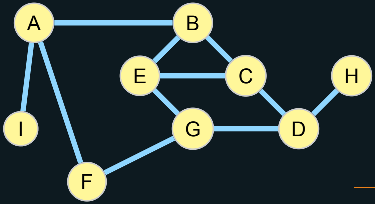



# יסודות האלגוריתמים והסיבוכיות

## תרגול 06 - גרפים ייצוג וחיפוש

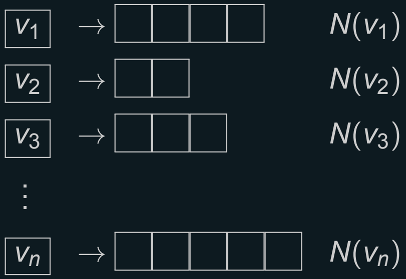
# גרפים

**גרף** $G = (V, E)$ מורכב מאוסף צמתים $V$ וקשתות $E$ המחברות בינהם.

# גרפים

**גרף** $G = (V, E)$ מורכב מאוסף צמתים $V$ וקשתות $E$ המחברות בינהם.

**רשימת שכנויות** היא מערך של רשימות שכנויות כאשר בתא $i$ במערך נשמור את כל השכנים של $v_i \in V$ ברשימה מקושרת $N(v_i)$.

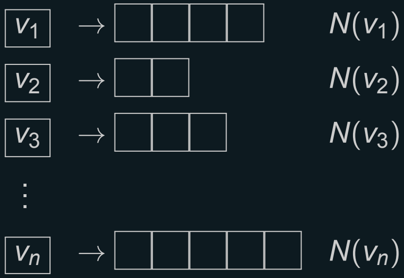


# גרפים

**גרף** $G=(V,E)$ מורכב מאוסף צמתים $V$ וקשתות $E$ המחוברות בינהם.

**רשימת שכנויות** היא מערך של רשימות שכנויות כאשר בתא $i$ במערך נשמור את כל השכנים של $v_i \in V$ ברשימה מקושרת $N(v_i)$.

**מטריצת שכנויות** היא מטריצה בינארית $A \in \{0,1\}^{n\times n}$ כאשר $A[i,j]=1$ אמ״ם $(i,j)\in E$.

# חיפוש בגרפים

**DFS**$(G = (V, E), v)$

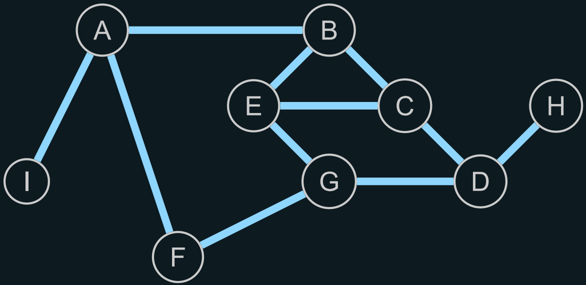
Mark $v$ as *visited*.
**for** $u \in N(v)$ **do**
  **if** $u$ is not *visited* **then**
    DFS$(G, u)$

**BFS**$(G = (V, E), v)$

Create empty list $L$.
Add $v$ to the tail of $L$.
**while** $L \neq \varnothing$ **do**
  Let $u$ be the head of $L$.
  Mark $u$ *visited* and remove it from $L$.
  **for** $w \in N(u)$ **do**
    **if** $w$ is *non-visited* **then**
      Add $w$ to the tail of $L$.

# תרגיל 1

נתון הגרף $G$ הבא:




(א) כתבו את גרף ה DFS המתקבל כאשר מתחילים בצומת $A$.

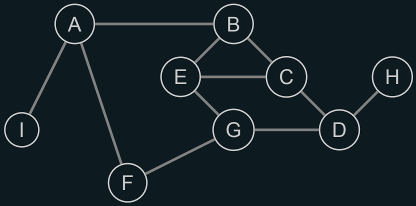
(ב) כתבו את גרף ה BFS המתקבל כאשר מתחילים בצומת $I$.

(ג) מהו סוג הגרפים? האם הם יחודיים ל $G$?

# תרגיל 1

$\mathrm{DFS}(G = (V, E), v)$

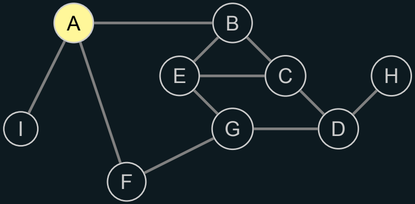
Mark $v$ as *visited*.
**for** $u \in N(v)$ **do**
    **if** $u$ is not *visited* **then**
        $\mathrm{DFS}(G, u)$



# תרגיל 1


**DFS**$(G = (V, E), v)$

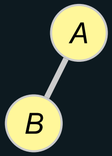
Mark $v$ as *visited*.
**for** $u \in N(v)$ **do**
    **if** $u$ is not *visited* **then**
        DFS$(G, u)$

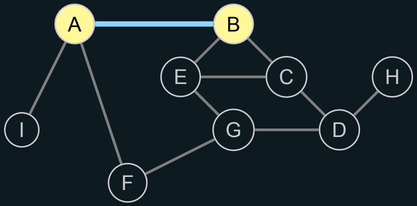


# תרגיל 1

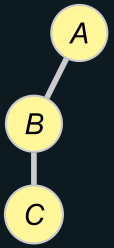
$\text{DFS}(G = (V, E), v)$

Mark $v$ as *visited*.
**for** $u \in N(v)$ **do**
    **if** $u$ is not *visited* **then**
        $\text{DFS}(G, u)$



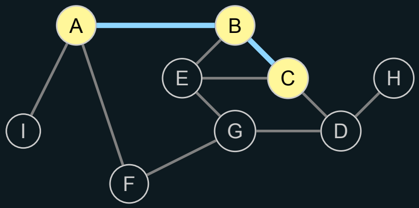



# תרגיל 1




DFS$(G = (V, E), v)$

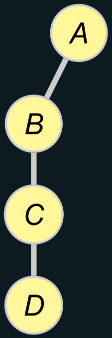
```
Mark v as visited.
for u ∈ N(v) do
    if u is not visited then
        DFS(G, u)
```

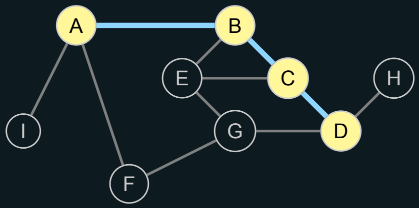


# תרגיל 1

DFS$(G=(V,E),v)$

Mark $v$ as *visited*.
**for** $u \in N(v)$ **do**
  **if** $u$ is not *visited* **then**
    DFS$(G,u)$

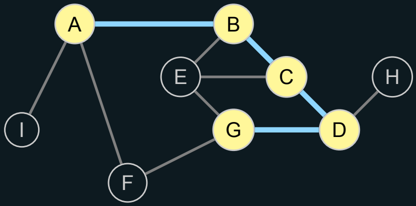


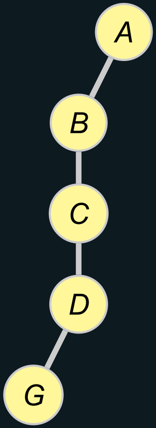



# תרגיל 1

$\text{DFS}(G = (V, E), v)$

Mark $v$ as *visited*.
**for** $u \in N(v)$ **do**
    **if** $u$ is not *visited* **then**
        $\text{DFS}(G, u)$

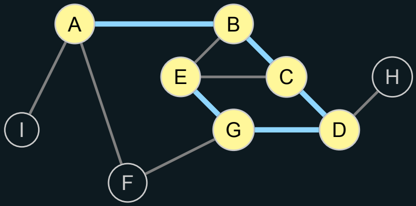


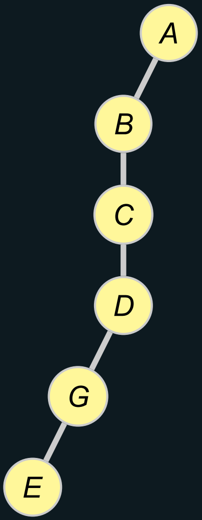



# תרגיל 1

$\text{DFS}(G = (V, E), v)$

Mark $v$ as *visited*.
**for** $u \in N(v)$ **do**
    **if** $u$ is not *visited* **then**
        $\text{DFS}(G, u)$

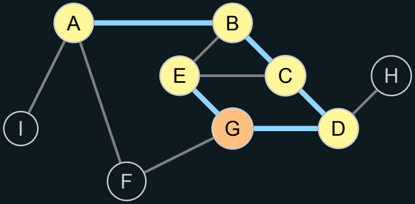


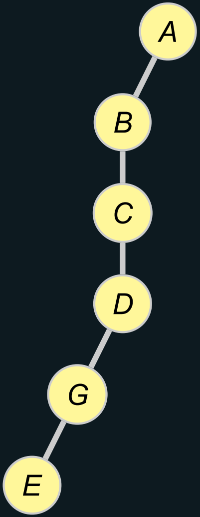



# תרגיל 1

**DFS**$(G = (V, E), v)$

1: Mark $v$ as *visited*.
2: **for** $u \in N(v)$ **do**
3: **if** $u$ is not *visited* **then**
4: DFS$(G, u)$

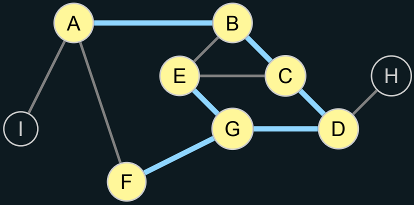


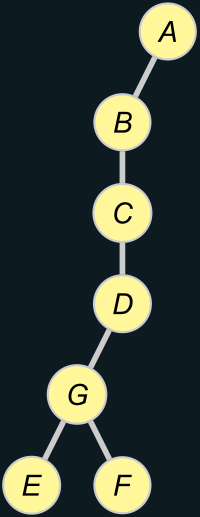



# תרגיל 1

DFS($G = (V, E), v$)

```
Mark v as visited.
for u ∈ N(v) do
    if u is not visited then
        DFS(G, u)
```

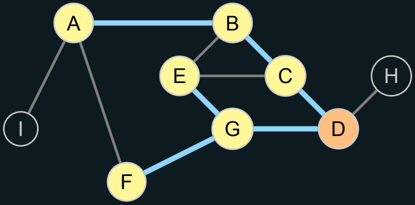


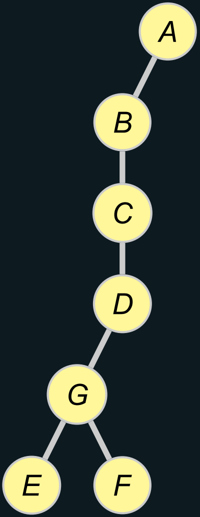



# תרגיל 1

$\text{DFS}(G = (V, E), v)$

Mark $v$ as *visited*.
**for** $u \in N(v)$ **do**
  **if** $u$ is not *visited* **then**
    $\text{DFS}(G, u)$

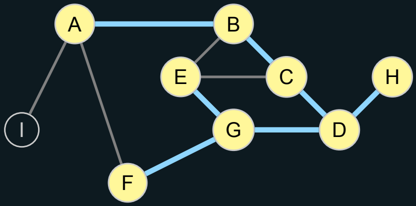


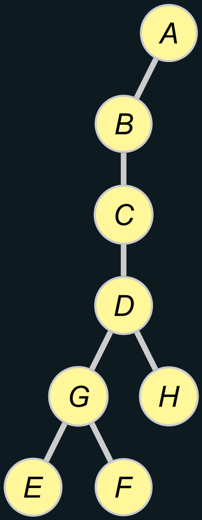



# תרגיל 1

**DFS**$(G = (V, E), v)$

Mark $v$ as *visited*.
**for** $u \in N(v)$ **do**
    **if** $u$ is not *visited* **then**
        DFS$(G, u)$

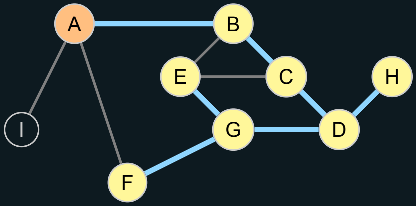


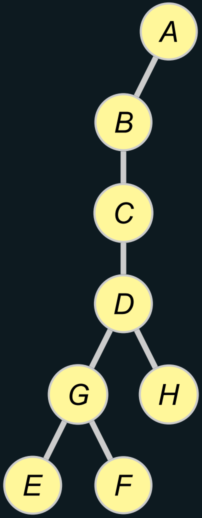



# תרגיל 1

$\text{DFS}(G = (V, E), v)$

Mark $v$ as *visited*.
**for** $u \in N(v)$ **do**
    **if** $u$ is not *visited* **then**
        $\text{DFS}(G, u)$

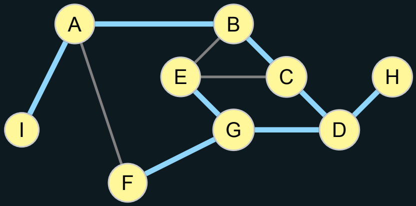


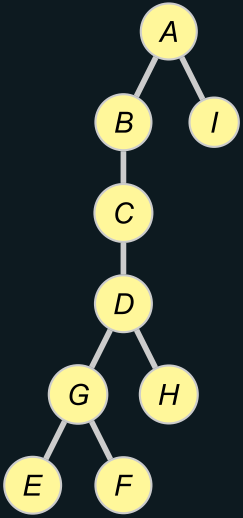



# תרגיל 1

$\text{DFS}(G = (V, E), v)$

```
Mark v as visited.
for u ∈ N(v) do
    if u is not visited then
        DFS(G, u)
```

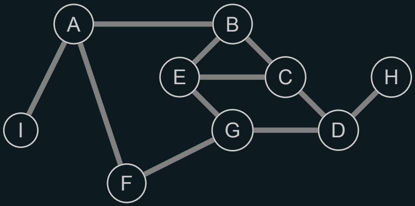


# תרגיל 1

**BFS**$(G = (V, E), v)$

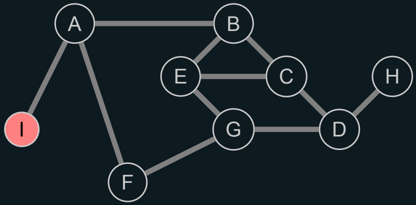
Create empty list $L$.
Add $v$ to the tail of $L$.
**while** $L \neq \emptyset$ **do**
    Let $u$ be the head of $L$.
    Mark $u$ *visited* and remove it from $L$.
    **for** $w \in N(u)$ **do**
        **if** $w$ is *non-visited* **then**
            Add $w$ to the tail of $L$.



# תרגיל 1

BFS($G = (V, E), v$)

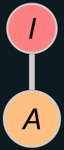
Create empty list $L$.
Add $v$ to the tail of $L$.
**while** $L \neq \varnothing$ **do**
    Let $u$ be the head of $L$.
    Mark $u$ *visited* and remove it from $L$.
    **for** $w \in N(u)$ **do**
        **if** $w$ is *non-visited* **then**
            Add $w$ to the tail of $L$.

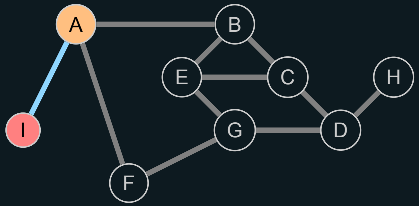



# תרגיל 1

**BFS**$(G = (V, E), v)$

Create empty list $L$.
Add $v$ to the tail of $L$.
**while** $L \neq \varnothing$ **do**
    Let $u$ be the head of $L$.
    Mark $u$ *visited* and remove it from $L$.
    **for** $w \in N(u)$ **do**
        **if** $w$ is *non-visited* **then**
            Add $w$ to the tail of $L$.

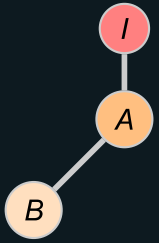


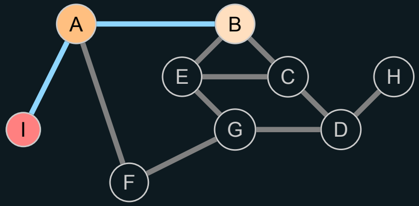



# תרגיל 1

BFS$(G = (V, E), v)$

Create empty list $L$.
Add $v$ to the tail of $L$.
**while** $L \neq \varnothing$ **do**
  Let $u$ be the head of $L$.
  Mark $u$ *visited* and remove it from $L$.
  **for** $w \in N(u)$ **do**
    **if** $w$ is *non-visited* **then**
      Add $w$ to the tail of $L$.



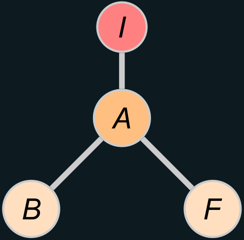



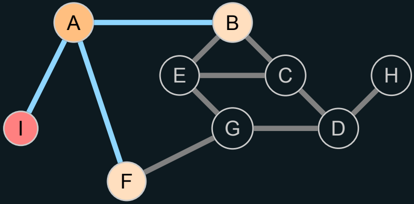
אוניברסיטת בן-גוריון בנגב
جامعة بن غوريون في النقب
Ben-Gurion University of the Negev

# תרגיל 1

**BFS**$(G = (V, E), v)$

```
Create empty list L.
Add v to the tail of L.
while L ≠ ∅ do
    Let u be the head of L.
    Mark u visited and remove it from L.
    for w ∈ N(u) do
        if w is non-visited then
            Add w to the tail of L.
```

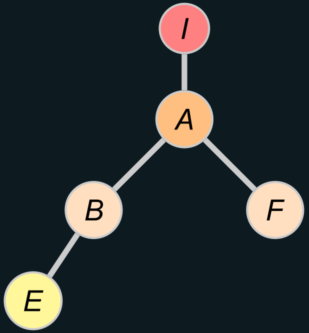


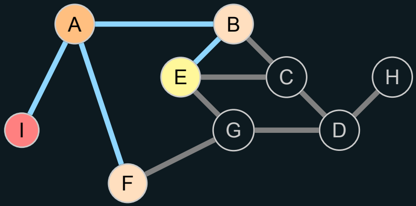



# תרגיל 1

BFS($G = (V, E), v$)

```
Create empty list L.
Add v to the tail of L.
while L ≠ ∅ do
    Let u be the head of L.
    Mark u visited and remove it from L.
    for w ∈ N(u) do
        if w is non-visited then
            Add w to the tail of L.
```

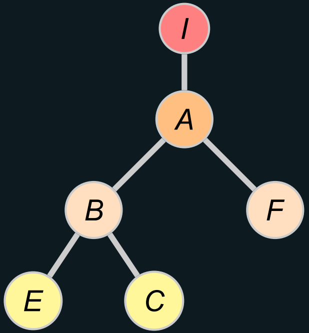


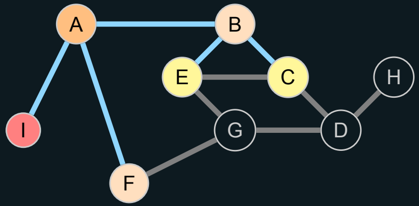



# תרגיל 1

BFS($G = (V, E), v$)

Create empty list $L$.
Add $v$ to the tail of $L$.
**while** $L \neq \emptyset$ **do**
    Let $u$ be the head of $L$.
    Mark $u$ *visited* and remove it from $L$.
    **for** $w \in N(u)$ **do**
        **if** $w$ is *non-visited* **then**
            Add $w$ to the tail of $L$.

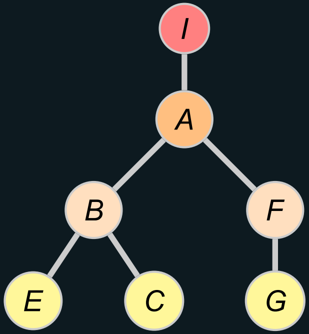


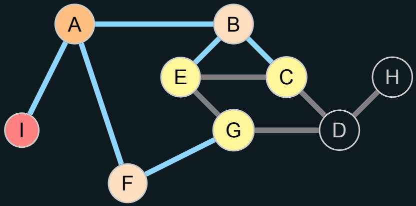



# תרגיל 1

**BFS**$(G=(V,E),v)$

```
Create empty list L.
Add v to the tail of L.
while L ≠ ∅ do
    Let u be the head of L.
    Mark u visited and remove it from L.
    for w ∈ N(u) do
        if w is non-visited then
            Add w to the tail of L.
```

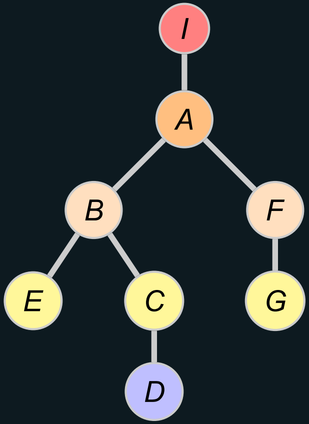


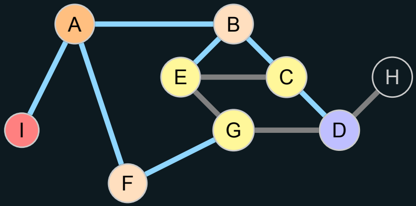



# תרגיל 1

BFS($G = (V, E), v$)

Create empty list $L$.
Add $v$ to the tail of $L$.
**while** $L \neq \varnothing$ **do**
    Let $u$ be the head of $L$.
    Mark $u$ *visited* and remove it from $L$.
    **for** $w \in N(u)$ **do**
        **if** $w$ is *non-visited* **then**
            Add $w$ to the tail of $L$.

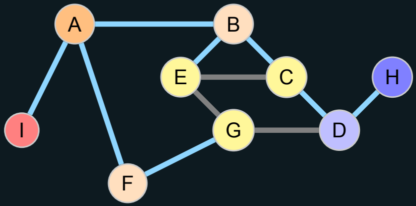


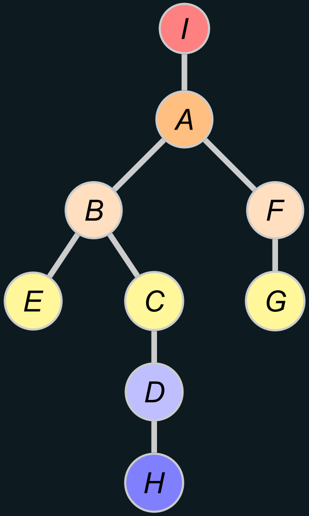



# תרגיל 1

**BFS**$(G = (V, E), v)$

Create empty list $L$.
Add $v$ to the tail of $L$.
**while** $L \neq \emptyset$ **do**
    Let $u$ be the head of $L$.
    Mark $u$ *visited* and remove it from $L$.
    **for** $w \in N(u)$ **do**
        **if** $w$ is *non-visited* **then**
            Add $w$ to the tail of $L$.



# תרגיל 1

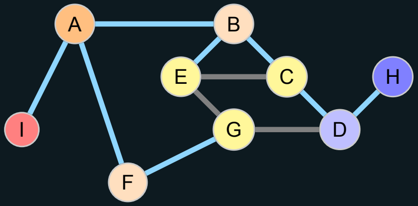
---

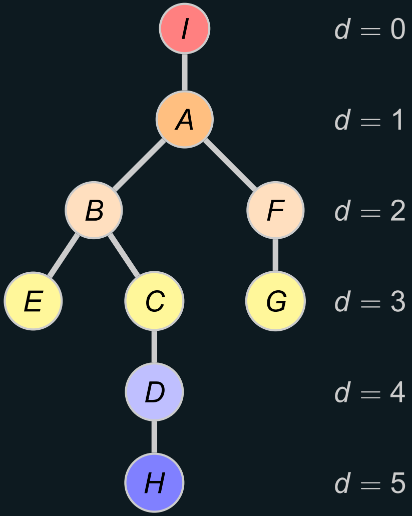
$\text{BFS}(G = (V, E), v)$

---

Create empty list $L$.
Add $v$ to the tail of $L$.
**while** $L \neq \emptyset$ **do**
  Let $u$ be the head of $L$.
  Mark $u$ *visited* and remove it from $L$.
  **for** $w \in N(u)$ **do**
    **if** $w$ is *non-visited* **then**
      Add $w$ to the tail of $L$.

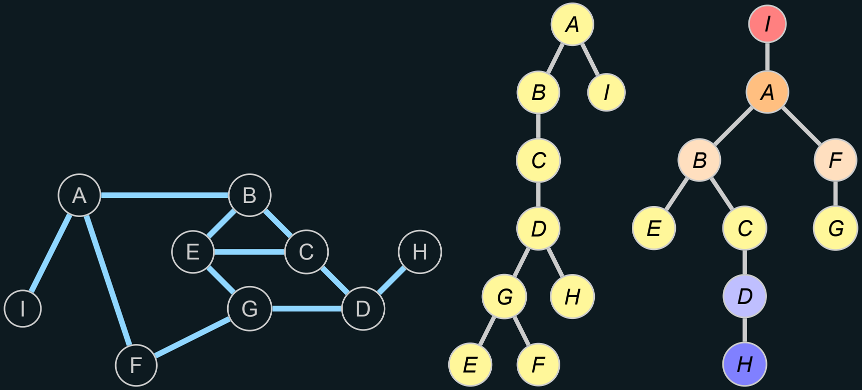
---



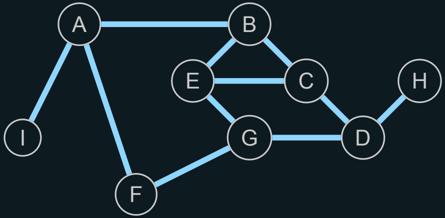
# תרגיל 1

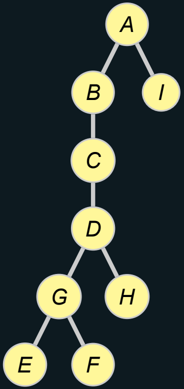
ג. מהו סוג הגרפים? האם הם יחודיים ל $G$?

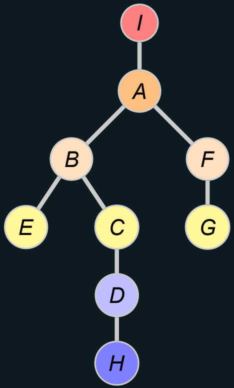


# תרגיל 1

ג. מהו סוג הגרפים? האם הם יחודיים ל $G$?



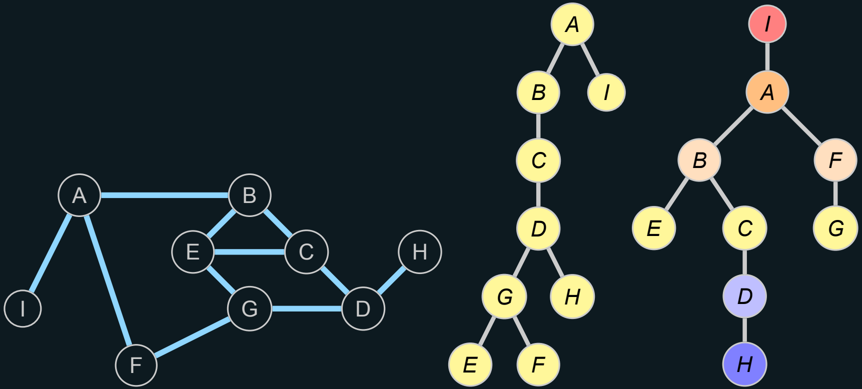




בשני הגרפים אין מעגלים, זוהי תכונה של עצים.

# תרגיל 1

ג.  מהו סוג הגרפים? האם הם יחודיים ל $G$?




בשני הגרפים אין מעגלים, זוהי תכונה של עצים.

שני הגרפים אינם ייחודיים ל $G$ ותלויים בסדר הצמתים שעולה בסריקות של BFS ו DFS.

# תרגיל 2

יש $n$ ערים ו $m$ כבישים. הניחו שכל דרך היא דו-כיוונית. הציעו אלגוריתם לעיבוד המידע (preprocessing) בזמן $\mathcal{O}(n+m)$ כך שלאחר מכן אפשר לענות בזמן $\mathcal{O}(1)$ על שאלה "האם ניתן להגיע מעיר $a$ לעיר $b$".

# תרגיל 2

יש $n$ ערים ו $m$ כבישים. הניחו שכל דרך היא דו-כיוונית. הציעו אלגוריתם לעיבוד המידע (preprocessing) בזמן $\mathcal{O}(n+m)$ כך שלאחר מכן אפשר לענות בזמן $\mathcal{O}(1)$ על שאלה "האם ניתן להגיע מעיר $a$ לעיר $b$".

נשתמש באלגוריתם לבדיקת קשירות על הגרף בו $V$ הערים ו $E$ הכבישים:

ConnectedComponents($G = (V, E)$)

```
Counter = 1.
while ∃s ∈ V do
    DFS(G, s)
    for v ∈ V do
        if v was visit by DFS then
            Component(v) = Counter
            Delete v from G.
    Counter = Counter +1.
```

# תרגיל 2

יש $n$ ערים ו $m$ כבישים. הניחו שכל דרך היא דו-כיוונית. הציעו אלגוריתם לעיבוד המידע (preprocessing) בזמן $\mathcal{O}(n+m)$ כך שלאחר מכן אפשר לענות בזמן $\mathcal{O}(1)$ על שאלה "האם ניתן להגיע מעיר $a$ לעיר $b$".

נשתמש באלגוריתם לבדיקת קשירות על הגרף בו $V$ הערים ו $E$ הכבישים:

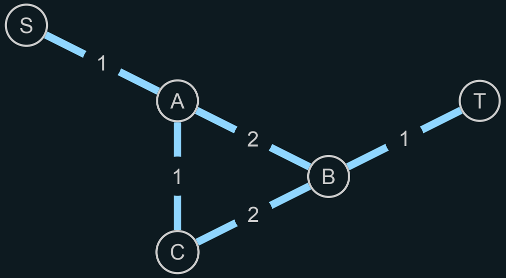
```
ConnectedComponents(G = (V, E))

Counter = 1.
while ∃s ∈ V do
    DFS(G, s)
    for v ∈ V do
        if v was visit by DFS then
            Component(v) = Counter
            Delete v from G.
    Counter = Counter + 1.
```

לאחר שלכל צומת מסומן רכיב הקשירות שלה, אם שתי ערים נמצאות באותו רכיב נוכל לוודא זאת בזמן קבוע.

# תרגיל 3

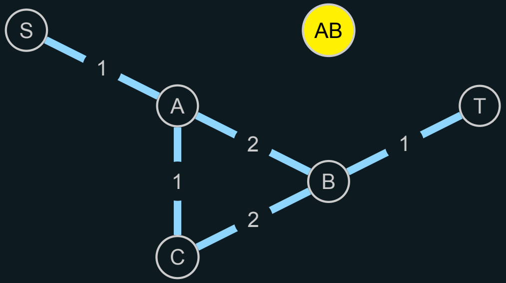
נתון גרף לא מכוון ממומש ברשימת שכנויות. לכל צלע יש משקל $w : E \to \{1, 2\}$. לדוגמא:




הציעו אלגוריתם שמוצא מסלול הכי קל מקודקוד $v$ לקודקוד $u$ בזמן $\mathcal{O}(|V| + |E|)$.

# תרגיל 3

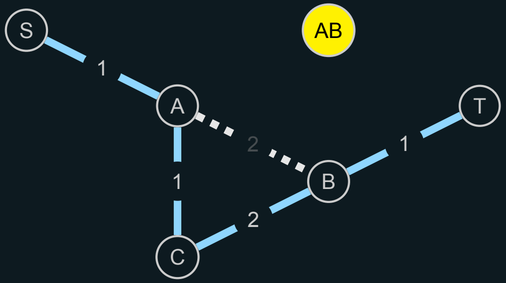
נתון גרף לא מכוון ממומש ברשימת שכנויות. לכל צלע יש משקל $w : E \to \{1, 2\}$. לדוגמא:




הציעו אלגוריתם שמוצא מסלול הכי קל מקודקוד $v$ לקודקוד $u$ בזמן $\mathcal{O}(|V| + |E|)$.

# תרגיל 3

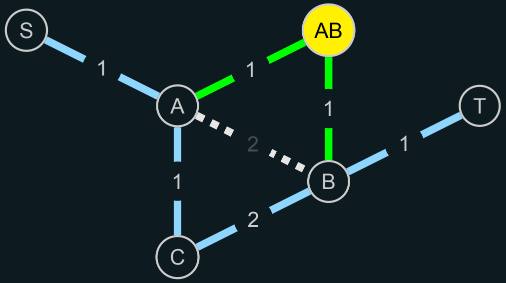
נתון גרף לא מכוון ממומש ברשימת שכנויות. לכל צלע יש משקל $w : E \to \{1, 2\}$. לדוגמא:




הציעו אלגוריתם שמוצא מסלול הכי קל מקודקוד $v$ לקודקוד $u$ בזמן $\mathcal{O}(|V| + |E|)$.

# תרגיל 3

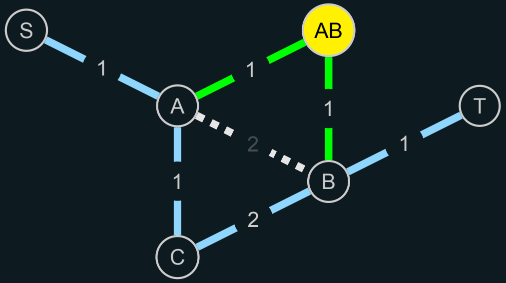
נתון גרף לא מכוון ממומש ברשימת שכנויות. לכל צלע יש משקל $w : E \to \{1, 2\}$. לדוגמא:




הציעו אלגוריתם שמוצא מסלול הכי קל מקודקוד $v$ לקודקוד $u$ בזמן $\mathcal{O}(|V| + |E|)$.

# תרגיל 3

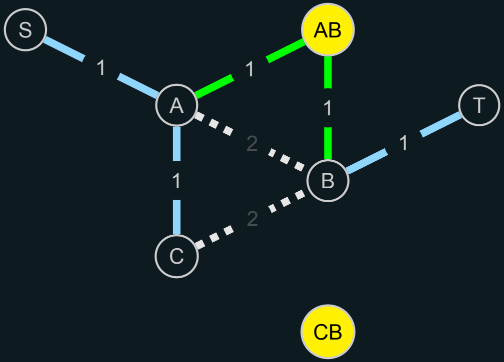
נתון גרף לא מכוון ממומש ברשימת שכנויות. לכל צלע יש משקל $w : E \to \{1, 2\}$. לדוגמא:




הציעו אלגוריתם שמוצא מסלול הכי קל מקודקוד $v$ לקודקוד $u$ בזמן $\mathcal{O}(|V| + |E|)$.

# תרגיל 3

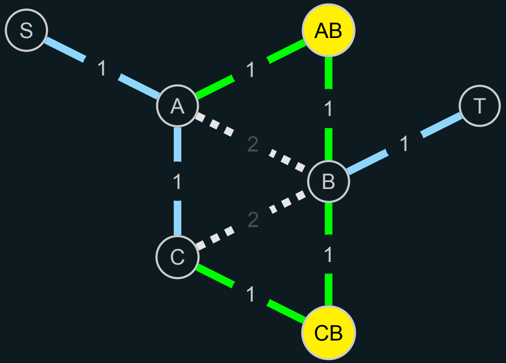
נתון גרף לא מכוון ממומש ברשימת שכנויות. לכל צלע יש משקל $w : E \to \{1, 2\}$. לדוגמא:




הציעו אלגוריתם שמוצא מסלול הכי קל מקודקוד $v$ לקודקוד $u$ בזמן $\mathcal{O}(|V| + |E|)$.

# תרגיל 3

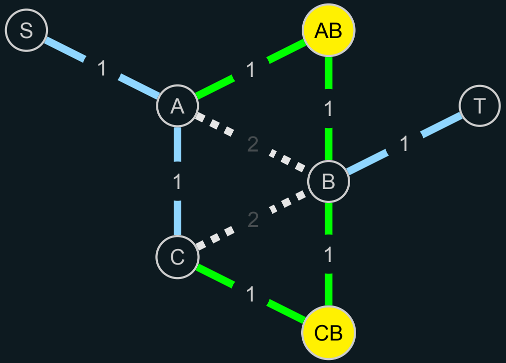
נתון גרף לא מכוון ממומש ברשימת שכנויות. לכל צלע יש משקל $w : E \to \{1, 2\}$. לדוגמא:




הציעו אלגוריתם שמוצא מסלול הכי קל מקודקוד $v$ לקודקוד $u$ בזמן $\mathcal{O}(|V| + |E|)$.

# תרגיל 3

נתון גרף לא מכוון ממומש ברשימת שכנויות. לכל צלע יש משקל $w : E \rightarrow \{1, 2\}$. לדוגמא:

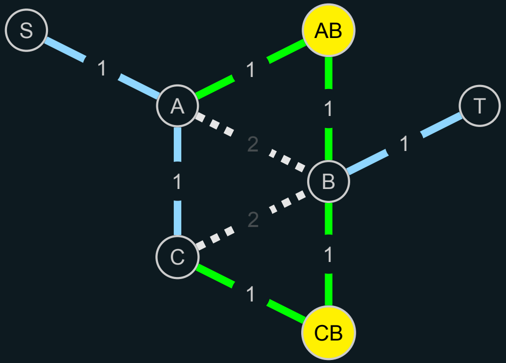



הציעו אלגוריתם שמוצא מסלול הכי קל מקודקוד $v$ לקודקוד $u$ בזמן $\mathcal{O}(|V| + |E|)$.

נעבור על קשתות הגרף ונחליף כל קשת במשקל 2 במסלול פשוט באורך 2, נוסיף אם כך לגרף $\mathcal{O}(m)$ צמתים ו $\mathcal{O}(m)$ קשתות.

# תרגיל 3

נתון גרף לא מכוון ממומש ברשימת שכנויות. לכל צלע יש משקל $w : E \to \{1, 2\}$. לדוגמא:




הציעו אלגוריתם שמוצא מסלול הכי קל מקודקוד $v$ לקודקוד $u$ בזמן $\mathcal{O}(|V| + |E|)$.

נעבור על קשתות הגרף ונחליף כל קשת במשקל 2 במסלול פשוט באורך 2, נוסיף אם כך לגרף $\mathcal{O}(m)$ צמתים ו $\mathcal{O}(m)$ קשתות.

גודל הגרף החדש יהיה $\mathcal{O}(n' + m') = \mathcal{O}(n + m + m) = \mathcal{O}(n + m)$ ולכן אלגוריתם BFS ימצא בו מסלולים קצרים בזמן $\mathcal{O}(m + n)$.

# מיון טופולוגי

**מיון טופולוגי** של גרף מכוון וקשיר $G = (V, E)$ הוא סידור לנארי של $V$ כך שאם קיימת קשת $(v_1, v_2)$ אז $v_1$ יהיה לפני $v_2$.

כל צאצא תמיד יבוא אחרי האב שלו.

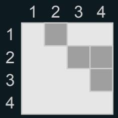
מיון טופולגי קיים רק בגרף ללא מעגלים.

למיון טופולוגי יש חשיבות להרבה משימות כגון ניהול פרויקטים (איזה משימה קודמת לאחרת), סדר תהליכים במחשב ועוד.

# תרגיל 4

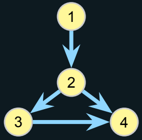
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

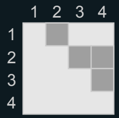
1. אתחול: $k = 0$
2. בנה מערך עזר בגודל $|V|$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות):
   א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר.
   ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$.
   ג. הכנס את $v$ למערך העזר
   ד. מחק את $v$ ואת הקשתות היוצאות ממנו.
4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון.



# תרגיל 4

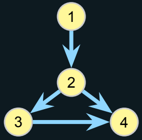
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

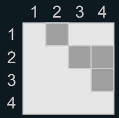
1. אתחול: $k = 0$ $\mathcal{O}(1)$
2. בנה מערך עזר בגודל $|V|$ $\mathcal{O}(|V|)$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות):
   - א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר.
   - ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$.
   - ג. הכנס את $v$ למערך העזר
   - ד. מחק את $v$ ואת הקשתות היוצאות ממנו.
4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון.



# תרגיל 4

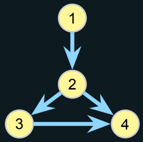
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

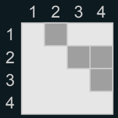
1. אתחול: $k = 0$ $\mathcal{O}(1)$
2. בנה מערך עזר בגודל $|V|$ $\mathcal{O}(|V|)$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות):
   א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר.
   כדי לבדוק שצומת הוא מקור נצטרך לבדוק במטריצת השכנויות את הטור שלו ולוודא שכולו מלא ב0.
   ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$.
   ג. הכנס את $v$ למערך העזר
   ד. מחק את $v$ ואת הקשתות היוצאות ממנו.
4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון.



# תרגיל 4

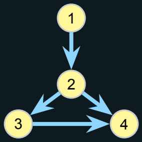
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

1. אתחול: $k = 0$ — $\mathcal{O}(1)$
2. בנה מערך עזר בגודל $|V|$ — $\mathcal{O}(|V|)$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות):
   - א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר. — $\mathcal{O}(|V|^2)$
     כדי לבדוק שצומת הוא מקור נצטרך לבדוק במטריצת השכנויות את הטור שלו ולוודא שכולו מלא ב0.
   - ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$.
   - ג. הכנס את $v$ למערך העזר
   - ד. מחק את $v$ ואת הקשתות היוצאות ממנו.

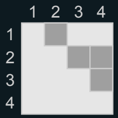



4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון.

# תרגיל 4

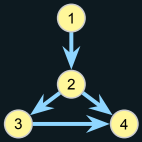
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

1. אתחול: $k = 0$ — $\mathcal{O}(1)$
2. בנה מערך עזר בגודל $|V|$ — $\mathcal{O}(|V|)$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות):
   - א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר. — $\mathcal{O}(|V|^2)$
     כדי לבדוק שצומת הוא מקור נצטרך לבדוק במטריצת השכנויות את הטור שלו ולוודא שכולו מלא ב0.
   - ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$. — $\mathcal{O}(1)$
   - ג. הכנס את $v$ למערך העזר
   - ד. מחק את $v$ ואת הקשתות היוצאות ממנו.

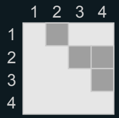



4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון.

# תרגיל 4

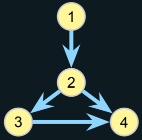
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

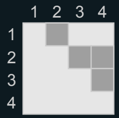
1. אתחול: $k = 0$ — $\mathcal{O}(1)$
2. בנה מערך עזר בגודל $|V|$ — $\mathcal{O}(|V|)$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות):
   א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר. — $\mathcal{O}(|V|^2)$
   כדי לבדוק שצומת הוא מקור נצטרך לבדוק במטריצת השכנויות את הטור שלו ולוודא שכולו מלא ב0.
   ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$. — $\mathcal{O}(1)$
   ג. הכנס את $v$ למערך העזר — $\mathcal{O}(1)$
   ד. מחק את $v$ ואת הקשתות היוצאות ממנו.
4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון.



# תרגיל 4

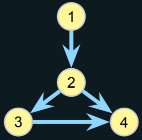
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

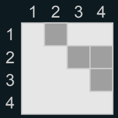
1. אתחול: $k = 0$ $\mathcal{O}(1)$
2. בנה מערך עזר בגודל $|V|$ $\mathcal{O}(|V|)$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות):
   - א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר. $\mathcal{O}(|V|^2)$
     כדי לבדוק שצומת הוא מקור נצטרך לבדוק במטריצת השכנויות את הטור שלו ולוודא שכולו מלא ב0.
   - ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$. $\mathcal{O}(1)$
   - ג. הכנס את $v$ למערך העזר $\mathcal{O}(1)$
   - ד. מחק את $v$ ואת הקשתות היוצאות ממנו.
     אפס את השורה של $v$.
4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון.



# תרגיל 4

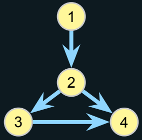
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

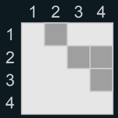
1. אתחול: $k = 0$ $\mathcal{O}(1)$
2. בנה מערך עזר בגודל $|V|$ $\mathcal{O}(|V|)$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות):
   א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר. $\mathcal{O}(|V|^2)$
   כדי לבדוק שצומת הוא מקור נצטרך לבדוק במטריצת השכנויות את הטור שלו ולוודא שכולו מלא ב0.
   ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$. $\mathcal{O}(1)$
   ג. הכנס את $v$ למערך העזר $\mathcal{O}(1)$
   ד. מחק את $v$ ואת הקשתות היוצאות ממנו. $\mathcal{O}(|V|)$
   אפס את השורה של $v$.
4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון.



# תרגיל 4

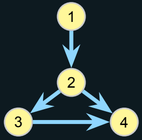
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

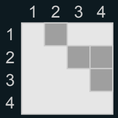
1. אתחול: $k = 0$ — $\mathcal{O}(1)$
2. בנה מערך עזר בגודל $|V|$ — $\mathcal{O}(|V|)$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות): — $\mathcal{O}(|V|)$
   - א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר. — $\mathcal{O}(|V|^2)$
     כדי לבדוק שצומת הוא מקור נצטרך לבדוק במטריצת השכנויות את הטור שלו ולוודא שכולו מלא ב0.
   - ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$. — $\mathcal{O}(1)$
   - ג. הכנס את $v$ למערך העזר — $\mathcal{O}(1)$
   - ד. מחק את $v$ ואת הקשתות היוצאות ממנו. — $\mathcal{O}(|V|)$
     אפס את השורה של $v$.
4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון.



# תרגיל 4

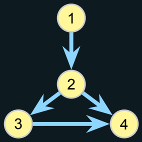
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

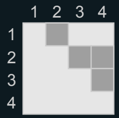
1. אתחול: $k = 0$ $\mathcal{O}(1)$
2. בנה מערך עזר בגודל $|V|$ $\mathcal{O}(|V|)$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות): $\mathcal{O}(|V|)$
   א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר. $\mathcal{O}(|V|^2)$
   כדי לבדוק שצומת הוא מקור נצטרך לבדוק במטריצת השכנויות את הטור שלו ולוודא שכולו מלא ב0.
   ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$. $\mathcal{O}(1)$
   ג. הכנס את $v$ למערך העזר $\mathcal{O}(1)$
   ד. מחק את $v$ ואת הקשתות היוצאות ממנו. $\mathcal{O}(|V|)$
   אפס את השורה של $v$.
4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון. $\mathcal{O}(1)$



# תרגיל 4

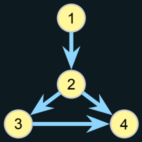
נתון אלגוריתם למציאת מיון טופולוגי בגרף.

הגרף ממומש במטריצת שכנויות. מה תהיה סבוכיות האלגוריתם במקרה זה?




**מיון טופולוגי:**

1. אתחול: $k = 0$ $\mathcal{O}(1)$
2. בנה מערך עזר בגודל $|V|$ $\mathcal{O}(|V|)$
3. כל עוד קיימים מקורות (צמתים ללא קשתות נכנסות): $\mathcal{O}(|V|)$
   - א. מצא מקור $v$, וודא שהוא לא נמצא במערך העזר. $\mathcal{O}(|V|^2)$
     כדי לבדוק שצומת הוא מקור נצטרך לבדוק במטריצת השכנויות את הטור שלו ולוודא שכולו מלא ב0.
   - ב. קדם את $k$ ב-1, תן ל-$v$ מספר $k$. $\mathcal{O}(1)$
   - ג. הכנס את $v$ למערך העזר $\mathcal{O}(1)$
   - ד. מחק את $v$ ואת הקשתות היוצאות ממנו. $\mathcal{O}(|V|)$
     אפס את השורה של $v$.
4. אם $k = |V|$ אז קיים סדר טופולוגי, אחרת בגרף יש מעגל מכוון. $\mathcal{O}(1)$




זמן הריצה הכולל הוא $\mathcal{O}(|V|^3)$