תכנון וניתוח אלגוריתמים
הרצאה 17

רכיבי קשירות ורק"חים

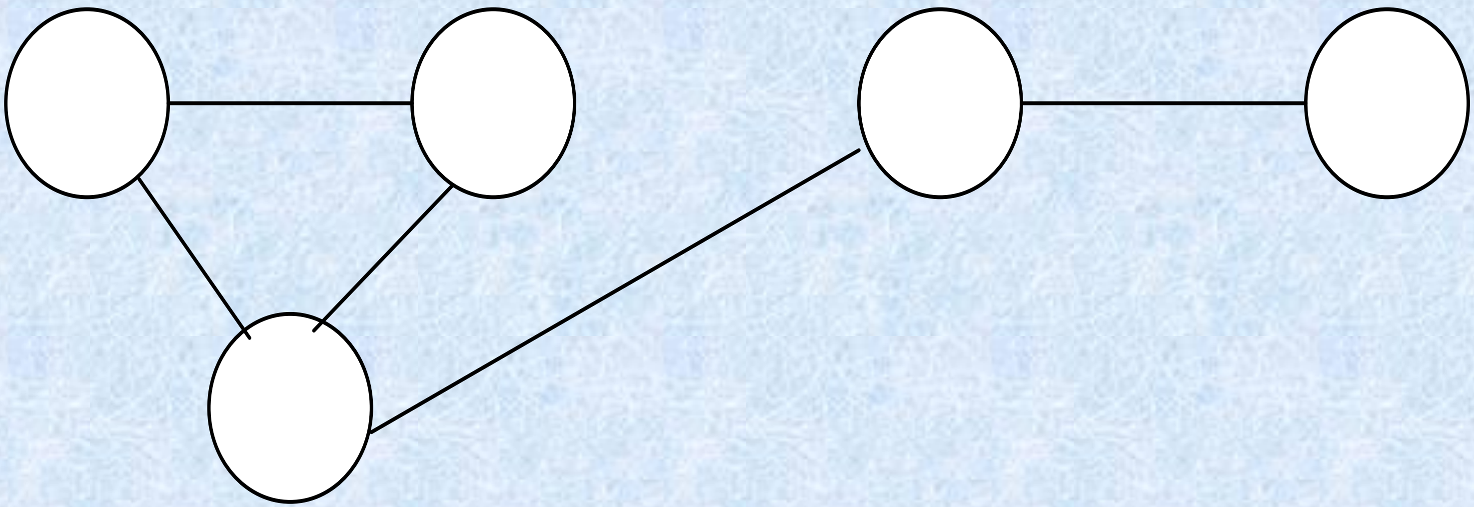
רכיבי קשירות

- **הגדרה** : גרף לא מכוון נקרא קשיר אם בין כל שני קדקודים קיים מסלול.
- **הגדרה** : נתון גרף לא מכוון $G = (V,E)$.

  רכיב קשיר זוהי קבוצה מקסימלית של קדקודים בגרף, כך שבין כל שני קדקודים בקבוצה זו קיים מסלול.

  קדקוד בודד ללא שכנים יקרא גם כן רכיב קשיר.

◈ דוגמא 1 : בתרשים הבא מוצג גרף קשיר.



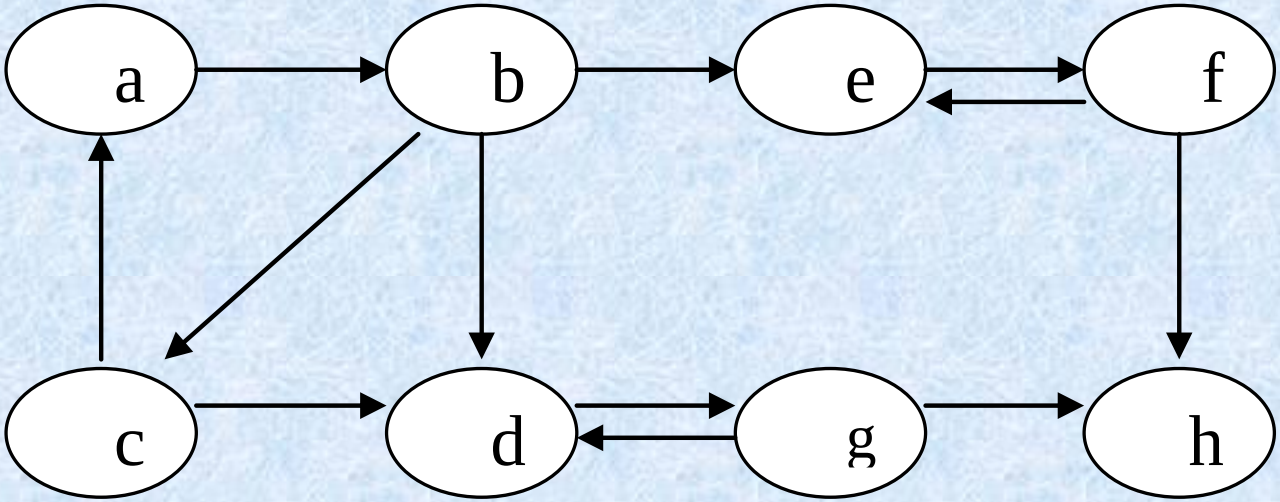
דוגמא 2 : בתרשים הבא מוצג גרף שאינו קשיר.

בדוגמא מס׳ 2 נתון גרף שבו שלושה רכיבים קשירים והם :
{ g }   { e,f }   { a,b,c,d } .

- <u>הגדרה</u> - גרף מכוון יקרא "גרף קשיר חזק" אם קיים מסלול מכוון מכל קדקוד לכל קדקוד אחר .
- * נתון גרף מכוון $G=(V,E)$.
- <u>הגדרה</u> -"רכיב קשיר חזק" (בקיצור רק"ח) זוהי קבוצה מקסימלית של קדקודים בגרף G, כך שבין כל שני קדקודים בקבוצה זו קיים מסלול מכוון.

◈ דוגמא בתרשים הבא המתאר גרף מכוון ישנם 4 רכיבי קשירות חזקה.

◈ הרכיבים הם :



◈ {a,b,c}

◈ {d,g}

◈ {e,f}

◈ {h}

- בגרף לא מכוון G , DFS(G) מוצא את רכיבי הקשירות.
- BFS(G,a) מוצא את רכיב הקשירות ש- a נמצא בו.
- אפשר לשכתב בקלות את האלגוריתם BFS(G) כך שנוכל למצוא את כל רכיבי הקשירות.

אלגוריתם למציאת רכיבים קשירים בגרף לא מכוון בעזרת BFS

- בתהליך של סריקת קדקודי הגרף בשיטת BFS נשתמש במערך בוליאני Used, כך ש Used[v] מציין האם ביקרנו בקדקוד v או לא.
- בנוסף נשתמש במשתנה count אשר מונה את מספר הרכיבים הקשירים.
- להלן אלגוריתם המבוקש:

- 1. count ← 0
- 2. לכל קדקוד v בגרף בצע:
- 2.1 אם ל- Used[v] ערך FALSE אז בצע:
- 2.1.1 קרא לשגרה BFS[G] כאשר קדקוד מקור הינו v.
- 2.1.2 count ← count + 1

- <u>הערות:</u>
- 1. קל לראות שסיבוכיות זמן הריצה של האלגוריתם הינה $O(|E| + |V|)$.
- 2. אם ברצוננו להדפיס את כל הקדקודים שנמצאים באותו רכיב קשיר של V, נוכל לעשות זאת בצורה רקורסיבית בעזרת פרישת עץ בשיטת inorder, או preorder, בזמן $O(|V|)$.

- 3. ברצוננו לבדוק האם קיים מסלול בין 2 צמתי גרף.
- אם התשובה שלילית נרצה להדפיס הודעה שאין מסלול
- כזה .
- אחרת נדפיס את המסלול עצמו.
- ***<u>פתרון</u>***
- קודם נריץ את האלגוריתם BFS(G)
- אחרי כן נפעיל את השגרה הרקורסיבית הבאה אשר מדפיסה את הקדקודים לאורכו של מסלול קצר ביותר מ-s ל-v:

- ***הדפס_מסלול (G,s,v)***
- אם v==s אז הדפס את הקדקוד s
- אחרת אם P[v]=nil
- אז הדפס שלא קיים מסלול ועצור!
- אחרת בצע:
- א. **הדפס_מסלול**(G,s,P[v]) (* רקורסיה*)
- **ב.** הדפס את הקדקוד v .

- 4. באלגוריתם ,למציאת רכיבים קשירים בגרף לא מכוון,

במקום המשפט:

"קרא לשגרה BFS(G) כאשר קדקוד מקור הינו v".

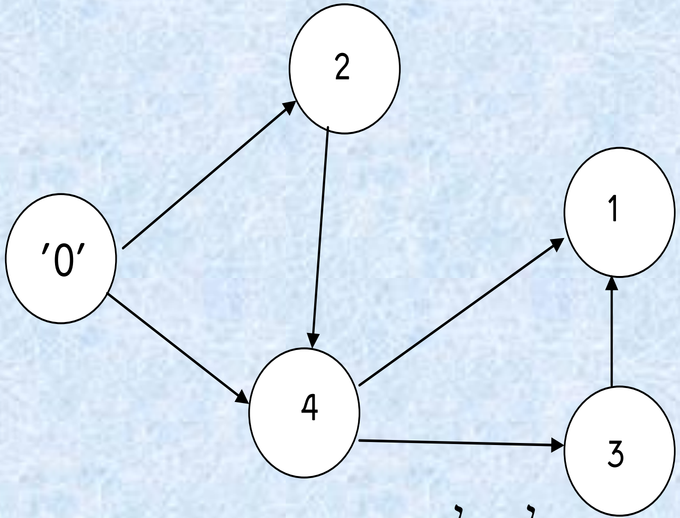
ניתן לרשום את המשפט הבא :

- "קרא לשגרה DFS(G) , כאשר קדקוד מקור הינו v"
- בסוף האלגוריתם תוחזר אותה תוצאה.

קבלת גרף הפוך G^T מ- G.

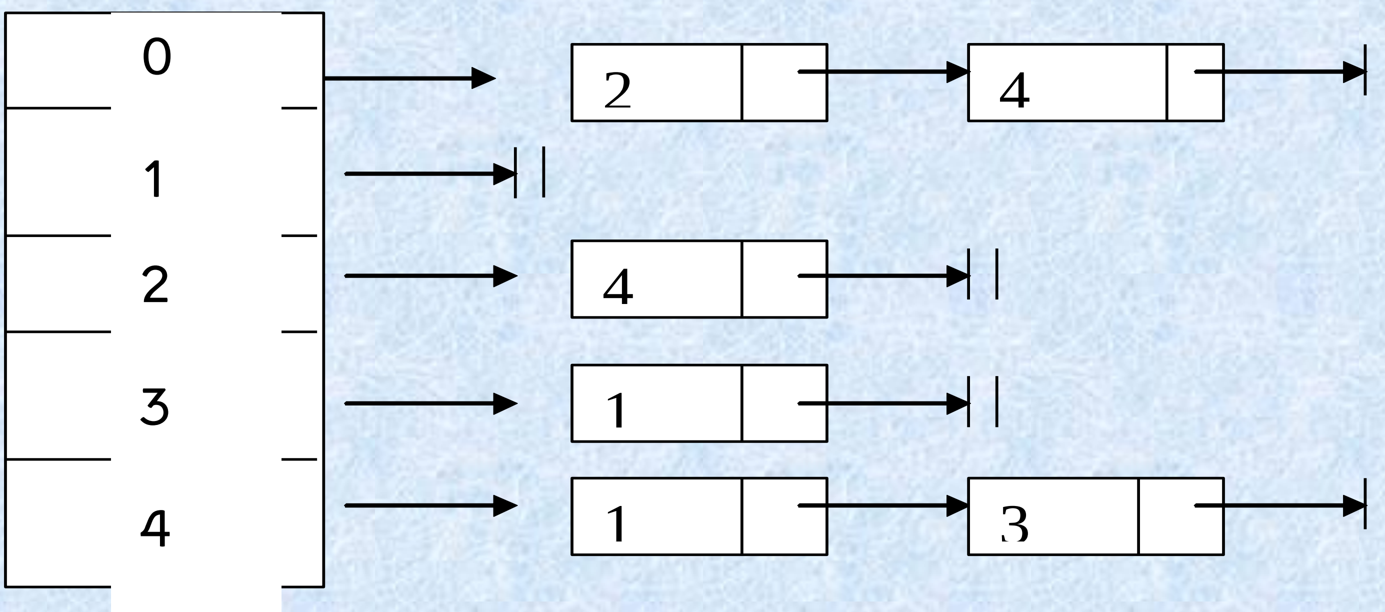
- דוגמא: נתון גרף $G=(V,E)$.
- נגדיר גרף חדש אשר יקרא גרף הפוך ומוגדר כדלקמן:
- $G^T=(V ,E^T)$
- כאשר $E^T = \{ (u,v) \mid (v,u) \in E\}$
- כלומר הופכים את כיווני הקשתות .
- קל לכתוב אלגוריתם בעל זמן הריצה $O(|E| +|V|)$ אשר קולט את הגרף G ויחזיר את הגרף G^T.

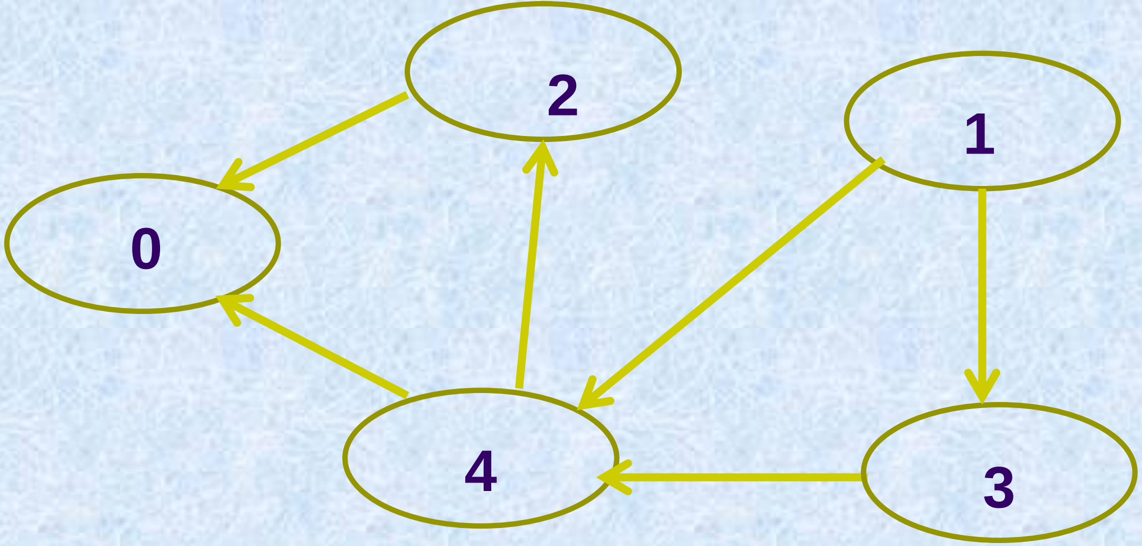
דוגמה-



נתון הגרף הבא:

- גרף זה מיוצג באמצעות רשימות סמיכות שלהלן:

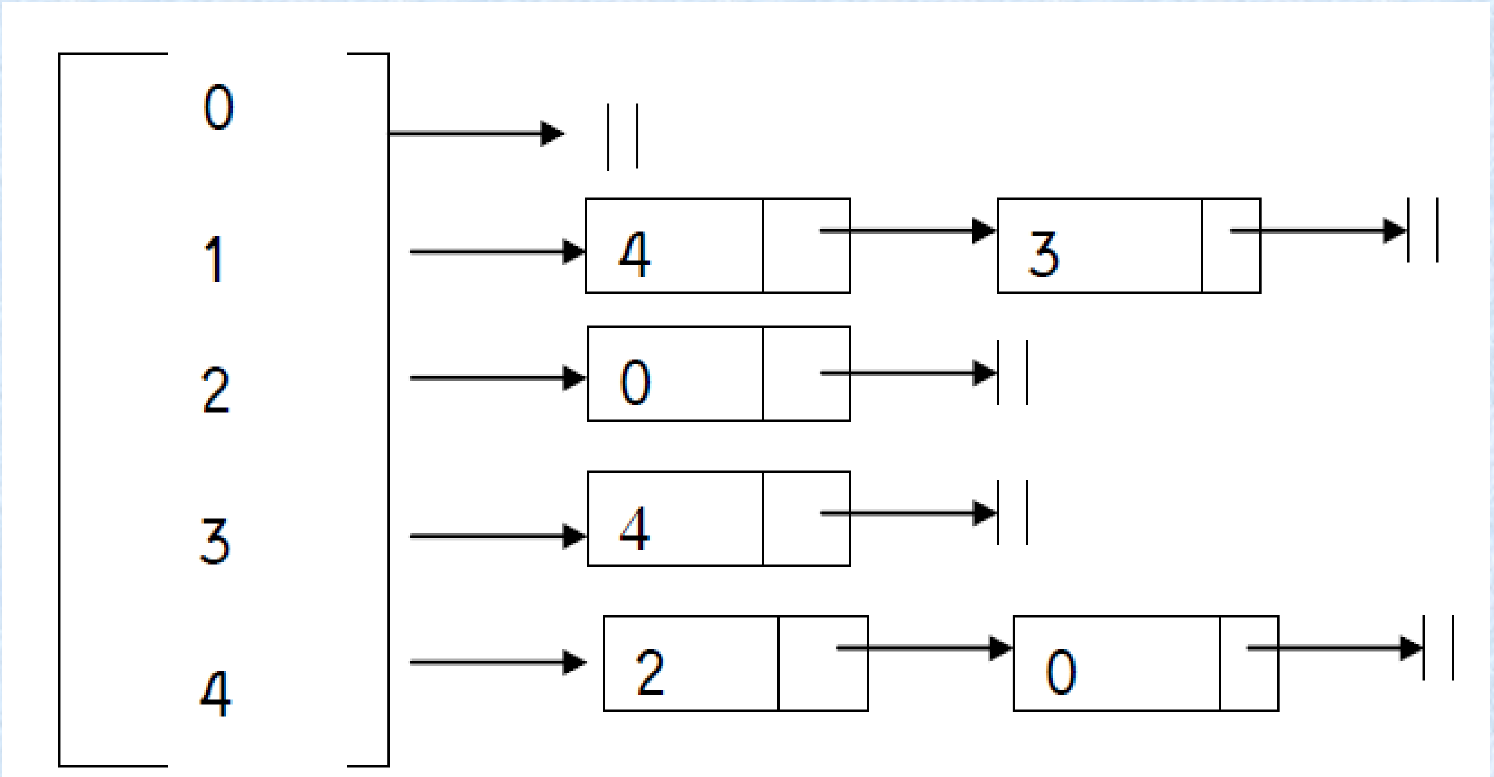




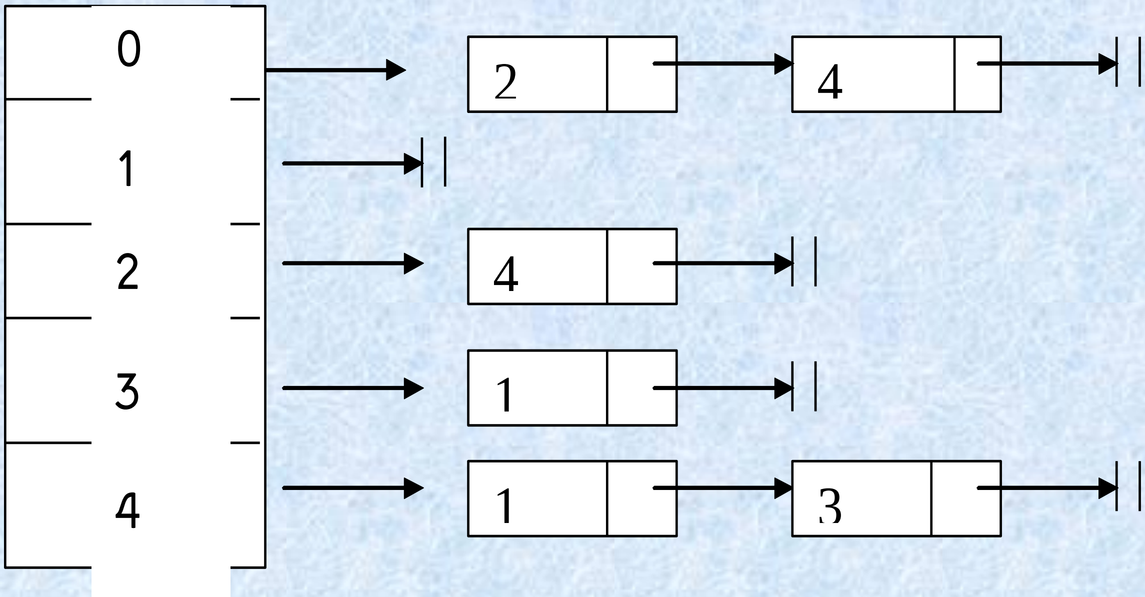
המטרה

לבנות גרף הבא:

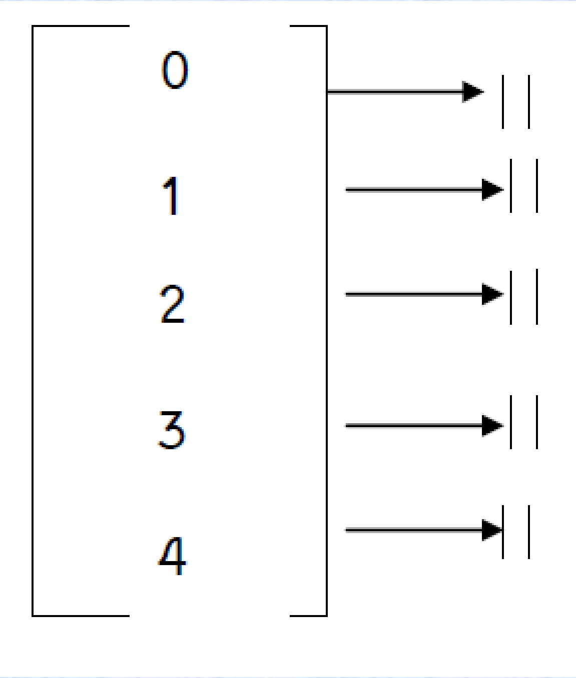
- המיוצג
- כך:



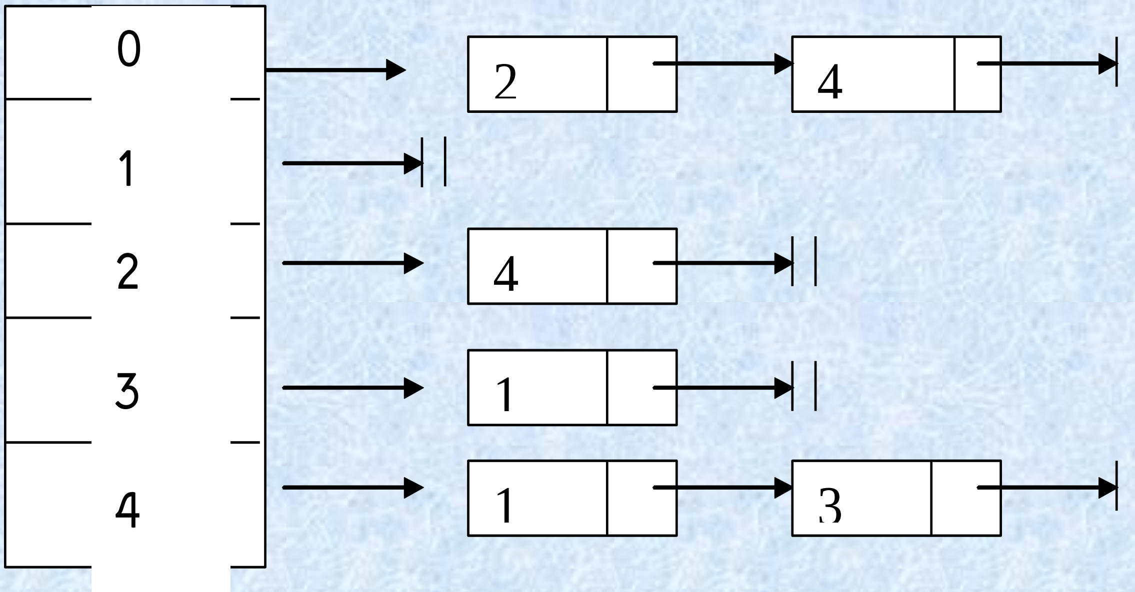
איך בונים ממנו גרף הפוך?



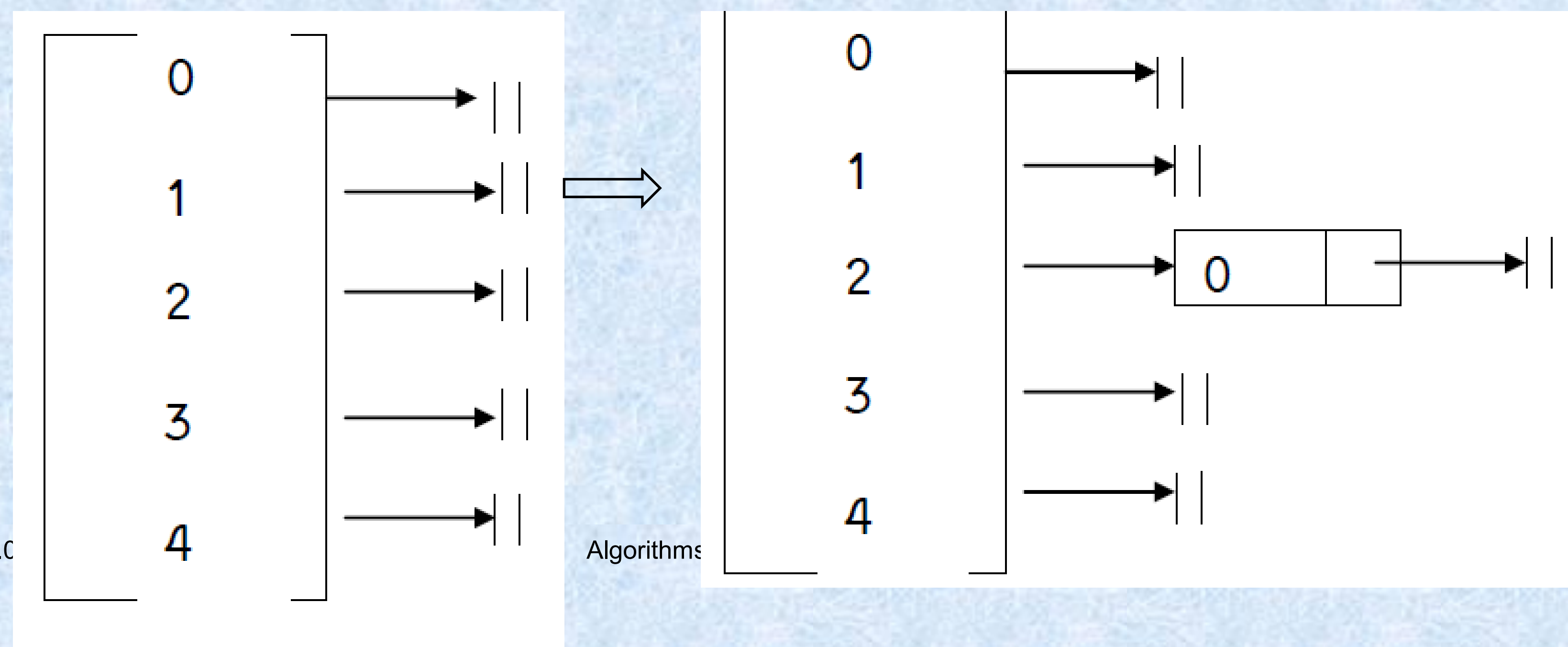
- בהתחלה



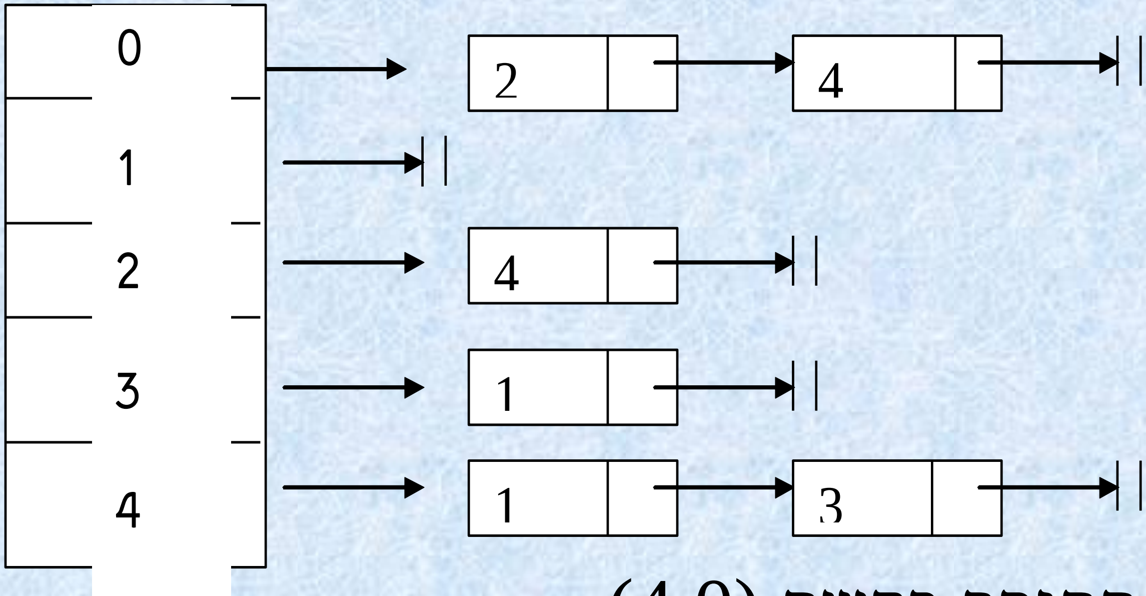
איך בונים ממנו גרף הפוך?



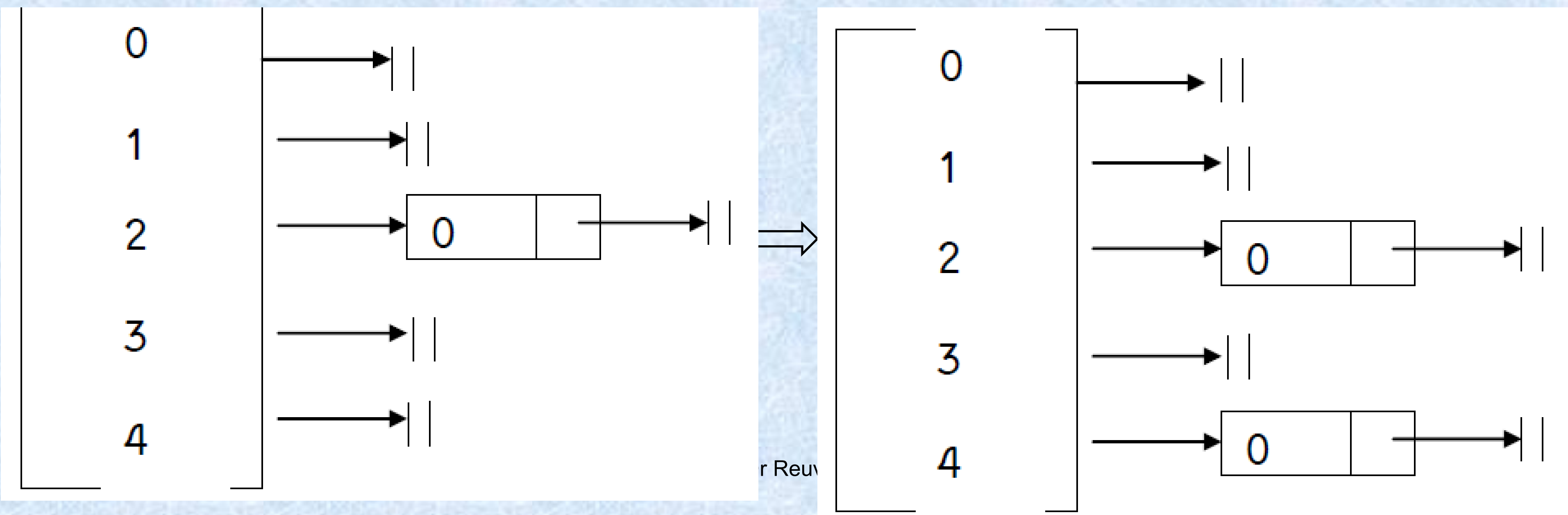
- בהתחלה קשת (0,2) תתוסף כקשת (2,0) .



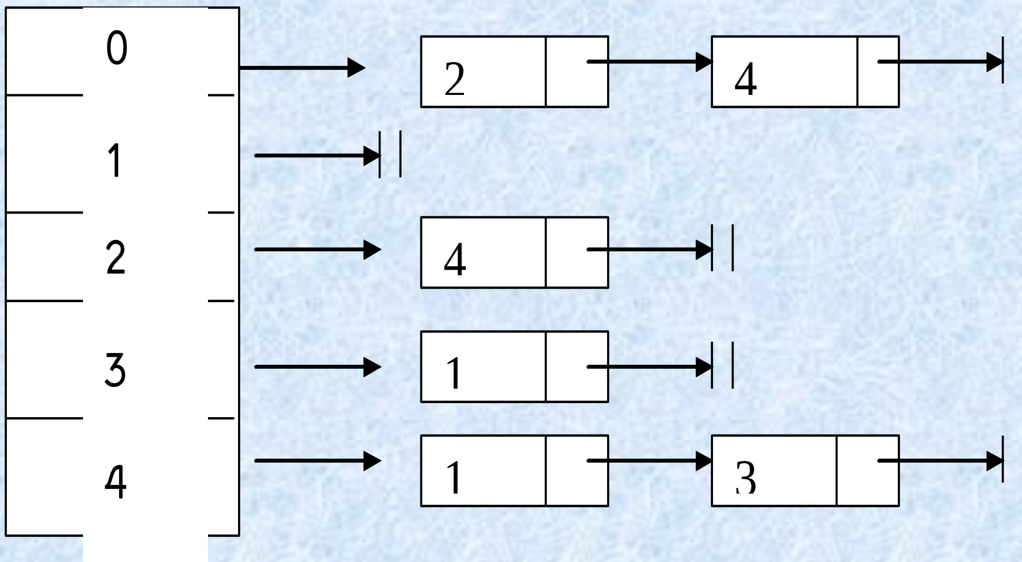
איך בונים ממנו גרף הפוך?



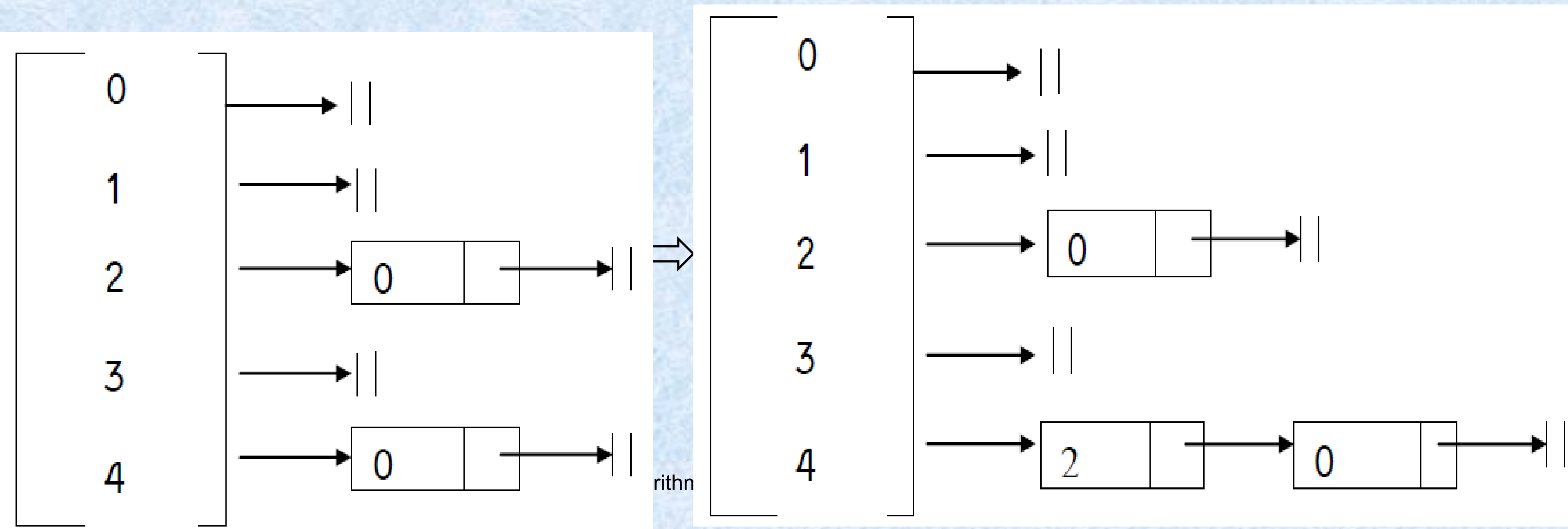
- בהתחלה קשת (0,4) תתוסף כקשת (4,0) .



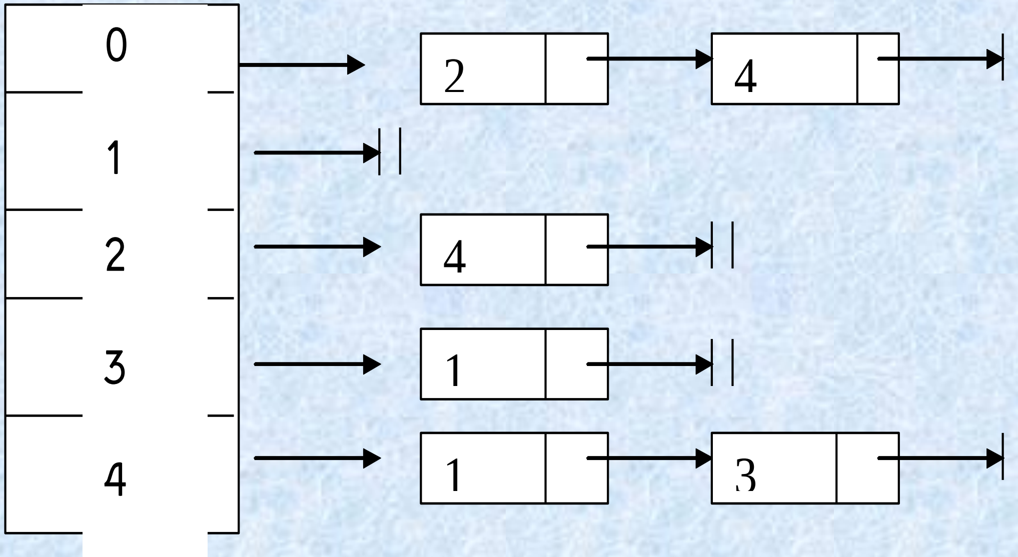
איך בונים ממנו גרף הפוך?



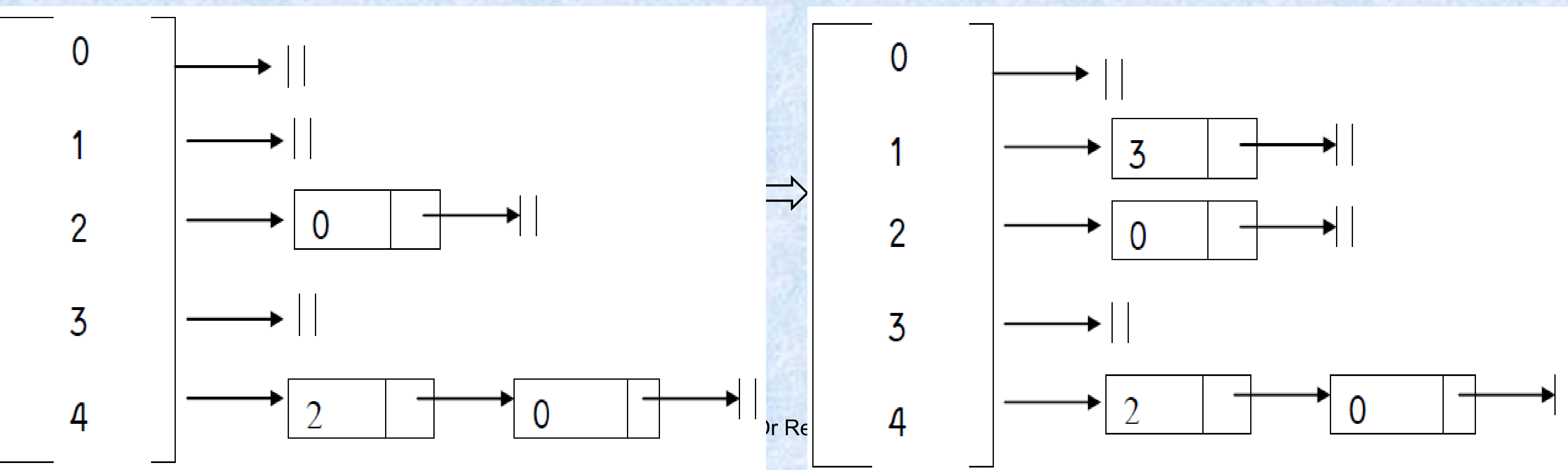
- בהתחלה קשת (2,4) תתוסף כקשת (4,2) בתחילת הרשימה.



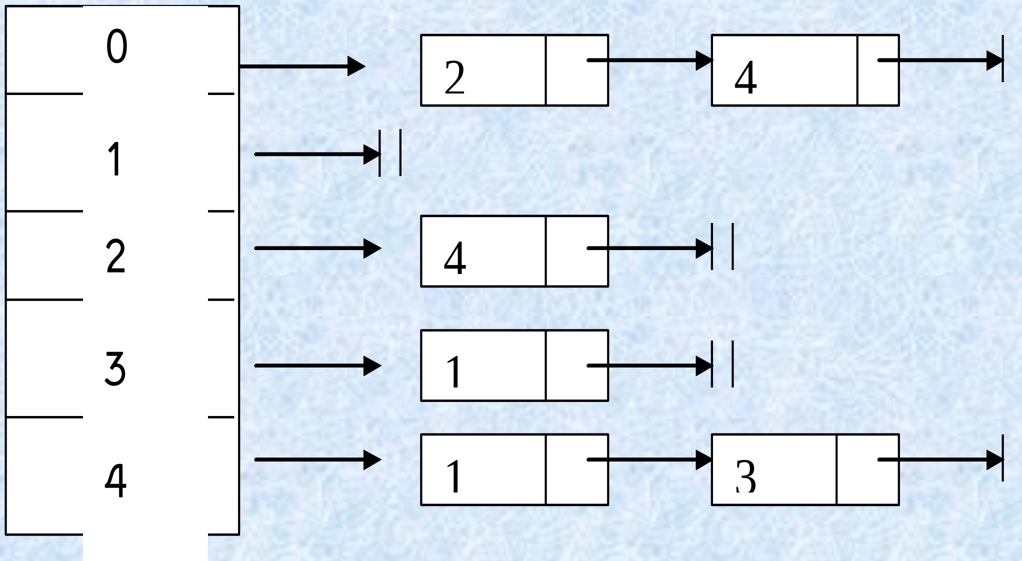
איך בונים ממנו גרף הפוך?



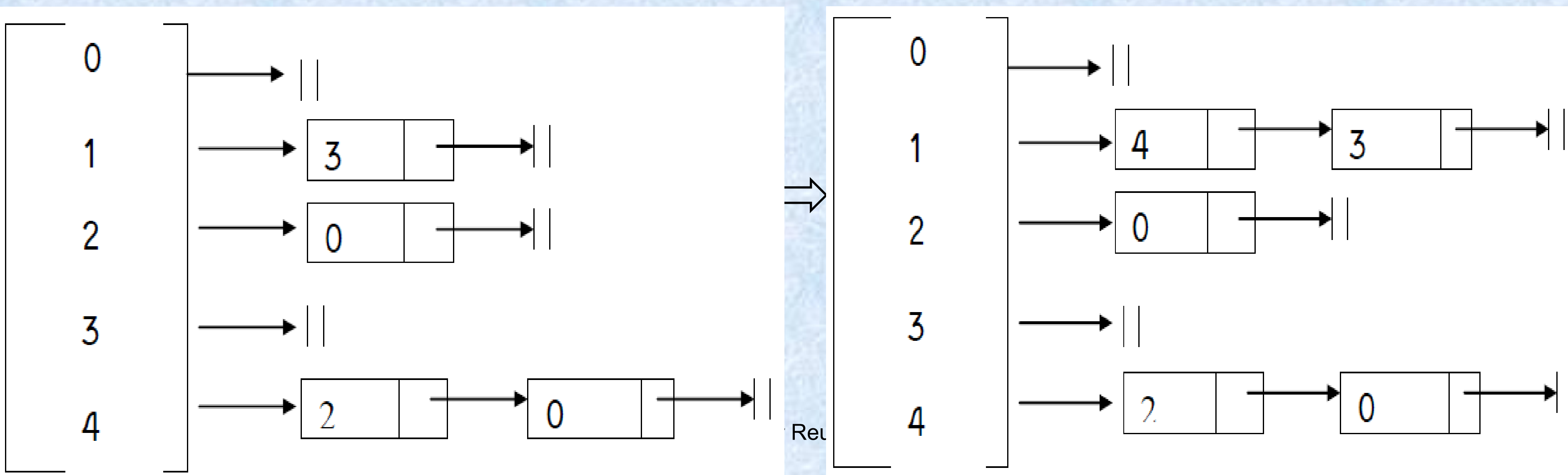
◈ בהתחלה קשת (3,1) תתוסף כקשת (1,3) בתחילת הרשימה.



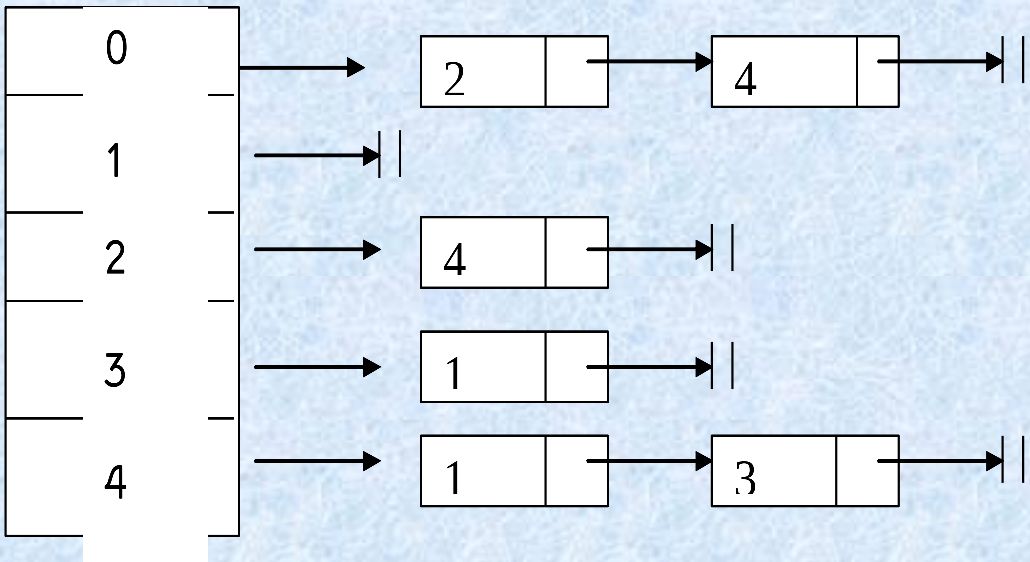
איך בונים ממנו גרף הפוך?



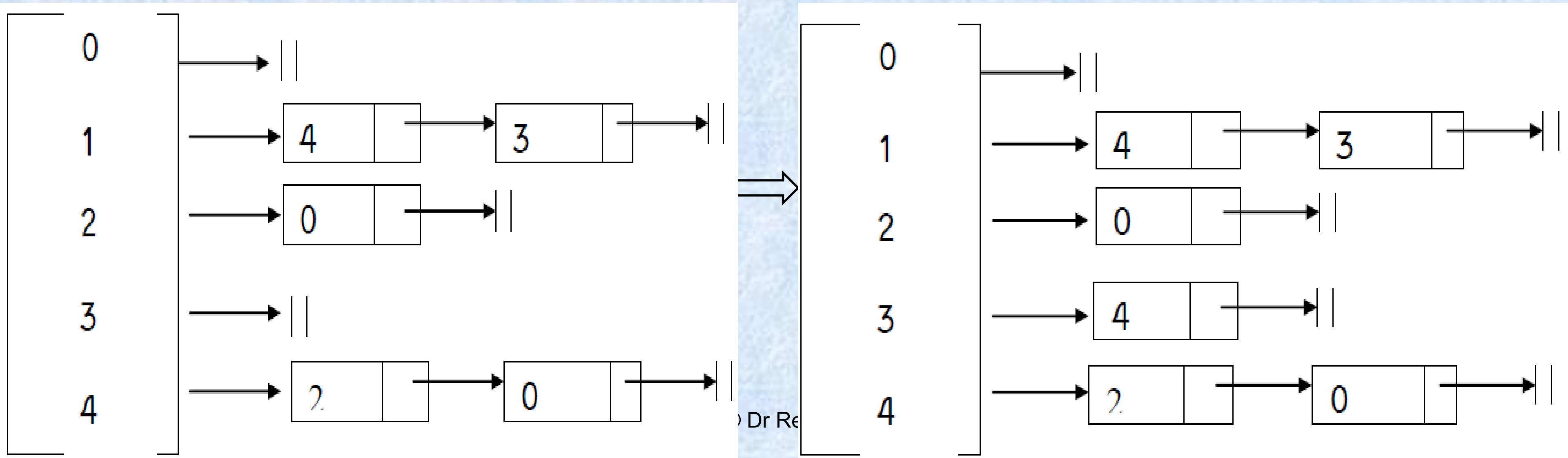
- בהתחלה קשת (4,1) תתוסף כקשת (1,4) בתחילת הרשימה.



איך בונים ממנו גרף הפוך?



◈ בהתחלה קשת (4,3) תתוסף כקשת (3,4) בתחילת הרשימה.



סיכום

- בסוף התהליך קיבלנו את המצופה.
- רואים בבירור שמספר הצעדים שאנו עושים באלגוריתם זה פרופורציוני לסכום אורכי רשימות הסמיכות.
- לכן סיבוכיות זמן הריצה של האלגוריתם הינה $O(|E| + |V|)$.

אלגוריתם למציאת רכיבי קשירות חזקה בגרפים מכוונים

- אלגוריתם למציאת רכיבי קשירות חזקה בגרפים מכוונים
- (SCC) Strong Connected Component
- צעד 1. מריצים את DFS(G) ויוצרים רשימת קדקודים (L)
- אשר ממוינת בסדר יורד לפי זמני סיום הטיפול בהם.
- צעד 2. הופכים את הגרף $G=(V,E)$ ומקבלים $G^T=(V,E^T)$

כאשר

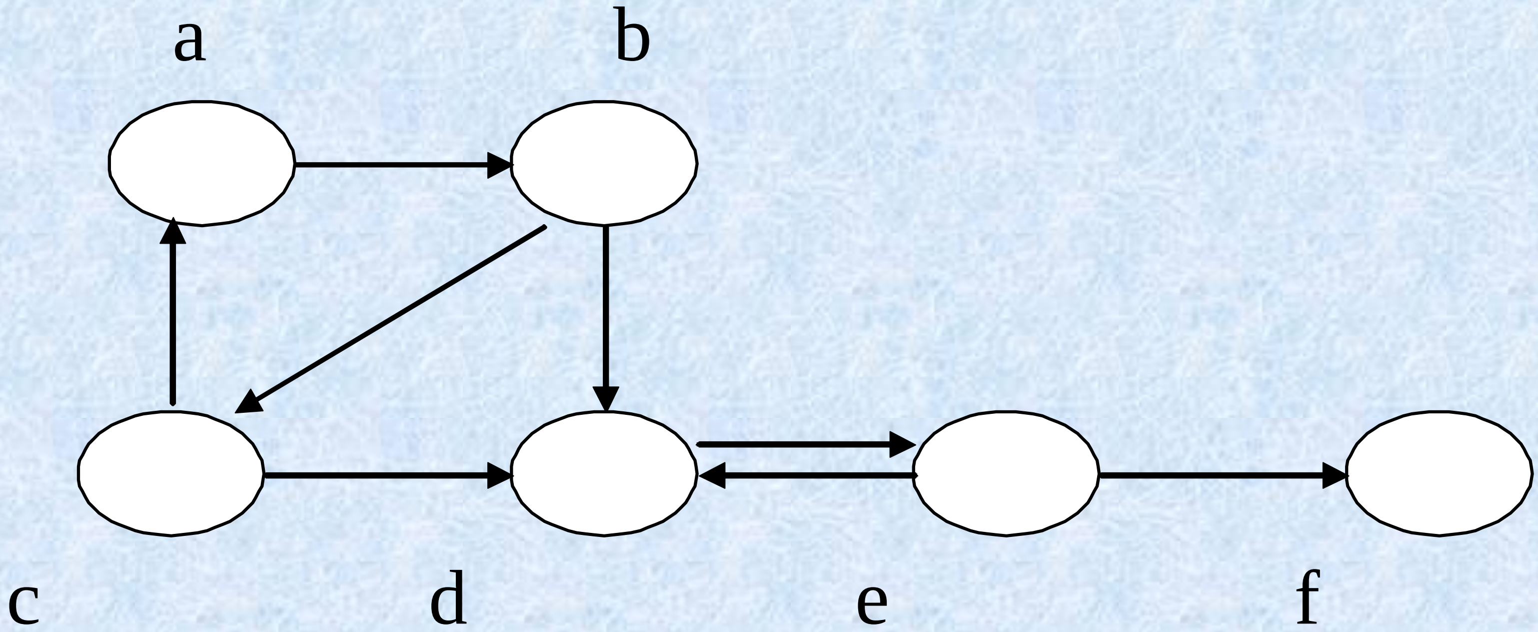
$$E^T = \{ (u,v) \mid (v,u) \in E \}$$

- כלומר הופכים את קשתות הגרף .
- צעד 3 . מריצים $DFS(G^T)$, כך שהלולאה המרכזית של
- DFS עוברת על קדקודי הגרף לפי הסדר , כפי
- שנקבע בצעד 1 ברשימה L .

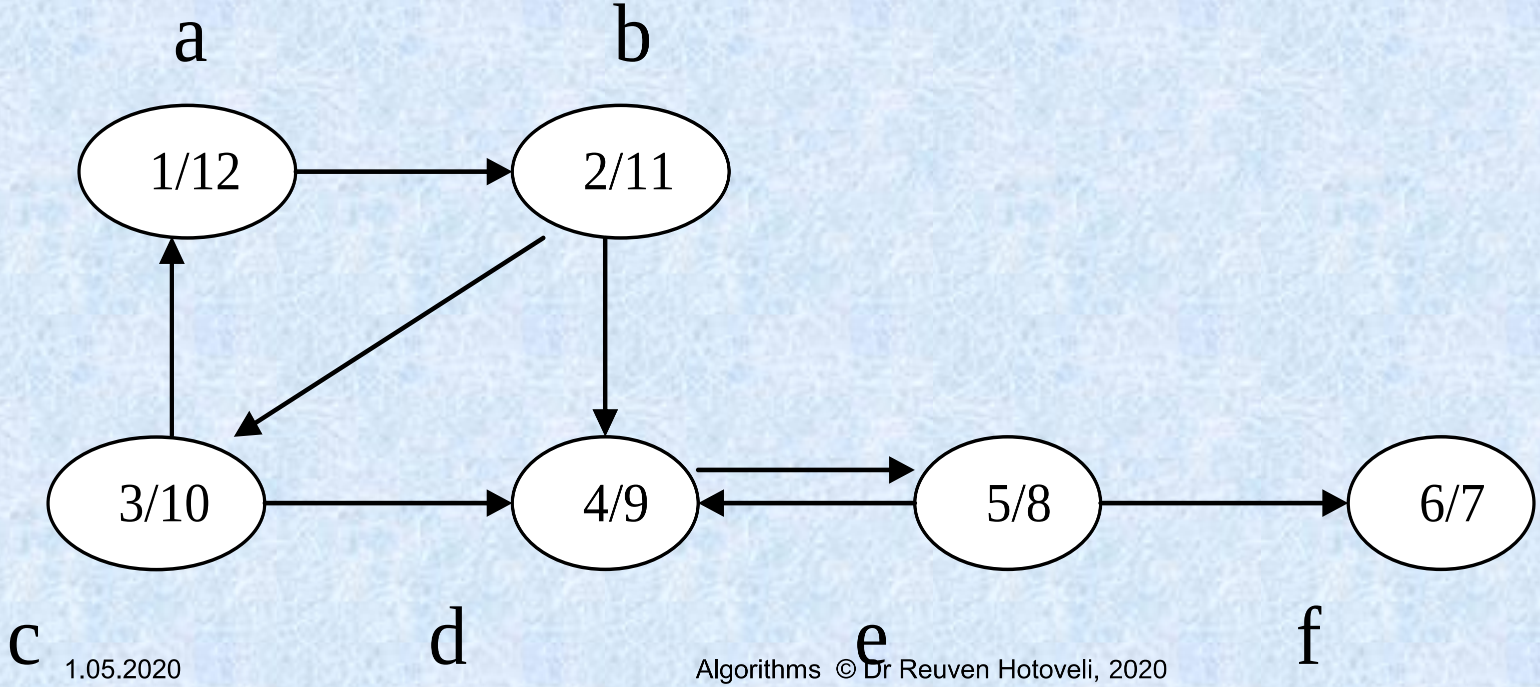
- <u>יעילות האלגוריתם</u>
- צעד 1 דורש זמן $O(|E| + |V|)$
- צעד 2 דורש זמן $O(|E| + |V|)$
- צעד 3 דורש זמן $O(|E| + |V|)$
- סופית : סיבוכיות זמן הריצה של האלגוריתם הנדון היא: $O(|E| + |V|)$.
- נדגים את אופן הפעולה של האלגוריתם (SCC) שתואר
- לעיל על הגרף הבא:

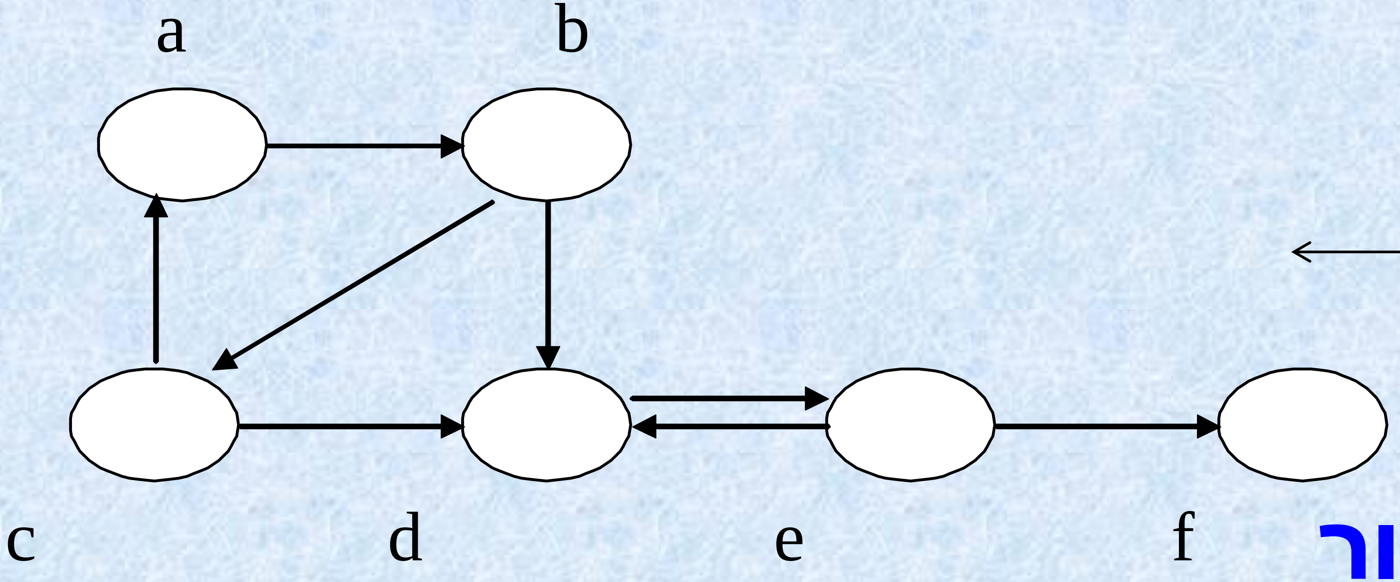
דוגמת הרצה

- נתון גרף מכוון הבא:



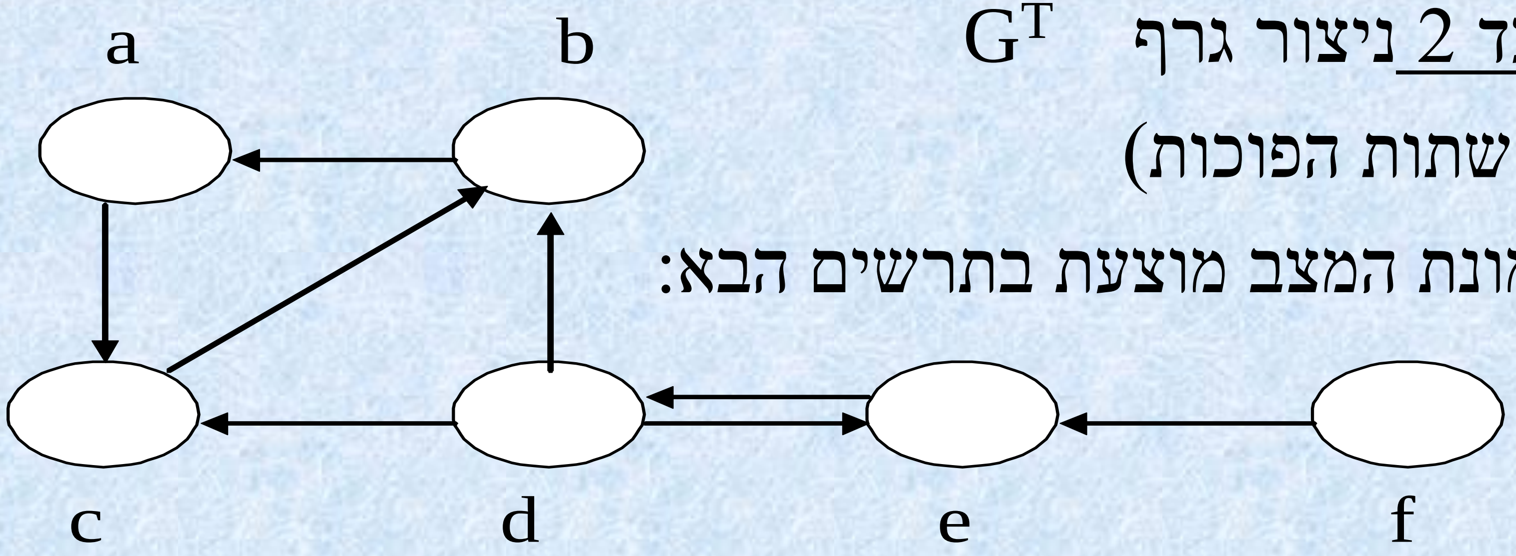
- צעד 1 נריץ את האלגוריתם DFS(G) כאשר קדקוד מקור הינו a.
- בשלב זה תמונת המצב מוצגת בתרשים הבא:





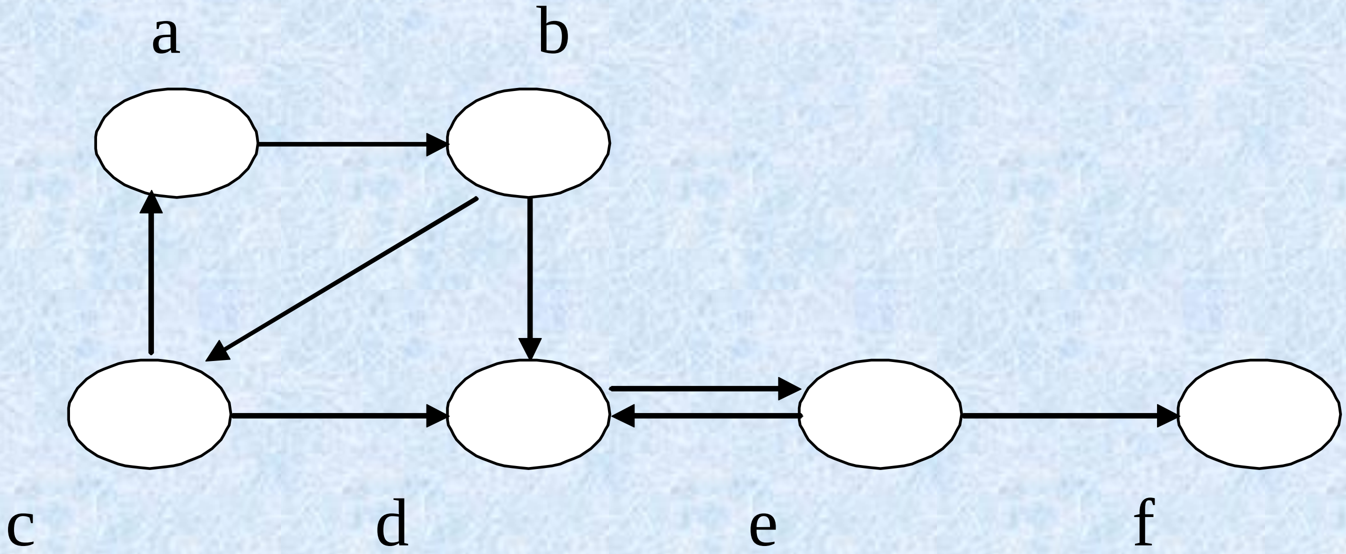
איור מקור

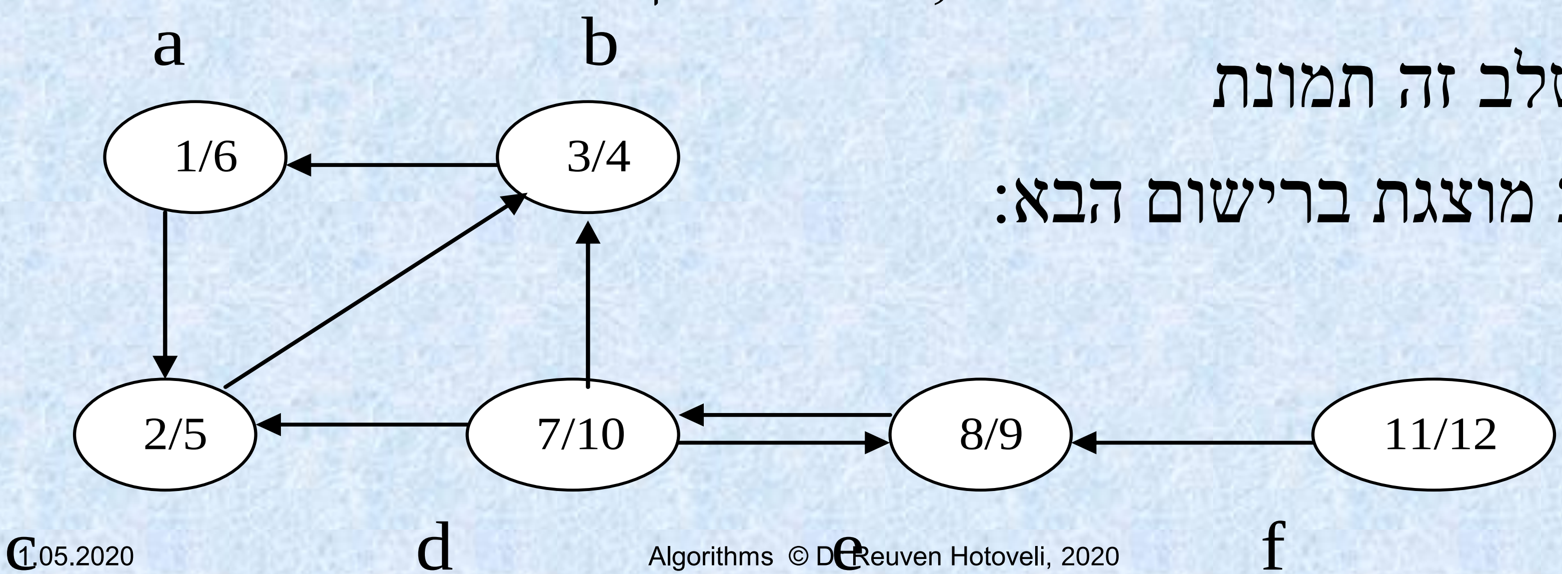
- ומקבלים כי: $L=\{\ a,b,c,d,e,f\ \}$
- <u>צעד 2</u> ניצור גרף G^T (בו קשתות הפוכות)
- תמונת המצב מוצעת בתרשים הבא:



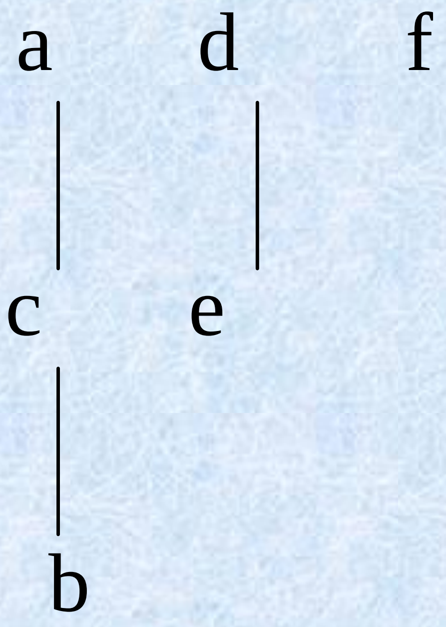
איור מקור

- צעד 3
- כעת נריץ את האלגוריתם DFS על הגרף G^T לפי הסדר כפי שמופיע ברשימה L, אשר התקבלה בצעד 1 .
- בשלב זה תמונת המצב מוצגת ברישום הבא:



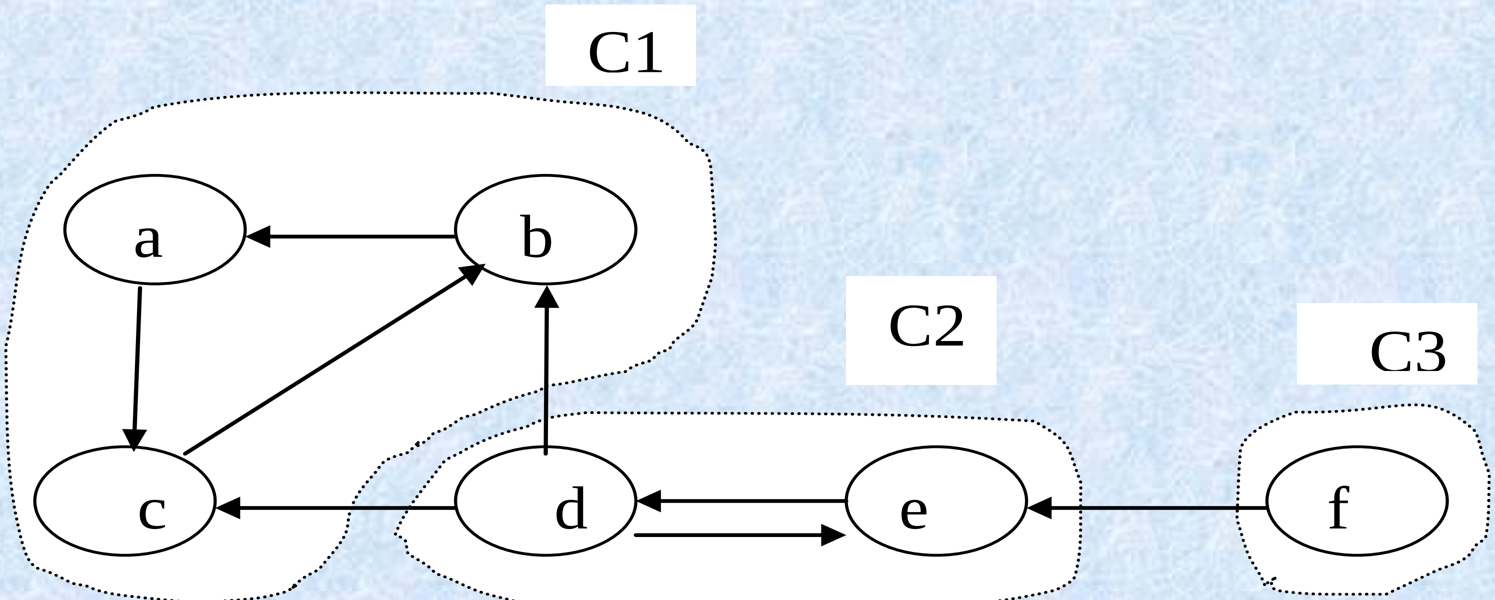


עצי פרישה DFS שמתקבלים הינם:



- לכן בגרף הנתון 3 רק"חים והם (כצפוי)
- {a,c,b}   {d,e}   {f}
- בהרצאה זו אנו לא נעסוק בנכונות האלגוריתם המודע כיוון שהוכחת נכונותו של השיטה חורגת מן הדרישות של התכנית.

- בהמשך לדוגמא קודמת קיבלנו 3 רכיבי קשירות חזקה
- (רק"ח) והם:
- C1={a,b,c}   C2={d,e}   C3={f}

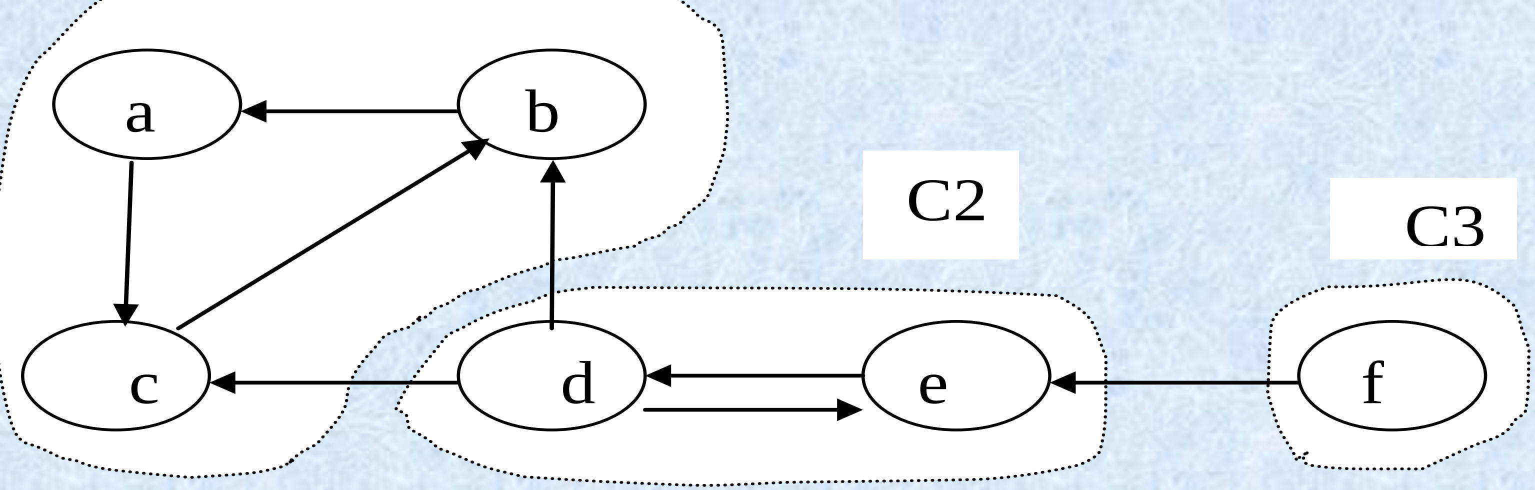


גרף על

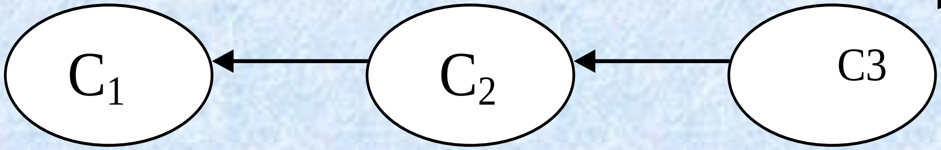
- <u>הגדרה</u>: <u>*גרף העל*</u> הוא: $G=(V,E)$ כאשר
- $V=\{C_1,C_2,\ldots\ldots.C_m$ / $m \geq 1$-ו C_i רק"ח$\}$
- $E=\{$ (C_i,C_j) / (x,y) קיימים $x\epsilon C_i$ ו- $y\epsilon C_j$ וקיימת קשת (x,y) בגרף G $\}$

בהמשך לדוגמה האחרונה גרף העל של הגרף ההפוך הנתון הוא הגרף הבא :

הגרף

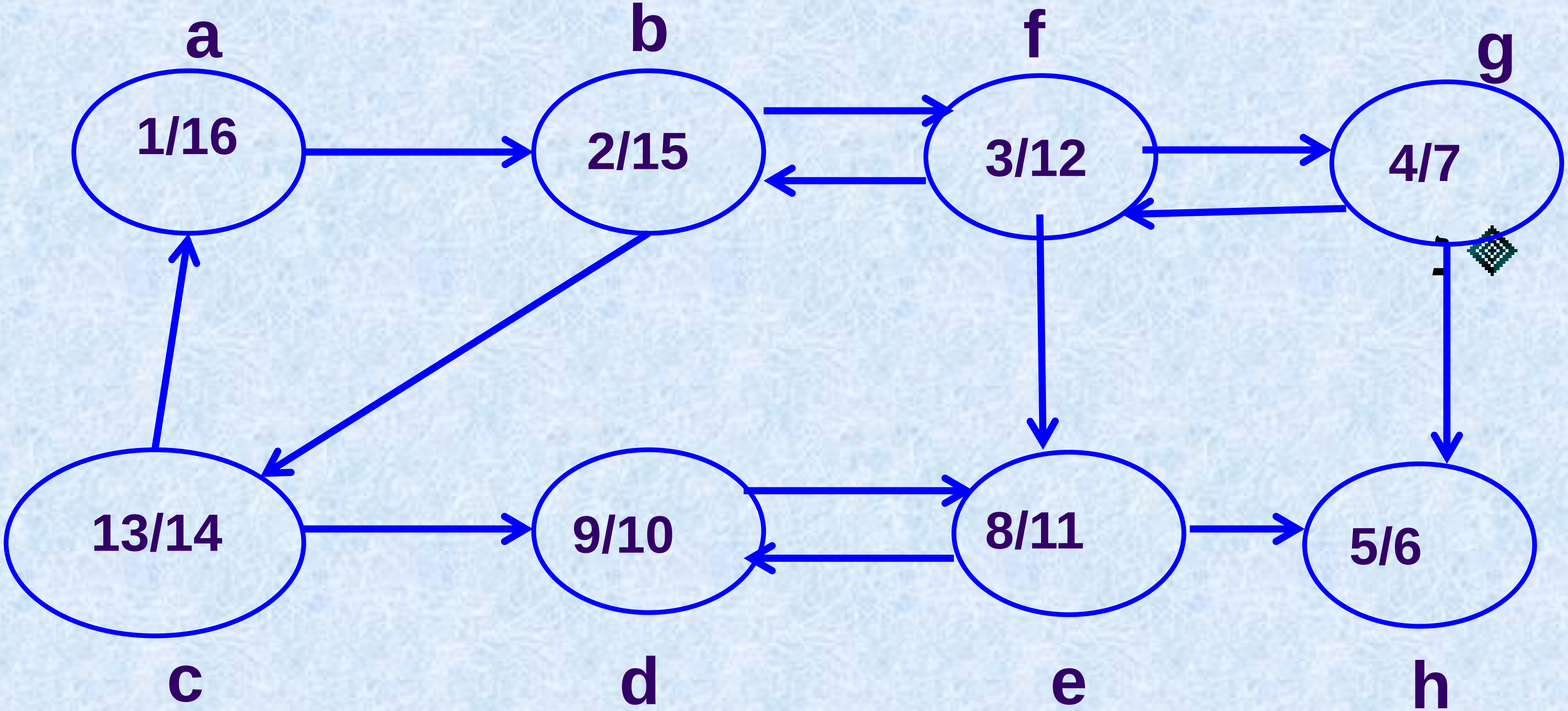


- גרף העל של הגרף הוא:



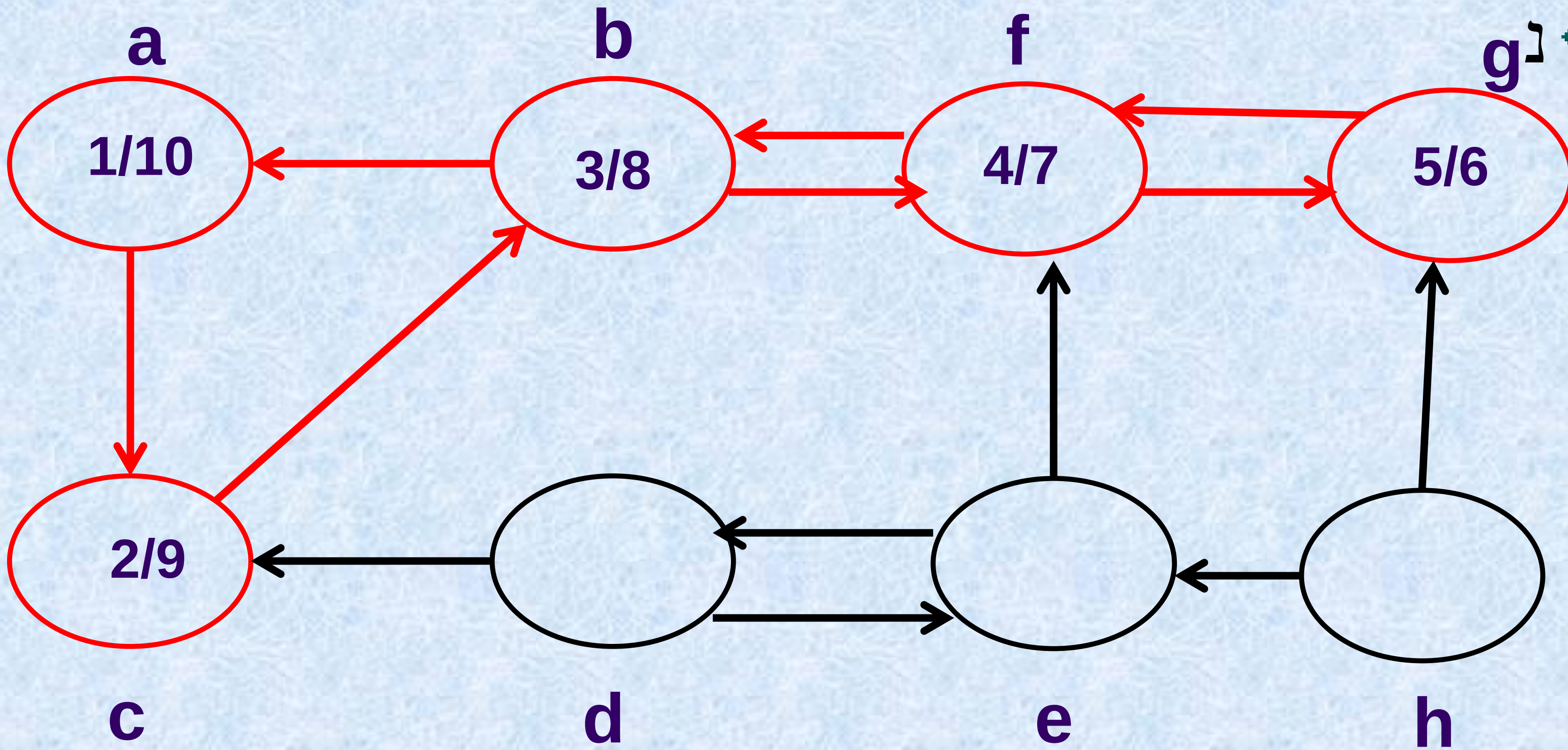
- קל לראות כי גרף העל הוא תמיד גרף מכוון **<u>ללא</u>** **<u>מעגלים</u>**. (מדוע ?).

דוגמת הרצה נוספת – הרשימה L שנקבל הינה: a, b, c, f, e, d, g , h



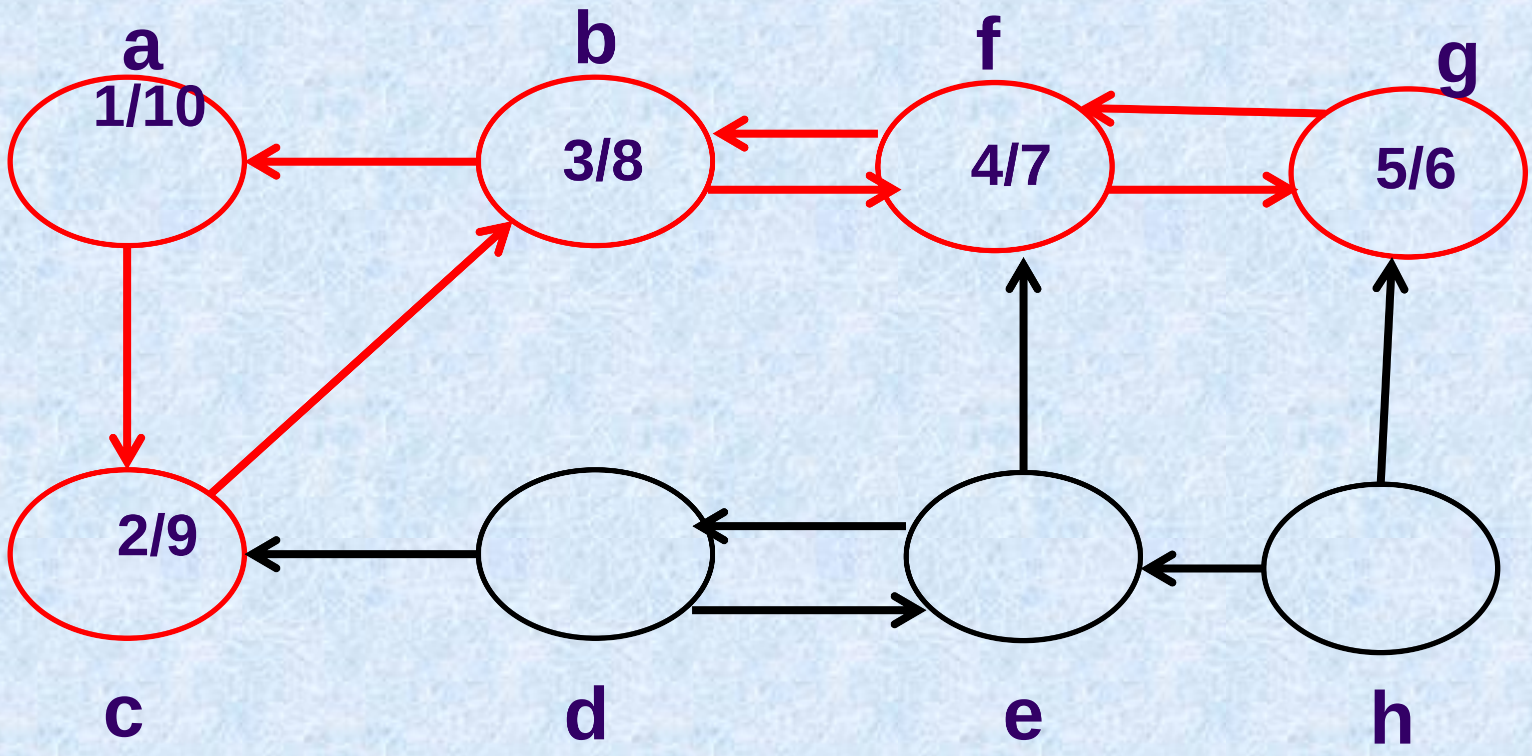
ע"פ הרשימה L שהינה a, b, c, f, e, d, g , h

נריץ DFS על גרף ההפוך מ-a ונקבל .



קדקודים שביקרנו בהם צבועים בצבע אדום :

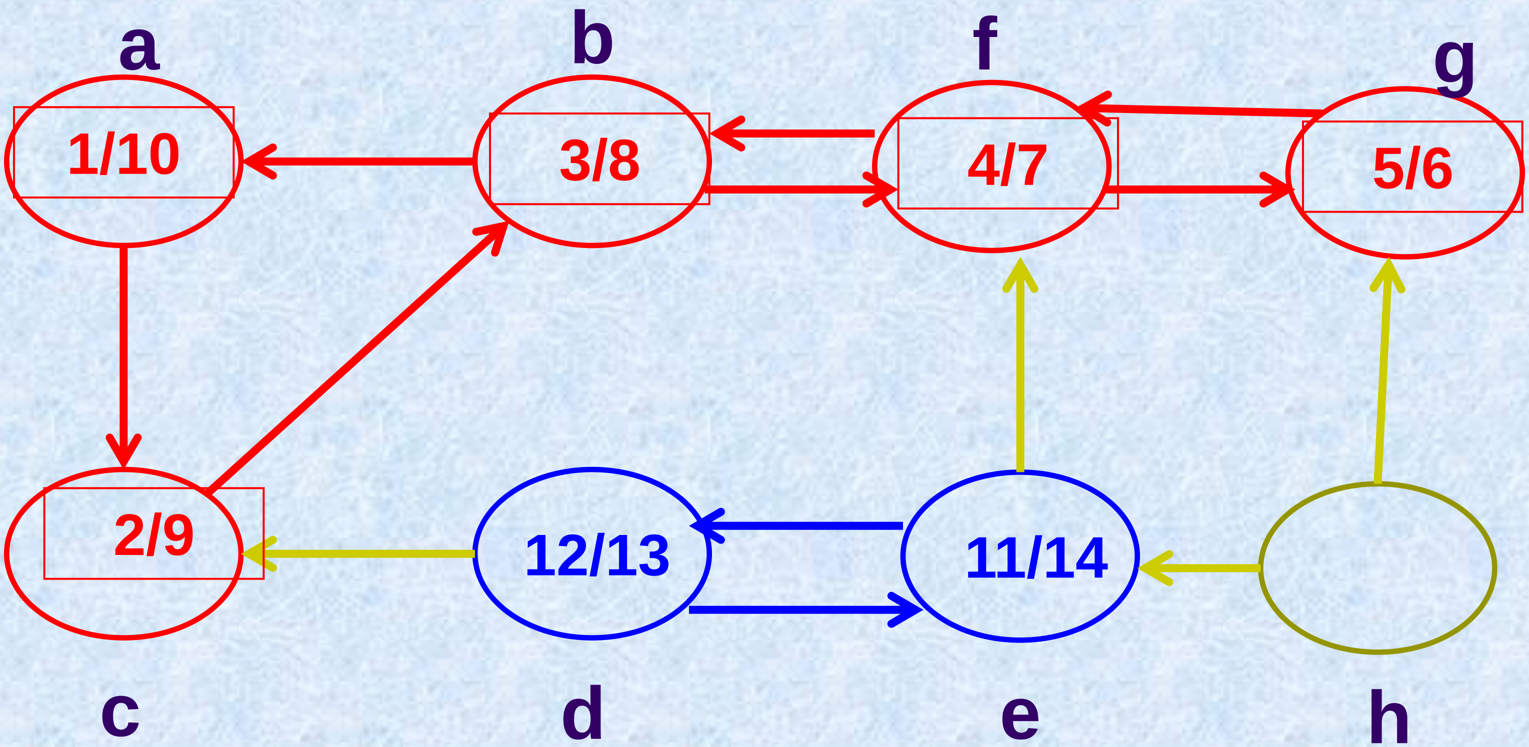
a, b, c, f, e, d, g , h



- עתה לפי הרשימה L יש להריץ את ה-DFS החל מ- e . נקבל:

קדקודים שביקרנו בהם זה עתה צבועים בצבע כחול:

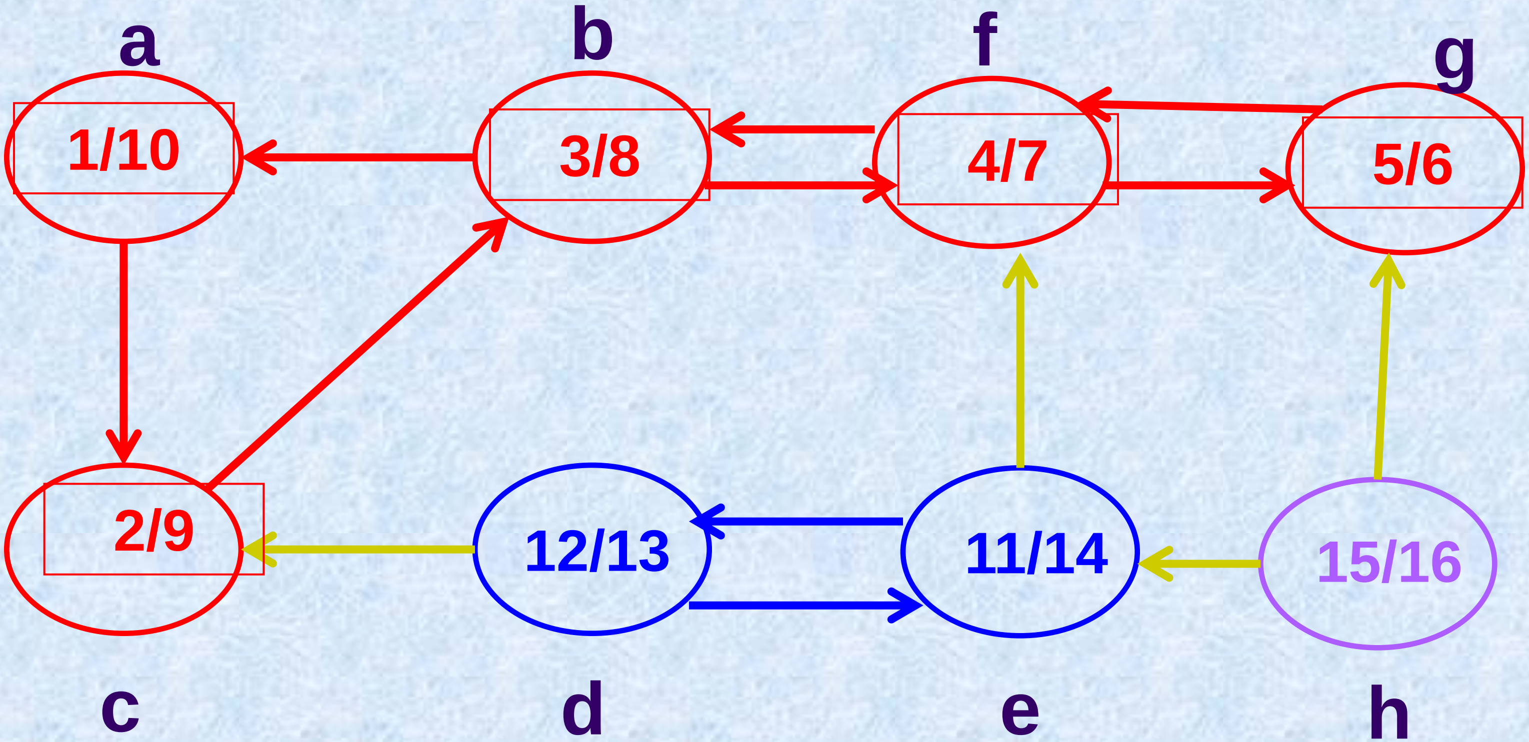
a, b, c, f, e, d, g , h



◈ עתה לפי הרשימה L יש להריץ את ה-DFS החל מ- h . נקבל:

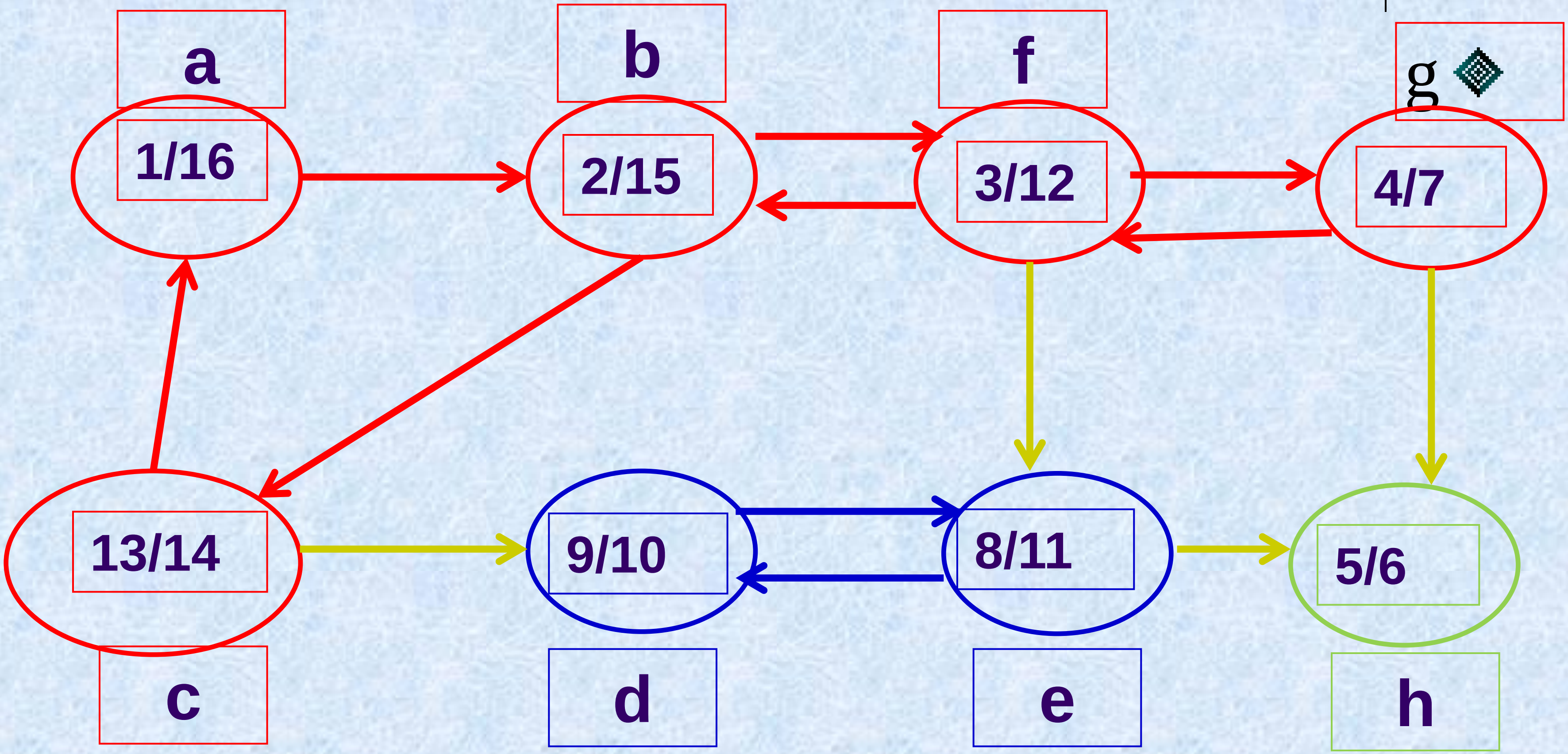
קדקודים שביקרנו בהם זה עתה צבועים בצבע סגול:

a, b, c, f, e, d, g , h



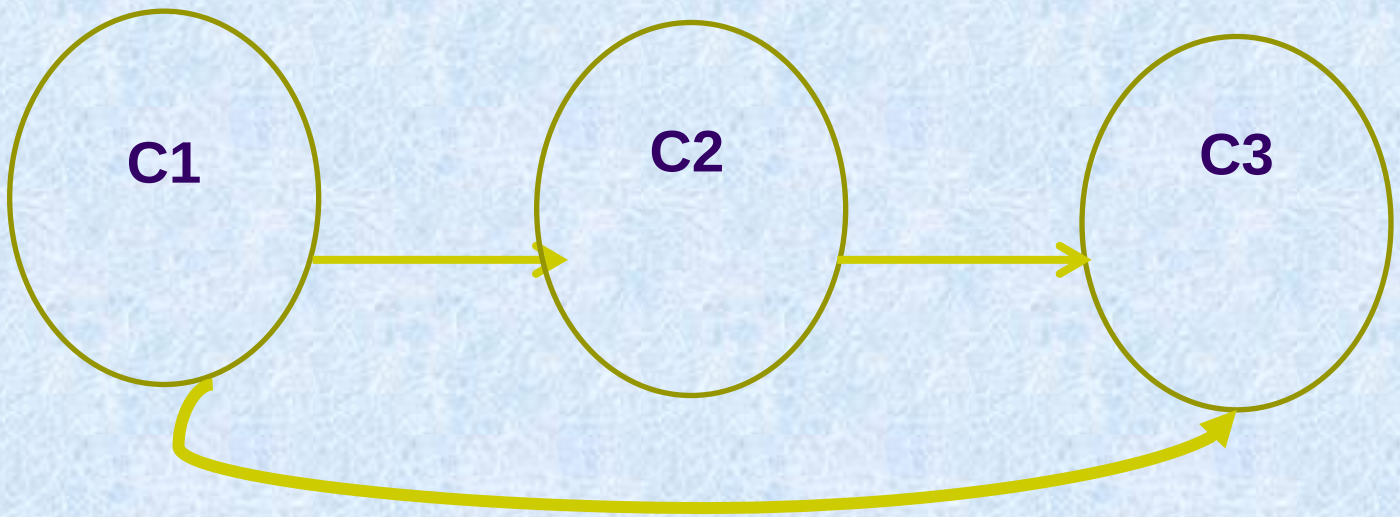
- סיימנו! כל אלה הצבועים באותו צבע משתייכים לאותו רק"ח.

בעבור הגרף המקורי קיבלנו 3 רק"ח-ים



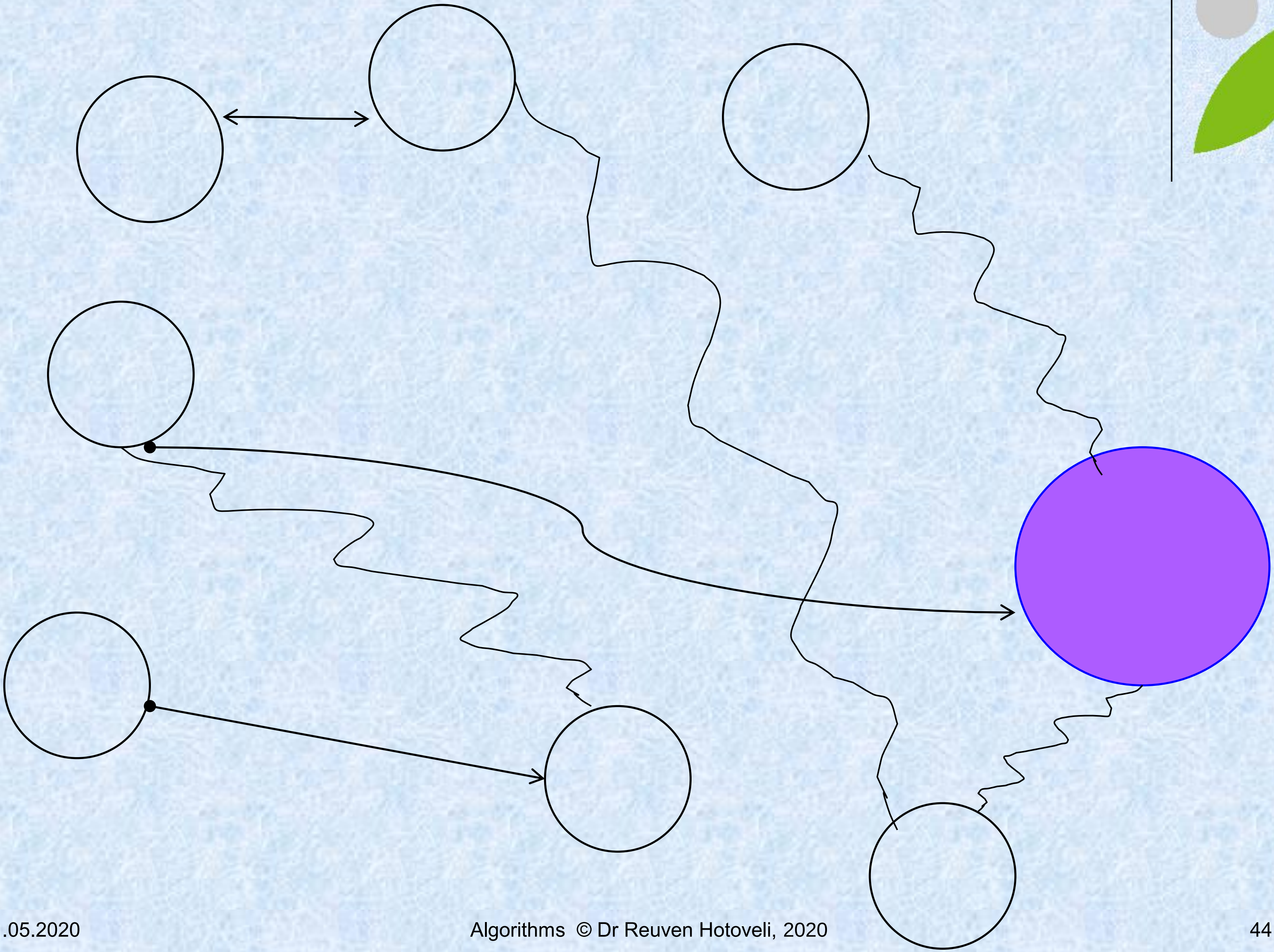
גרף העל של הגרף הוא:

- רואים שהגרף (גרף על) חסר מעגלים.



דוגמא1

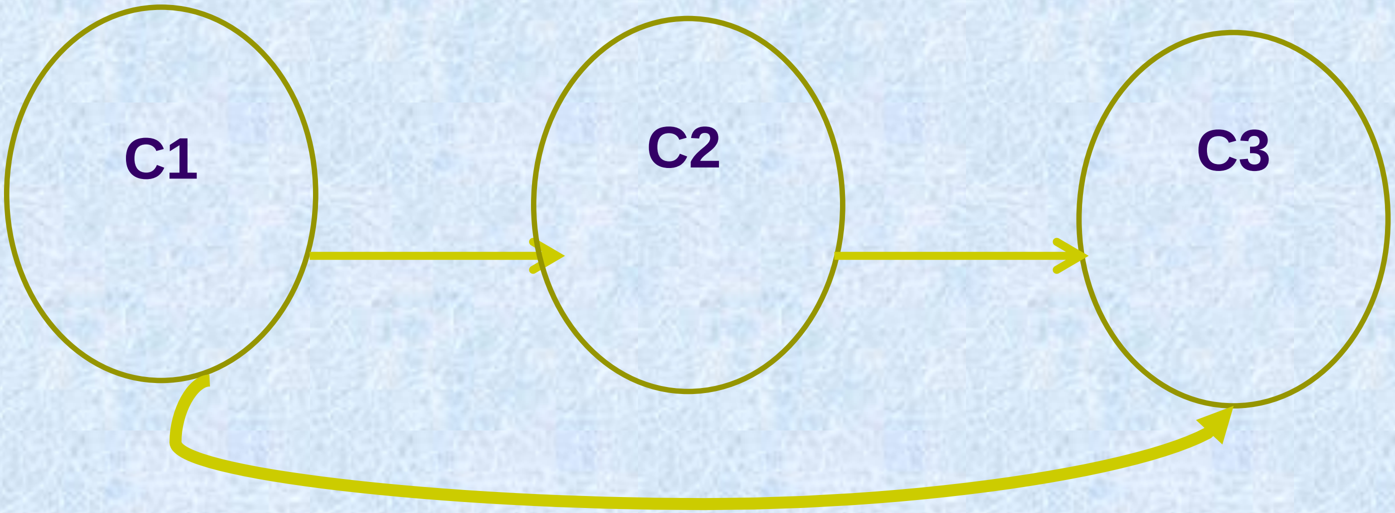
- דוגמא 1
- נתון גרף מכוון $G=(V,E)$ המיוצג על ידי רשימות שכנות.
- נתאר אלגוריתם המוצא את כל הקדקודים ב-G אם יש כאלה בכלל, אליהם אפשר להגיע במסלול מכוון מכל קדקודי הגרף



- פתרון
- 1- נמצא רכיבי קשירות חזקה .
- 2-עתה נבנה גרף על.
- 3- עתה נעבור על "גרף העל" ונספור את מספר הצמתים להם דרגת יציאה 0. (חייב להיות לפחות אחד כזה).

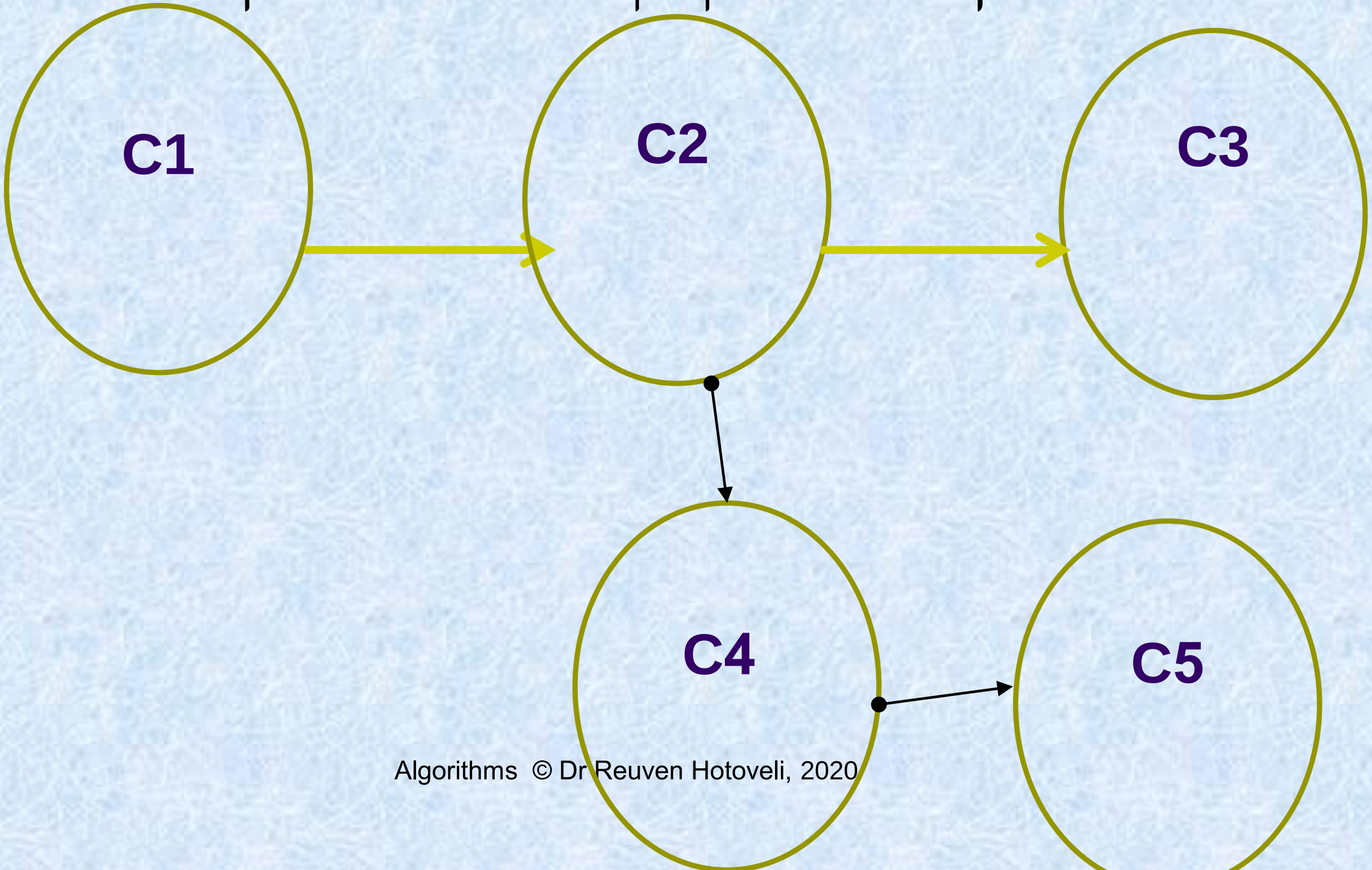
נניח שקיבלנו גרף על של גרף כלשהו הנראה כך:

- קל לראות שהקבוצה המבוקשת היא C3.



נניח שקיבלנו גרף על של גרף כלשהו אחר הנראה כך:

- קל לראות שהקבוצה המבוקשת לא קיימת כי מקדקודי C3 לא ניתן להגיע לקדקודי C5 ולהיפך.



מסקנה

- לכן המסקנה היא שבגרף על חייב להיות קדקוד אחד בלבד שדרגת היציאה שלו היא 0.
- אחרת, אם יש יותר מקדקוד אחד שדרגת היציאה שלו היא 0 אז לא קיימת הקבוצה המבוקשת.

המשך האלגוריתם

- 4- אם קיים קדקוד אחד בלבד ,בגרף העל ,שדרגת היציאה שלו 0 אז:
- כל הקדקודים שנמצאים ברכיב קשירות חזקה, המיוצגים באמצעות קדקוד זה בגרף העל, עונים לדרישת הבעיה.
- 5-אם קיים יותר מקדקוד אחד כזה אזי אין קדקודים כאלה בגרף שאפשר להגיע אליהם במסלול מכוון מכל קדקודי הגרף .

סיבוכיות זמן הריצה של האלגוריתם

- עתה ננתח את סיבוכיות זמן הריצה של האלגוריתם:
- צעד 1. דורש זמן $O(|E| + |V|)$.
- צעד 2. דורש זמן $O(|E|)$ מקסימום.
- הצעדים 4 ו 3. דורשים זמן $O(1)$ לכן ,
- סופית סיבוכיות זמן הריצה של האלגוריתם היא:
- $O(|E| + |V|)$.

דוגמא2

- דוגמא 2
- נתון גרף מכוון $G=(V,E)$ אשר מיוצג על ידי רשימות שכנות.
- נתאר אלגוריתם אשר מוצא קבוצה (אם קיימת קבוצה כזו)
- $V \supset S$ של קדקודים כך ש: $|S| > ½|V|$, כלומר מספר
- האיברים בקבוצה S גדול מחצי מספר הקדקודים בגרף,
- ולכל $X \in S$ ו $Y \in V$ קיים מסלול מכוון מ- X ל- Y .

- פתרון
- 1- נמצא רכיבי קשירות חזקה.
- 2- נבנה "גרף על" של רכיבי קשירות חזקה .
- 3- נספור לכל רכיב את דרגת הכניסה שלו.
- 4- אם יש יותר מרכיב אחד שדרגת הכניסה שלו 0 אזי עצור. ותן הודעה: <u>לא קיימת קבוצה כזו</u>.

- 5- אם קיים קדקוד אחד " בגרף על " המייצג רכיב אחד עם דרגת כניסה 0 אז הקבוצה המבוקשת S תכיל את כל הקדקודים שברכיב קשיר זה.
- 6- עתה נספור את מספר הקדקודים (K) בקבוצה S, ואחרי כן נבדוק :
- אם $K>|V|/2$ אז נודיע : קיימת קבוצה כזו.
- אם $K\leq|V|/2$ אז נודיע :לא קיימת קבוצה כזו.
- ***קל לראות*** כי סיבוכיות זמן הריצה של האלגוריתם הזה
- היא : $O(|E| + |V|)$.