תרגיל:</u>

הוכח/הפרך:

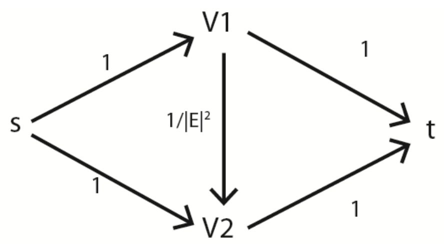
1) ברשת זרימה שבה הקיבולת של כל קשת קטנה מ-7, האלגוריתם של פורד-פלקרסון עושה *O(|E|)* איטרציות;.
2) כנ"ל ברשת זרימה לא שלמה

<u>פיתרון:</u>

1) נכון. הוכחה:
- נתון גרף כנ"ל.
- לכל חתך *(S,T)* מתקיים שמשקל החתך הוא לכל היותר

$$\sum_{(u,v)\in E} c(u,v) \leq \sum_{(u,v)\in E} 7 = 7|E|$$

- כל איטרציה של פורד-פלקרסון חייבת לשפר את הזרימה לפחות ב-*1* (הרשת שלמה),ולכן מספר השיפורים האפשריים הוא לכל היותר *7|E|* ← *O(|E|)*.



2) לא נכון. דוגמא נגדית:

נשפר את הזרימה לסירוגין על המסלולים:

{s,v1,v2,t}

{s,v2,v1,t}

האלגוריתם יסתיים אחרי $2|E|^2$ איטרציות כי הזרימה משופרת בכל איטרציה ב-$\frac{1}{|E|^2}$ והזרימה המקסימלית היא *2*.

<u>תרגיל:</u>

נתונה רשת זרימה *G = (V,E,c)* קבוצת מקורות $S = \{s_1, s_2, .., s_k\}$ וקבוצת יעדים $T = \{t_1, t_2, .., t_n\}$.
כיצד ניתן למצוא זרימה מקסימלית כאשר מותר להזרים מכל מקור לכל יעד?

פורמלית, הזרימה צריכה לקיים:

$f(u,v) \leq c(u,v)$

$f(u,v) = -f(v,u)$

$\sum_{v\in V} f(u,v) = 0$ לכל *u* למעט $u \in S \cup T$.

<u>פיתרון:</u>

ניצור רשת חדשה G_0 ע"י הוספת:

- מקור "על" s_0 עם קשת (s_0,s) לכל $s \in S$ שהקיבולת שלהן היא ∞ (כלומר אין מגבלה על הזרימה)
- יעד "על" t_0 עם קשת (t_0,t) לכל $t \in T$ שהקיבולת שלהן היא ∞.

נמצא זרימה מקסימלית ב- G_0 מ-s_0 ל-t_0.

נסיר את s_0 ו-t_0 ונקבל את הזרימה המקסימלית ב-G.

רכיבים קשירים היטב- *Strongly connected components*

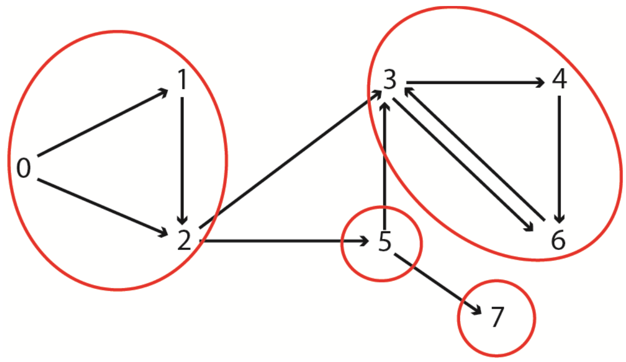
- בגרף לא מכוון –

נתון גרף מכוון $G = (V, E)$. נגדיר יחס R על V המסמל "**קשירות חזקה**" באופן הבא:

$(x, y) \in R$ אם קיים מסלול מ-x ל-y ומ-y ל-x ב-G.

מכיוון שהיחס רפלקסיבי, סימטרי וטרנזיטיבי, <u>הוא יחס שקילות</u>, המחלק את V למחלקות שקילות כל שכל מחלקה היא "**רכיב קשיר היטב**".

דוגמא:



אלגוריתם *Tarjan* למציאת רכיבים קשירים היטב:

N- מונה לסריקה סגרף
S- מחסנית
T – עץ פורש שנבנה בזמן ה-*DFS*.
dfs-num – מערך שמחזיק את המספר הסידורי בזמן הסריקה
low– מערך שמחזיק את ערך ה-*dfs-num* הנמוך ביותר אליו מצביע אחד מצאצאיו (צאצא של x בעץ *T*).

<u>Tarjan (G):</u>
create new node x, and (x,v) for each $v \in V$
N=0
S- empty stack
T – tree with x as root
Visit(x)

```
Visit(p)
S.push(p)
dfs-num(p) = N
low(p) = N
N++
for each edge (p,q)
        if q is not in T
                add(p,q) to T
                Visit(q)
                low(p) = min{low(p), low(q)}
        else
                low(p) = min{low(p), dfs-num(q)}
if low(p) == dfs-num(p) (≠ 0 – doesn't print x)
        print("SCC")
        do
                v = S.pop()
                print(v)
                remove v from G
        while v ≠ p
```

סיבוכיות: $O(|V|+|E|)$ (כמו *DFS*).

נקודות חשובות:

- כמו *DFS* עם כמה שינויים.
- מוסיפים קודקוד חדש וממנו קשת לכל קודקוד בגרף, ועוברים באמצעות *DFS* על הגרף. כך מובטח שנעבור על כל הקודקודים.
- הרכיבים הקשירים הם תתי-עצים ב-T.
- על מנת לגלות אותם צריך לחתוך קשתות מסוימות ומה שנשאר הם הרכיבים.
- נקבע ששורש של תת-עץ הוא הקודקוד העליון ביותר (צריך לקבוע אם V מסוים הוא שורש כזה).