עצים
Trees

מה נלמד היום?

- עץ - תכונות והגדרות
- אלגוריתמי סריקה בעץ
- ייצוגים שונים של העץ
- שימוש נכון באלגוריתמי הסריקה השונים

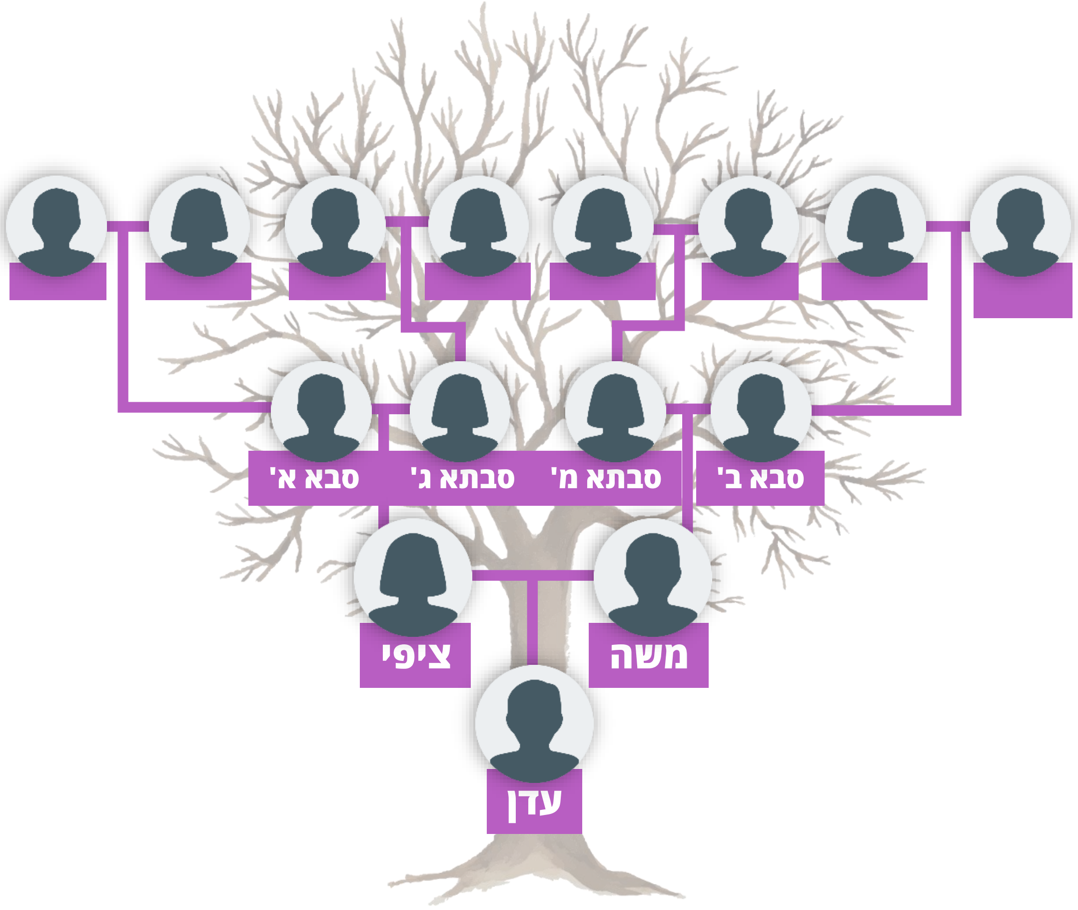
עדן קיבלה שיעורי בית

במסגרת עבודת שורשים, המורה ביקשה מעדן לספר על המשפחה שלה

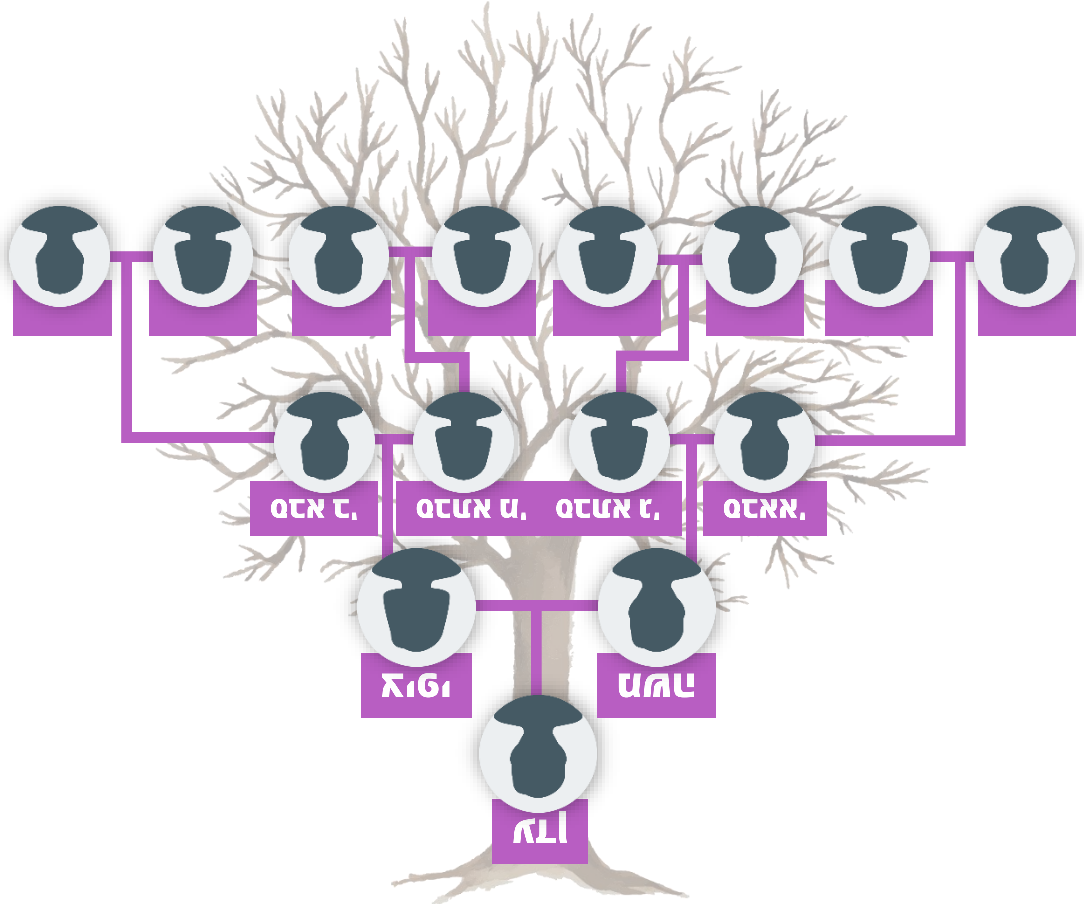
מטרה

- להציג מידע על בני משפחה
- להציג קשרים בין בני משפחה

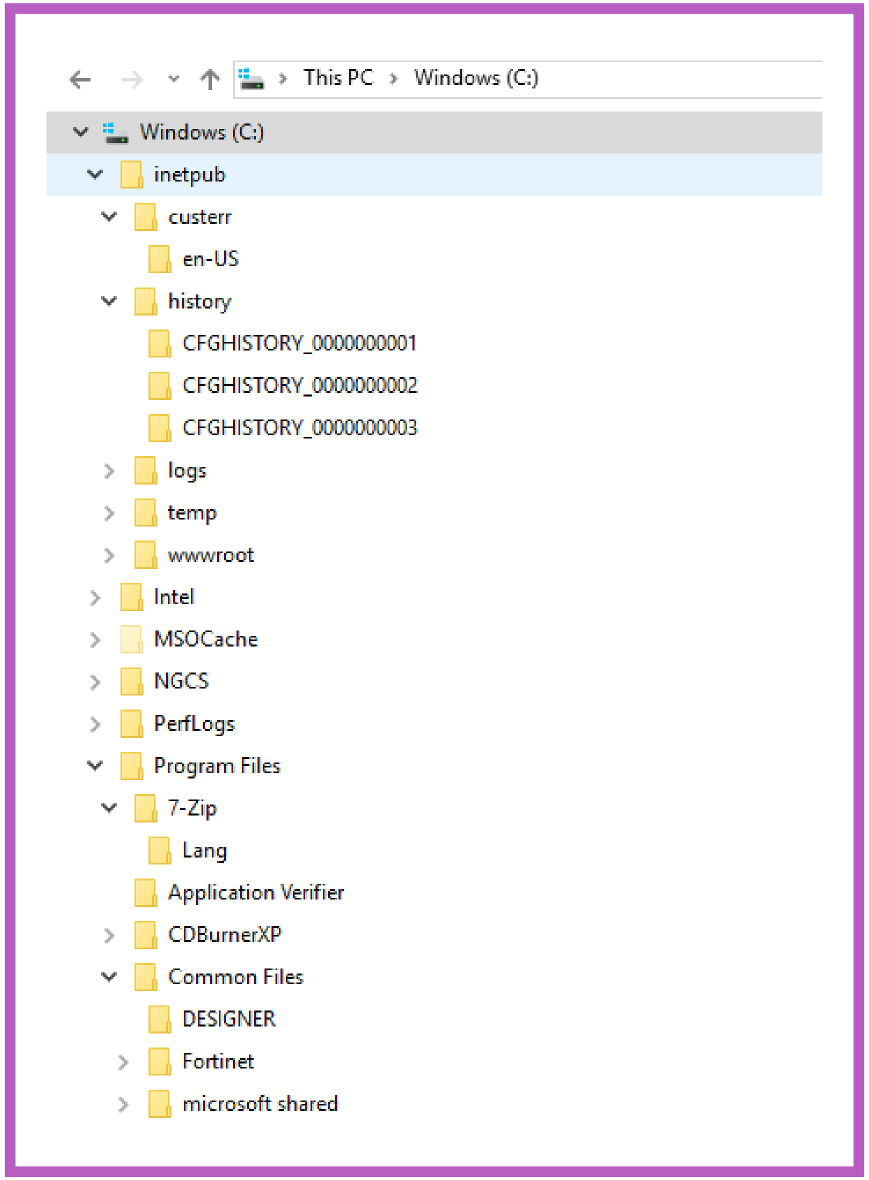
פתרון: אילן יוחסין



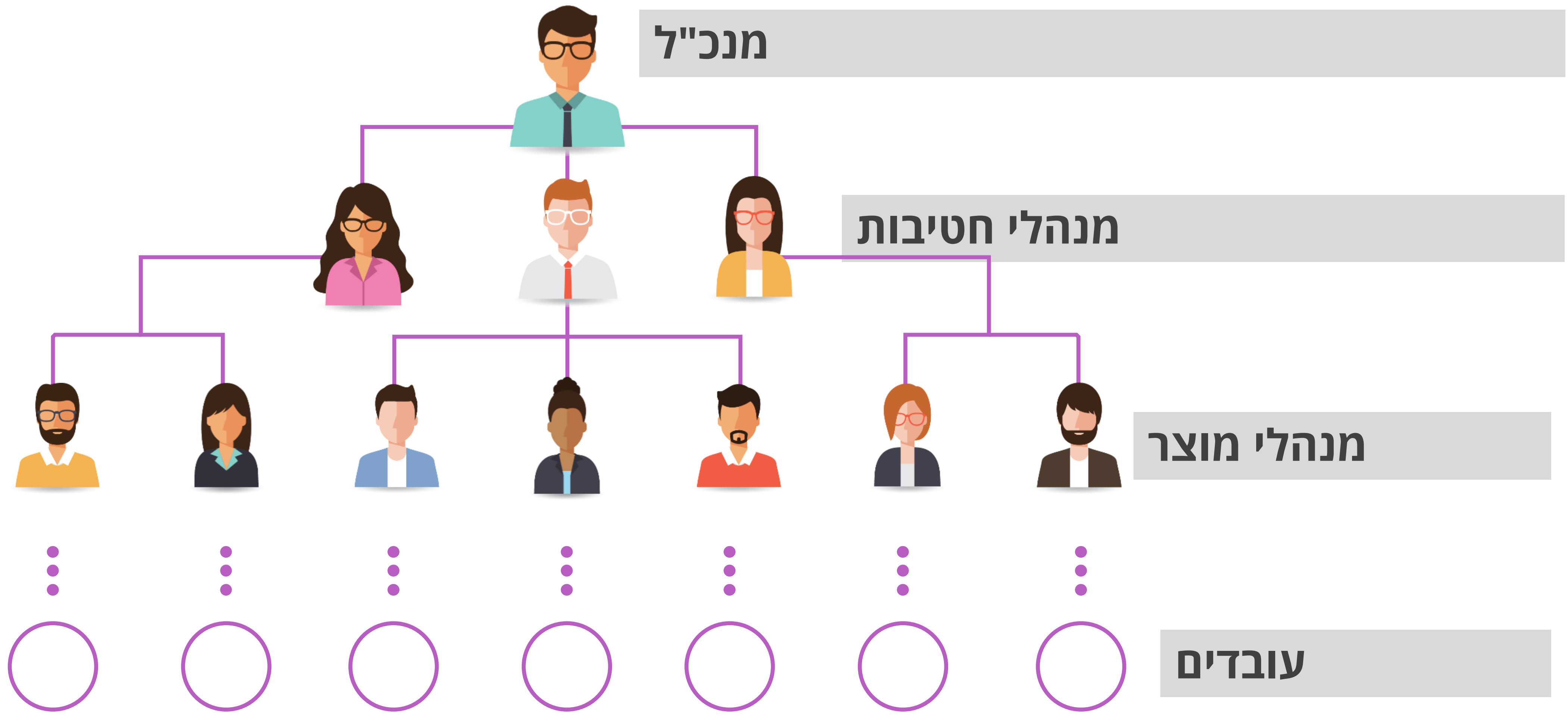
אילן יוחסין



מערכת קבצים כעץ במחשב

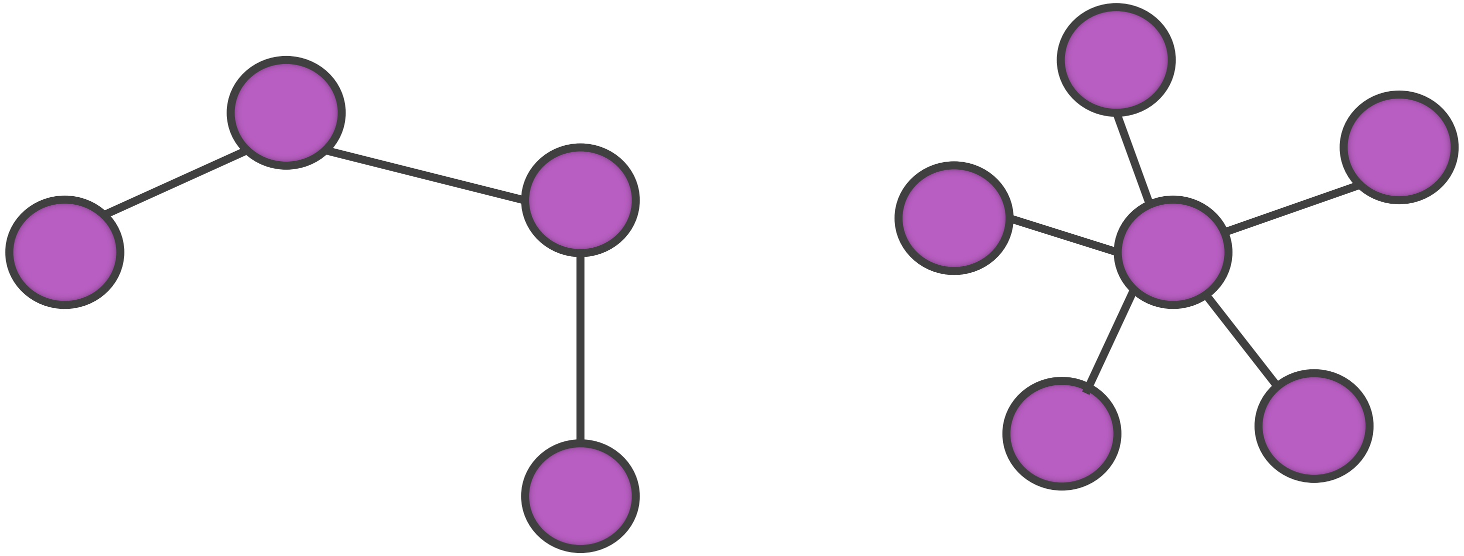


מבנה ארגוני של חברה כעץ



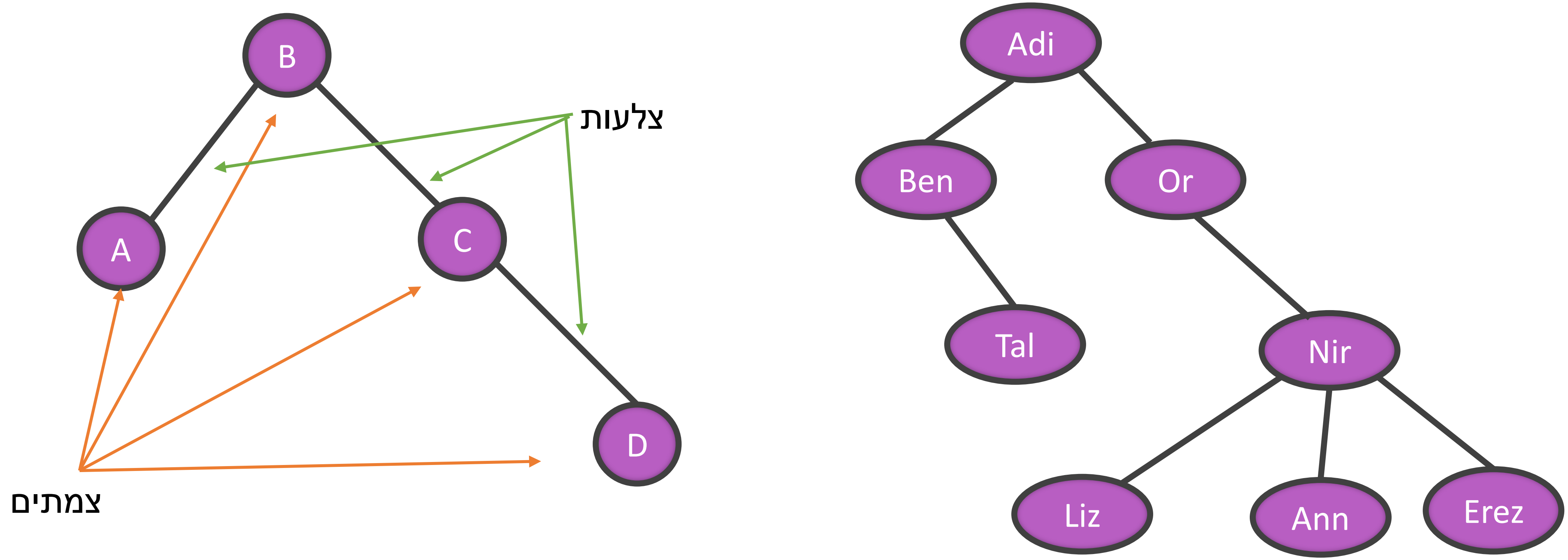
עצים

- עץ הוא גרף קשיר ללא מעגלים.



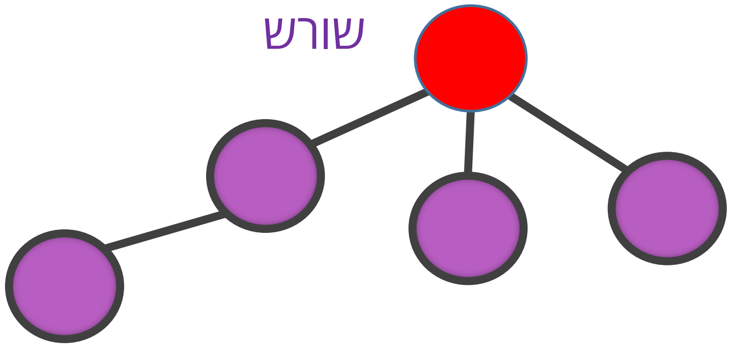
עצים

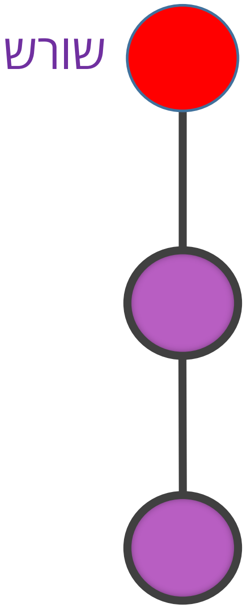
- עץ מורכב מצמתים וצלעות שמחברות ביניהם.

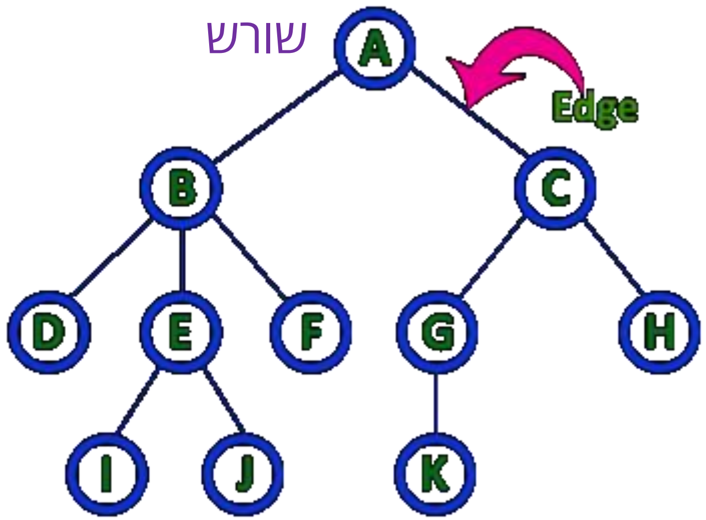


עץ מושרש

- עץ מושרש הוא עץ שבו אחד הצמתים נבחר להיות השורש
- נהוג לצייר את העץ עם השורש למעלה

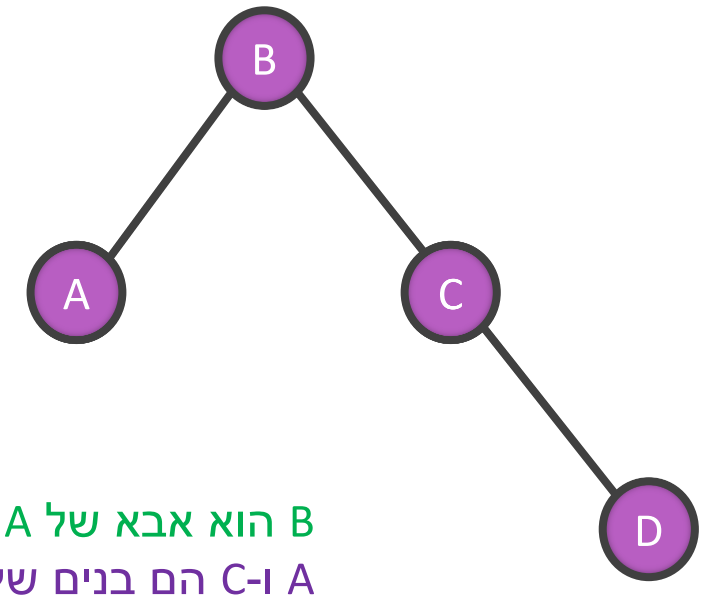




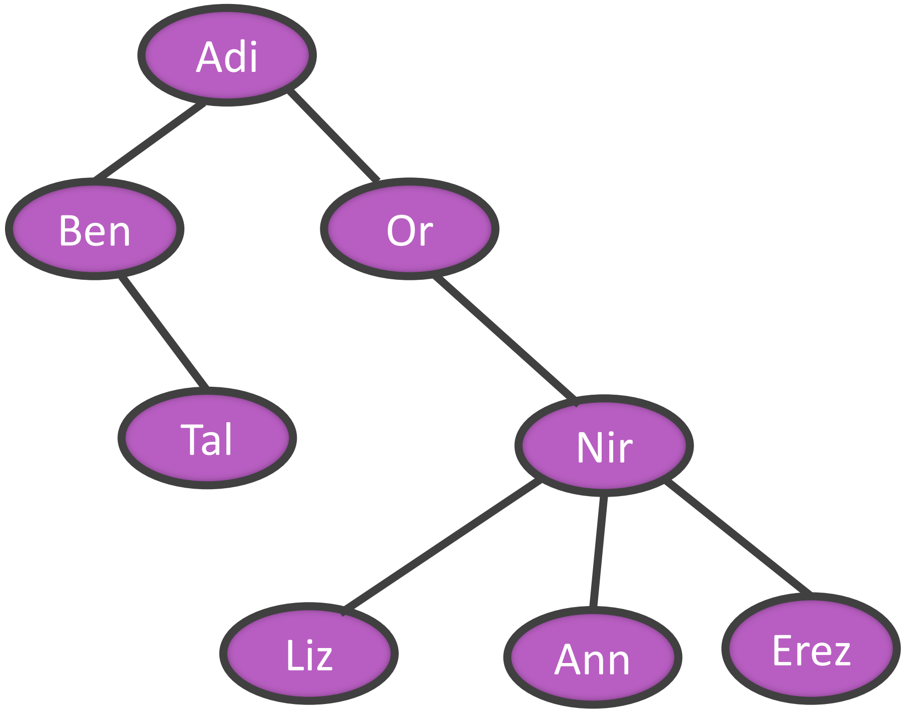


אב ובן

- **האב** של צומת בעץ הוא צומת שמעליו ומחובר אליו בצלע
- **הבן** של צומת בעץ הוא צומת שמתחתיו ומחובר אליו בצלע
- לכל צומת יש לכל היותר אב אחד, אבל יכולים להיות מספר בנים

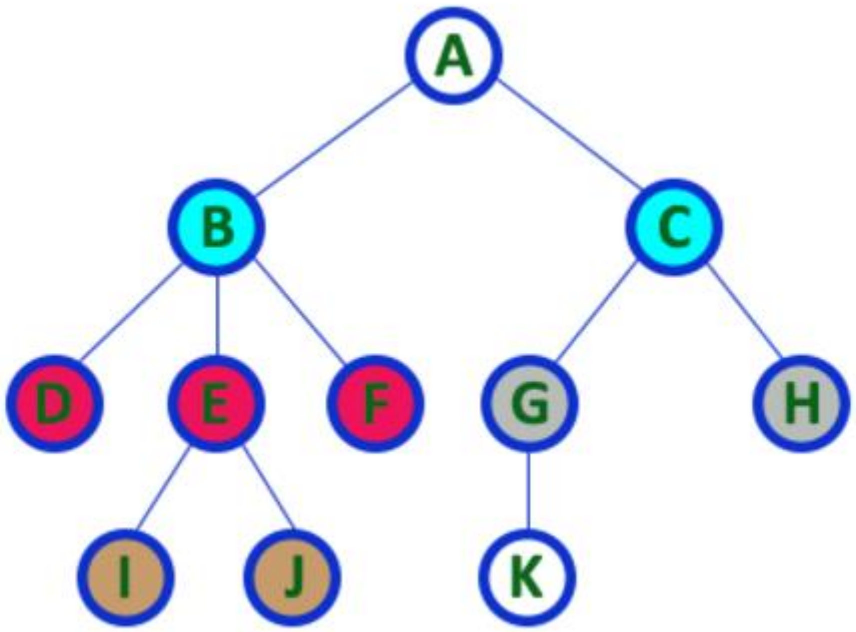


B הוא אבא של A ו-C
A ו-C הם בנים של B



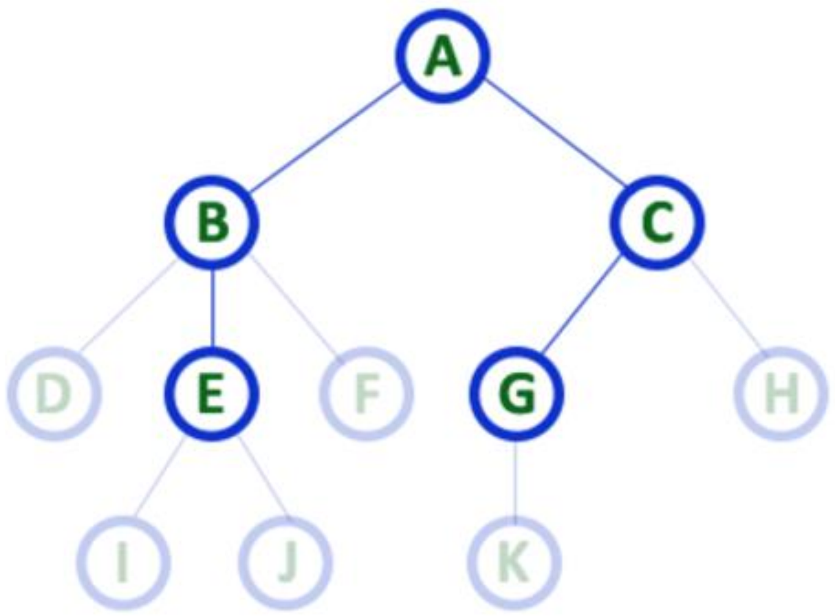
אחים (siblings)

• אם לשני צמתים יש אותו אב, הם נקראים **אחים (siblings)**



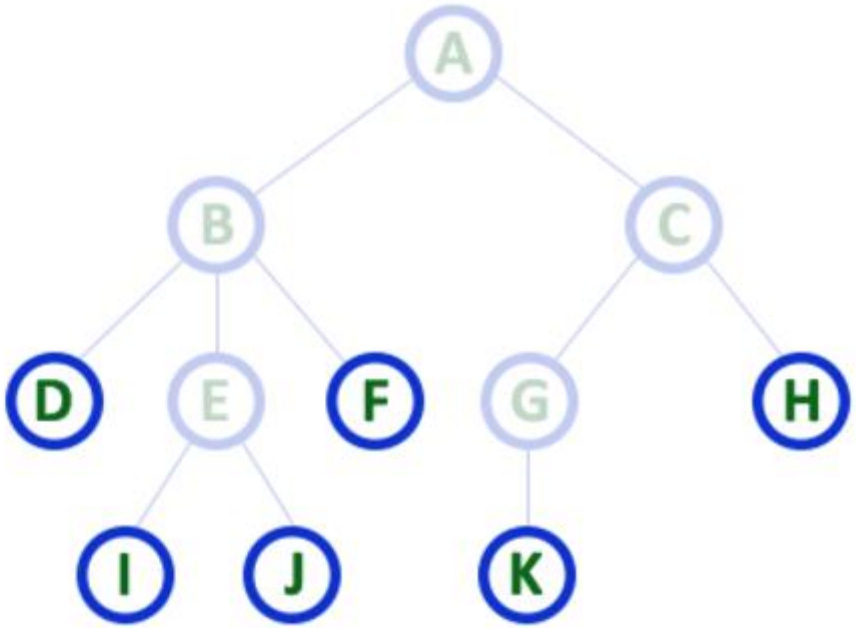
צומת פנימי (internal node)

- **<u>צומת פנימי (internal node)</u>** הוא צומת עם לפחות בן אחד



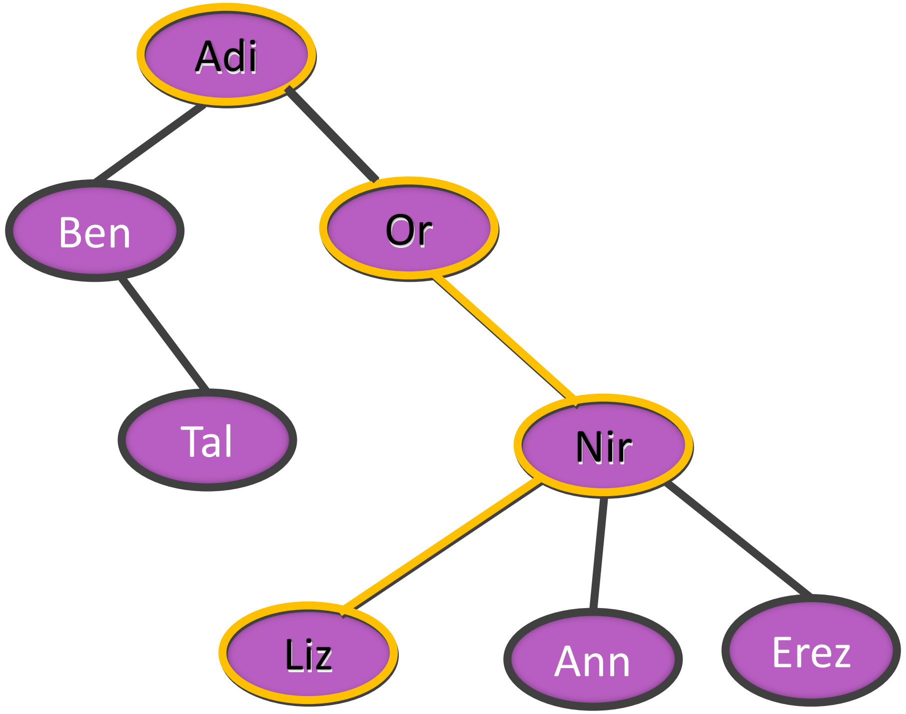
עלה ודרגה של צומת

- **<u>עלה (leaf)</u>** הוא צומת בלי בנים
- **<u>דרגה (degree)</u>** של צומת היא מספר הבנים שלו
  - סימון: degree(B)=3

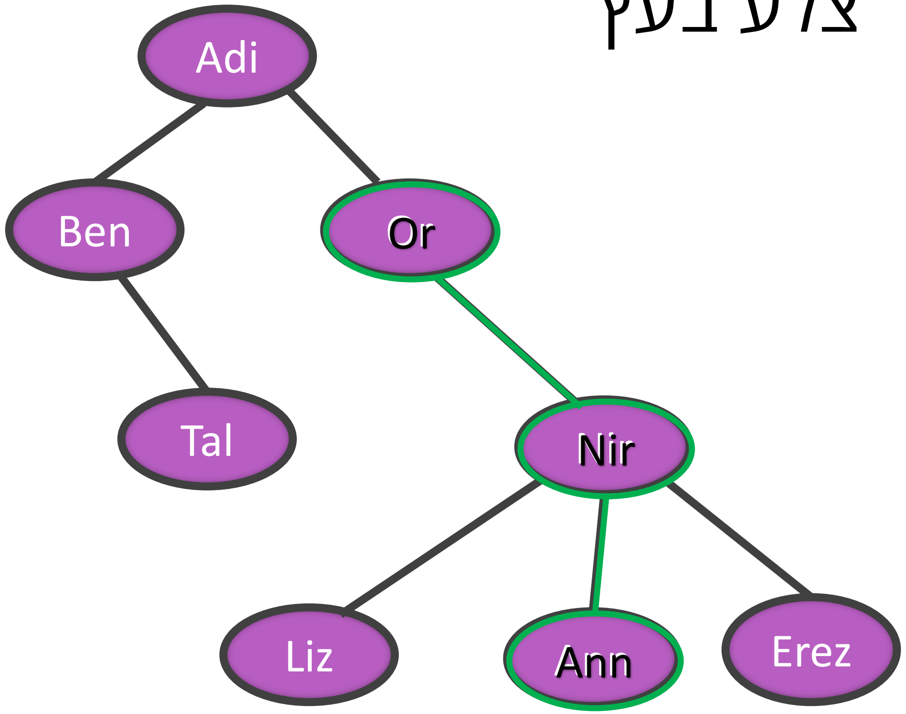


מסלול (Path)

- מסלול הוא סדרה של צמתים, כך שבין כל שני צמתים סמוכים בסדרה יש צלע בעץ



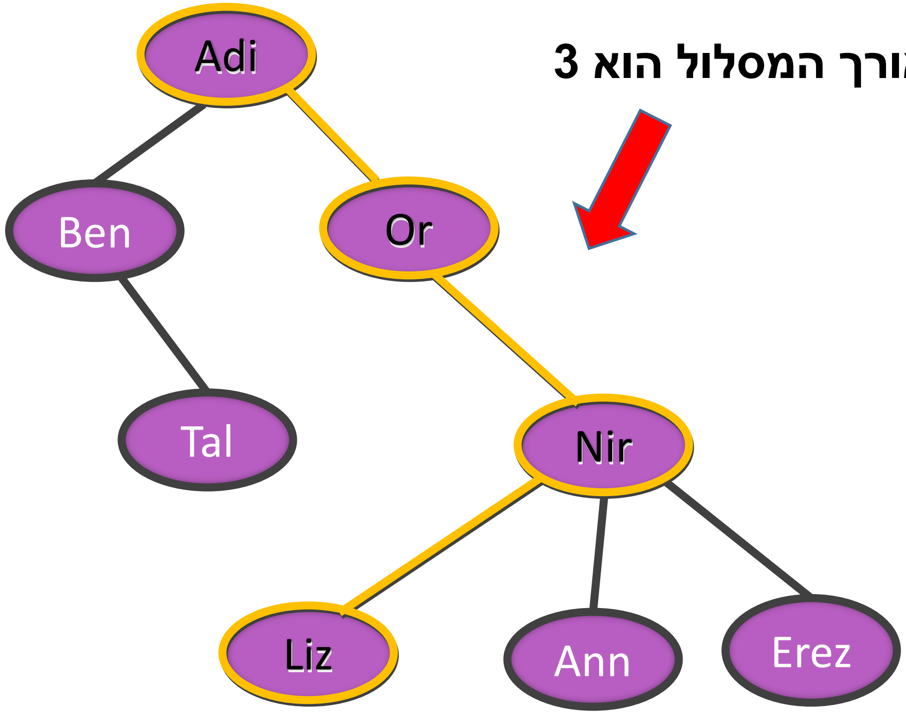
מסלול
P1=(Adi, Or, Nir, Liz)



מסלול
P2=(Or, Nir, Ann)

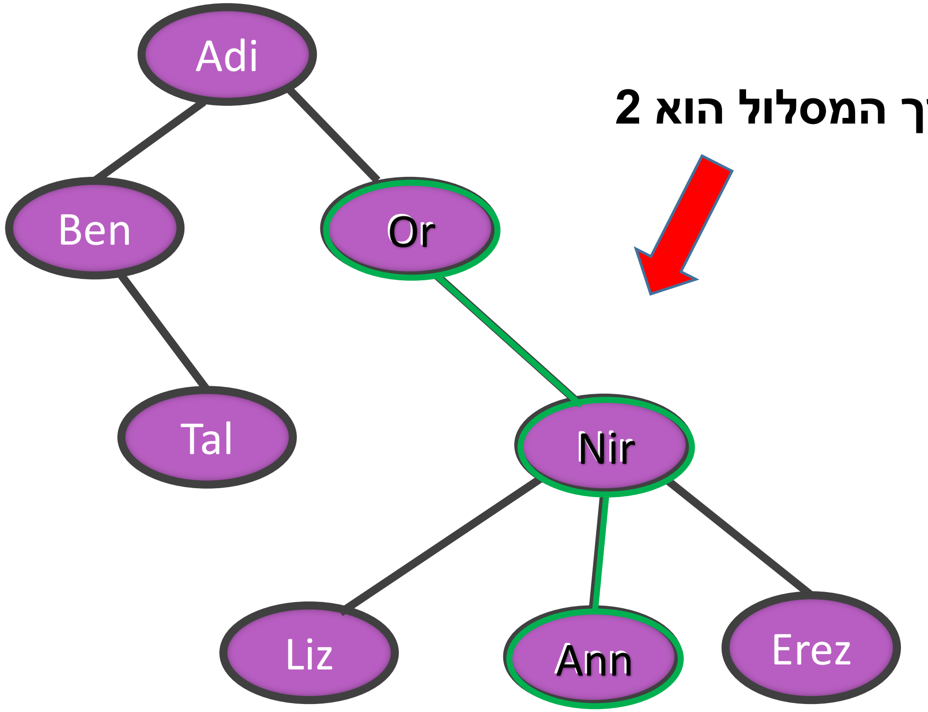
אורך המסלול (Path length)

- אורך המסלול הוא מספר הצלעות במסלול



אורך המסלול הוא 3

מסלול
P1=(Adi, Or, Nir, Liz)



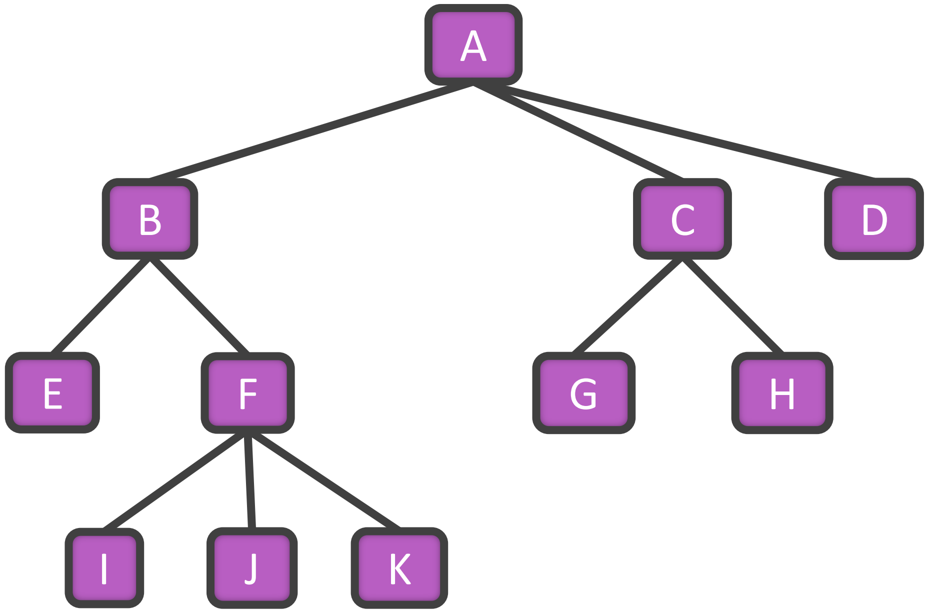
אורך המסלול הוא 2

מסלול
P2=(Or, Nir, Ann)

אב קדמון

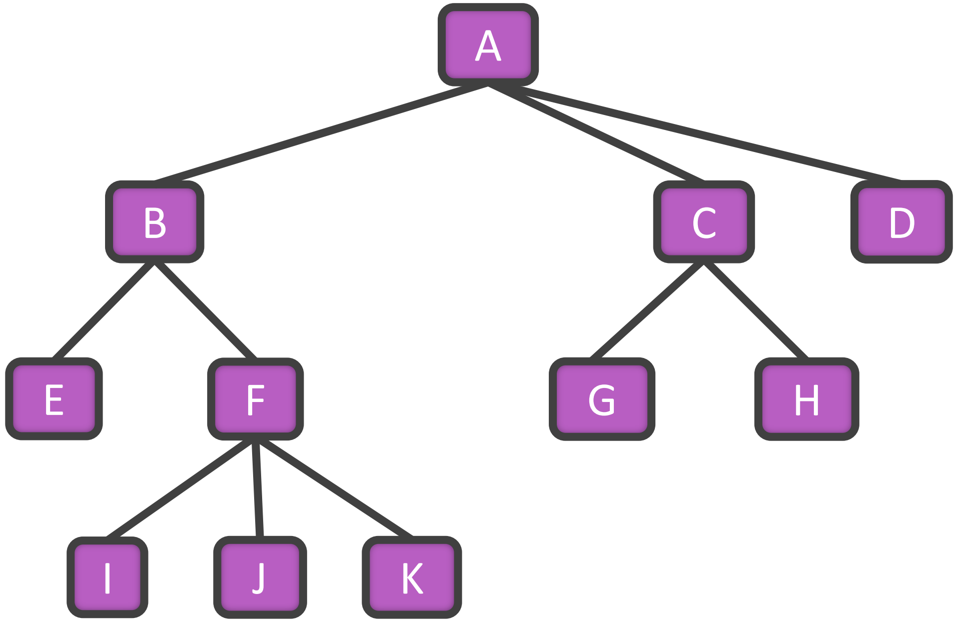
- כל צומת במסלול מהשורש לצומת x הוא **<u>אב קדמון (ancestor)</u>** של x

A,B,F-אבות קדומים של J (גם של I ו-K)



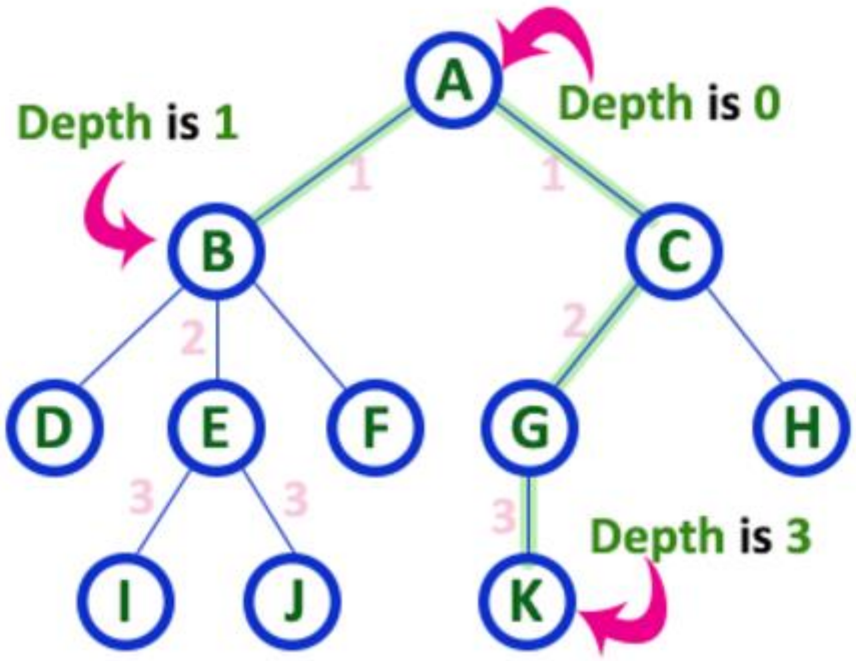
צאצא

- אם y אב קדמון של x, אזי x הוא **<u>צאצא (descendant)</u>** של y



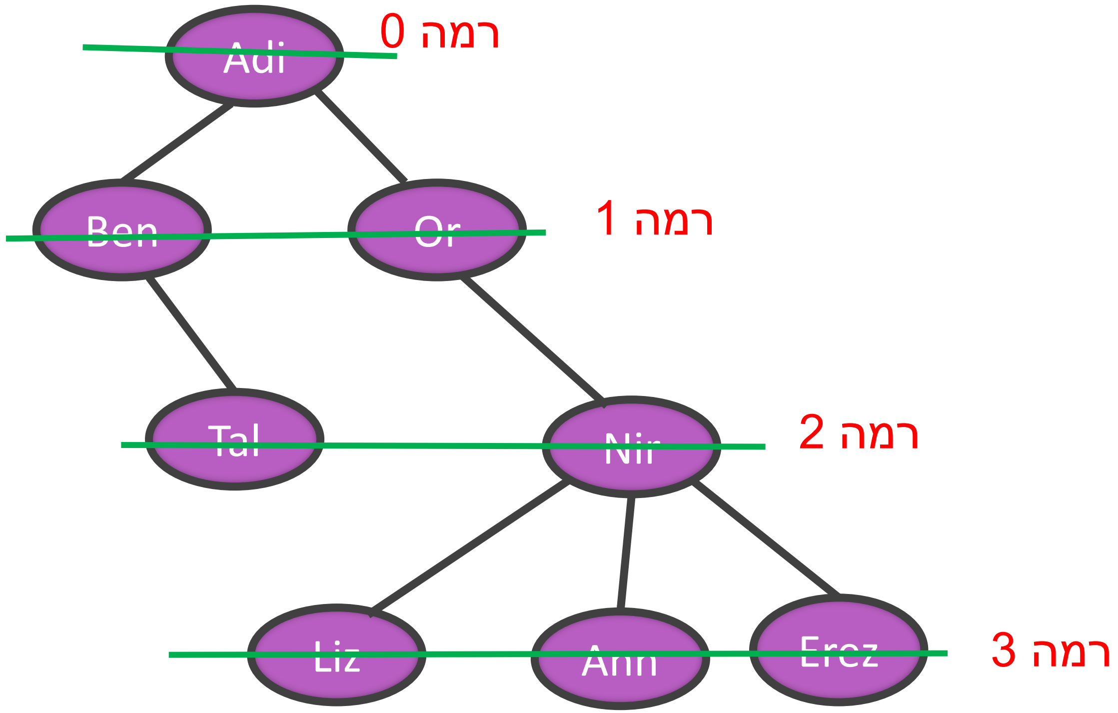
עומק של צומת (depth)

- העומק של צומת x הוא אורך המסלול (בצלעות) מהשורש לצומת x



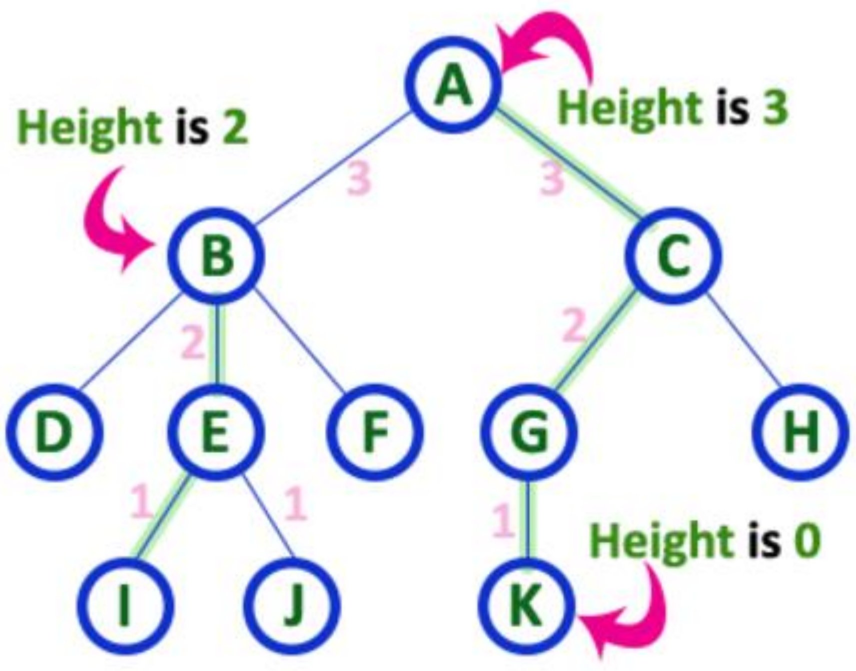
רמה (level)

- **רמה** היא אוסף צמתים בעלי עומק זהה



גובה (height)

- <u>**גובה הצומת** v</u> הוא אורך המסלול הארוך ביותר מצומת $\underline{v}$ לעלה בתת עץ של v

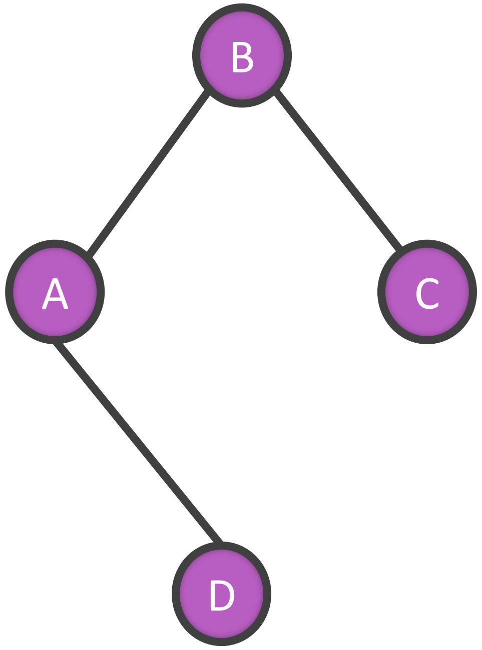


גובה העץ

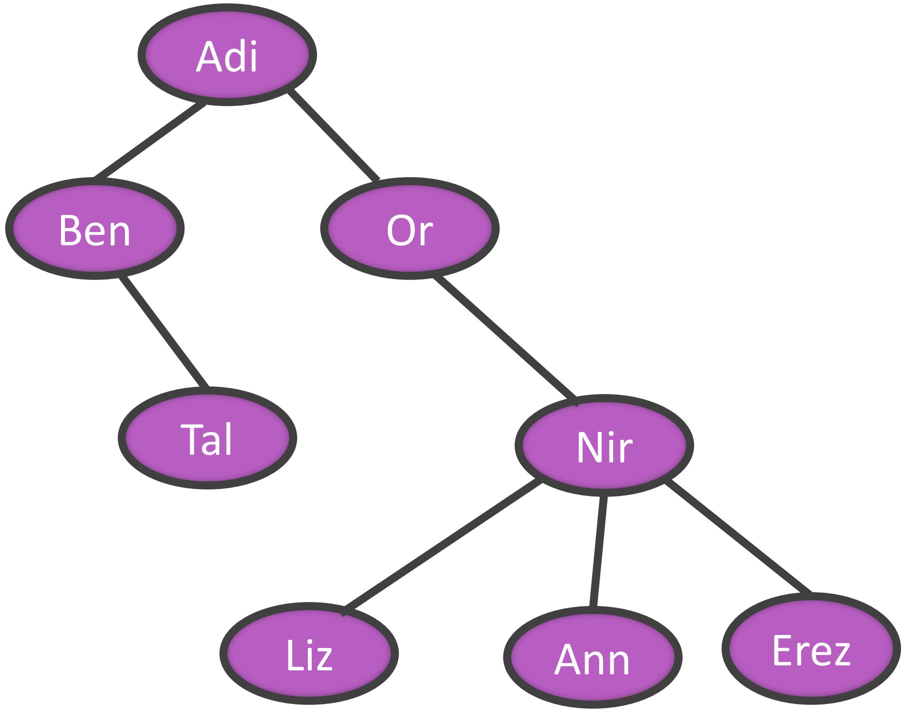
- **<u>גובה של עץ</u>** הוא גובה של שורש העץ
  - גובה של עץ ריק מוגדר להיות 1-

גובה העץ ?

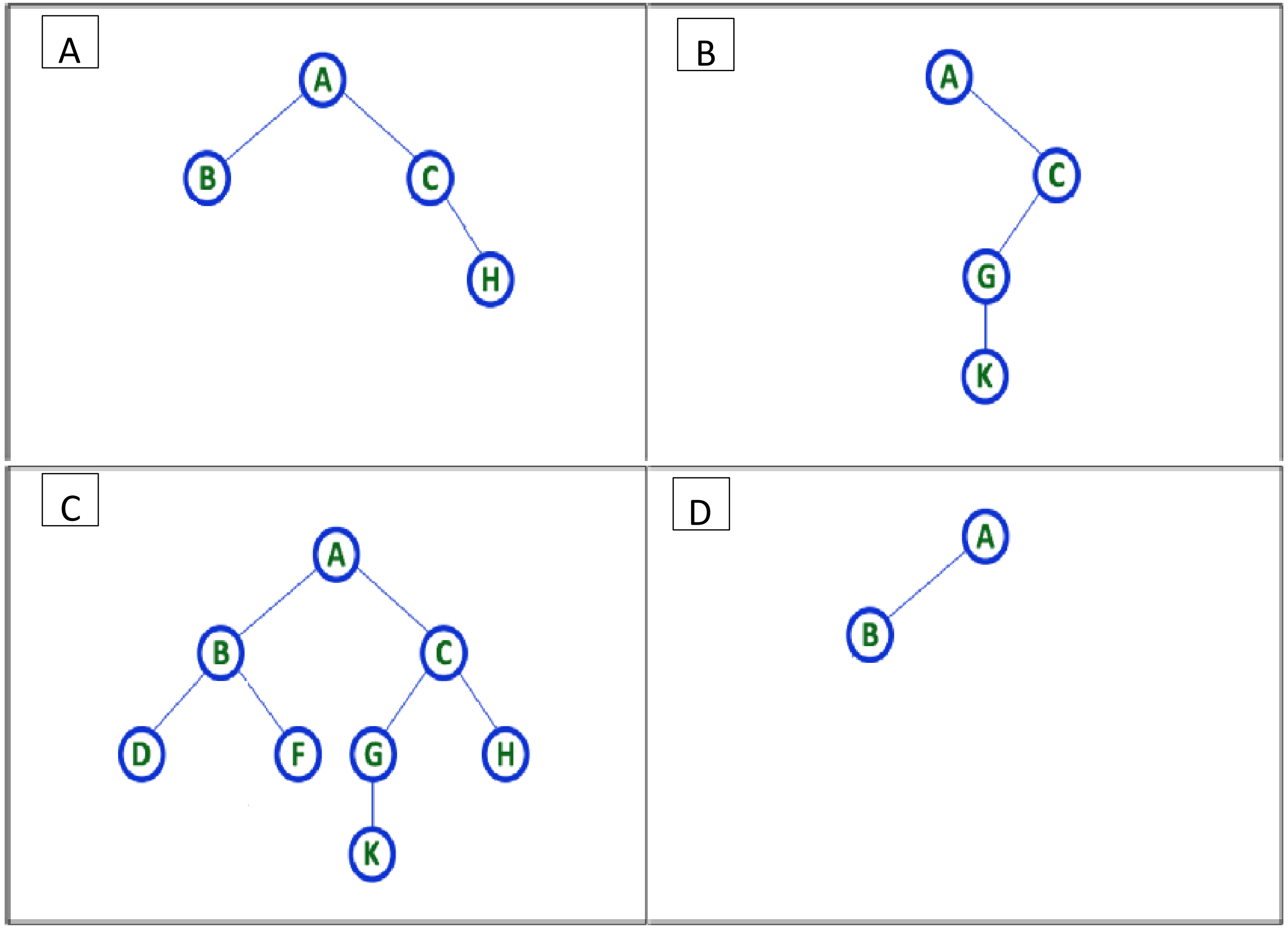
גובה העץ הוא 2



גובה העץ הוא 3



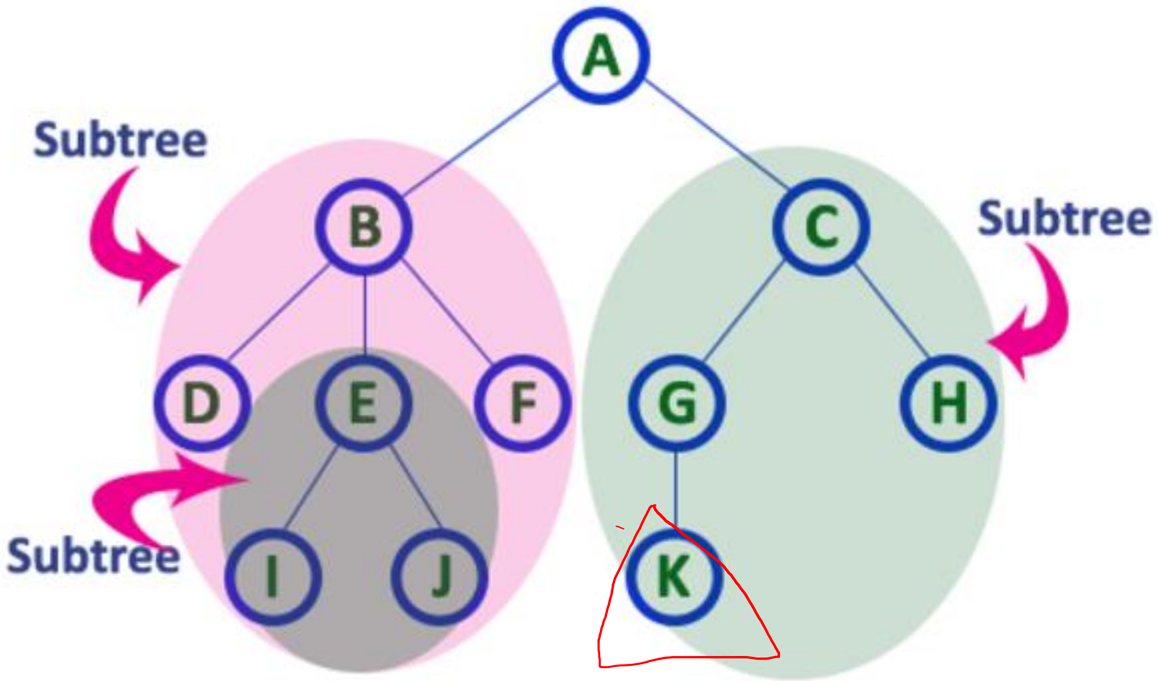
שאלה: אילו מהעצים הבאים הם עצים בעלי גובה 3?



1. A
2. B
3. C
4. D
5. אף אחד מהם

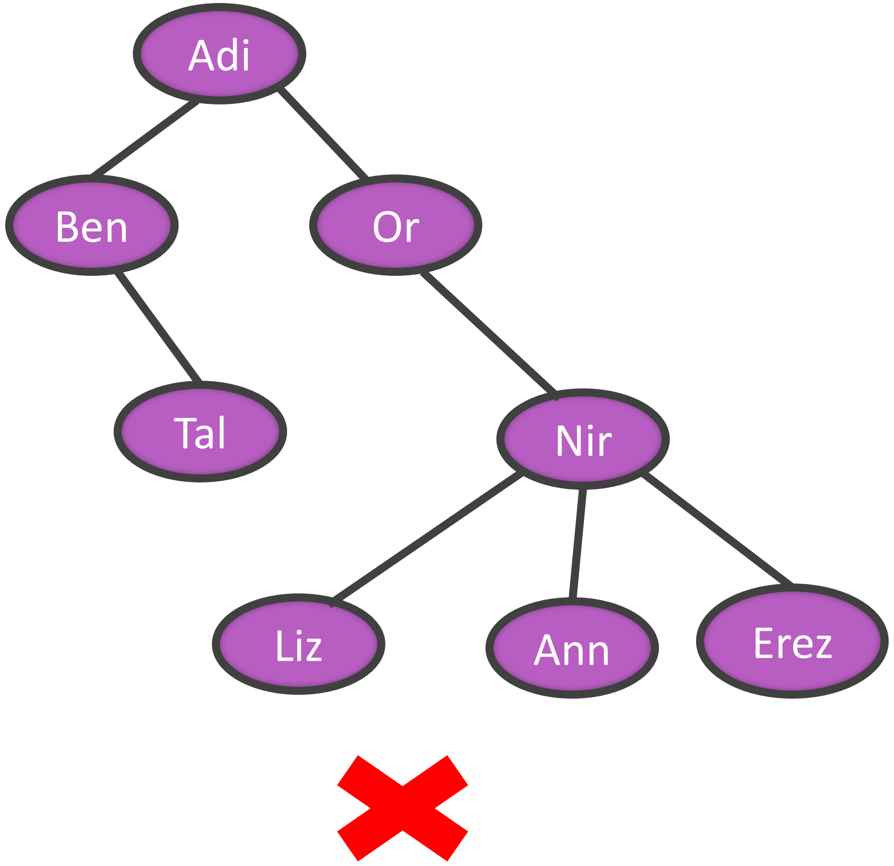
תת עץ (sub-tree)

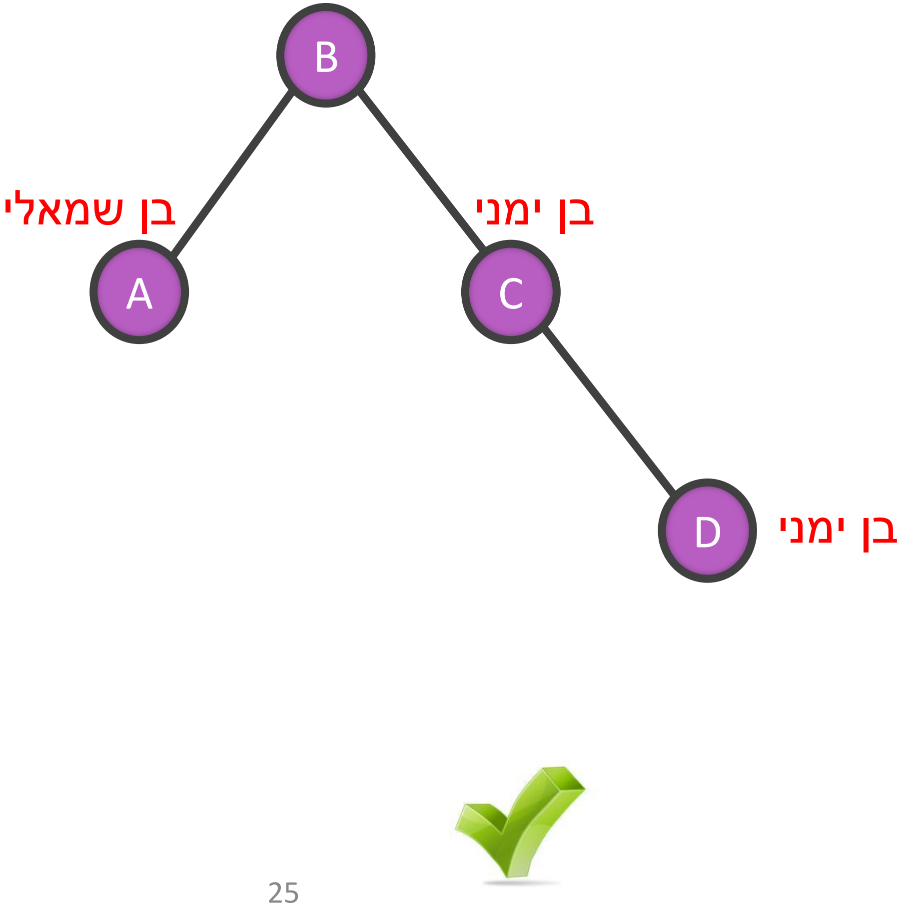
- **תת העץ המושרש ב-x** הוא העץ שיוצרים צאצאיו של x, ו-x הוא שורשו



עץ בינארי

עץ בינארי - עץ בו לכל צומת יש לכל היותר שני בנים, **בן שמאלי ובן ימני**.

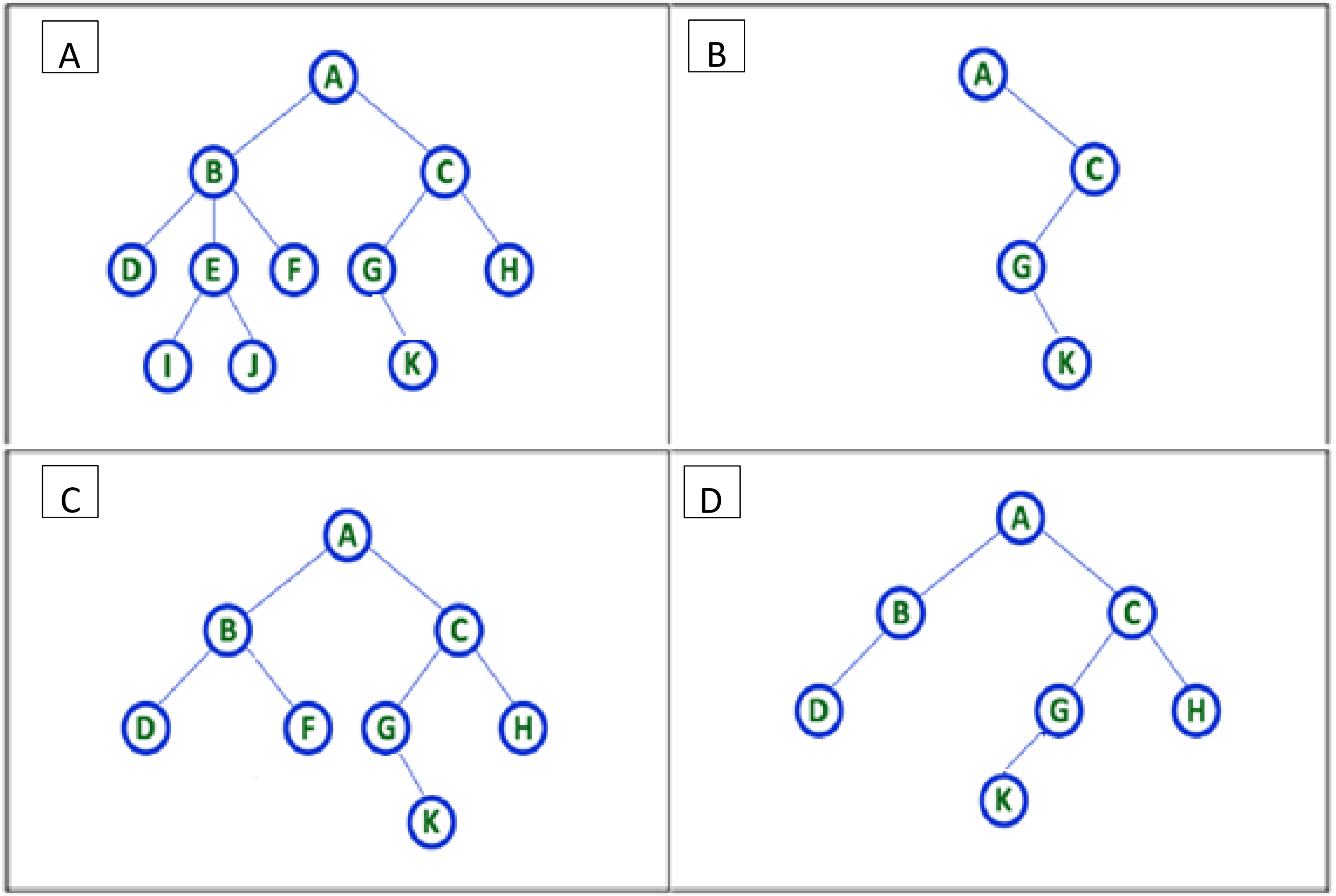




עץ בינארי (Binary tree)

- שני העצים הללו אינם אותו עץ בינארי

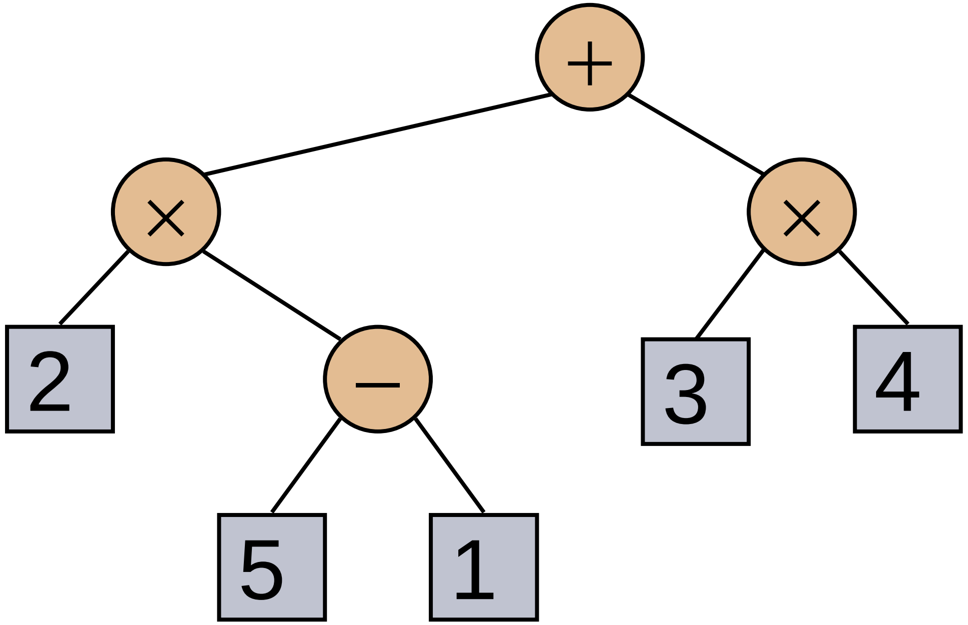
שאלה: אילו מהעצים הבאים הם עצים בינאריים?



1. A
2. B
3. C
4. D
5. אף אחד מהם

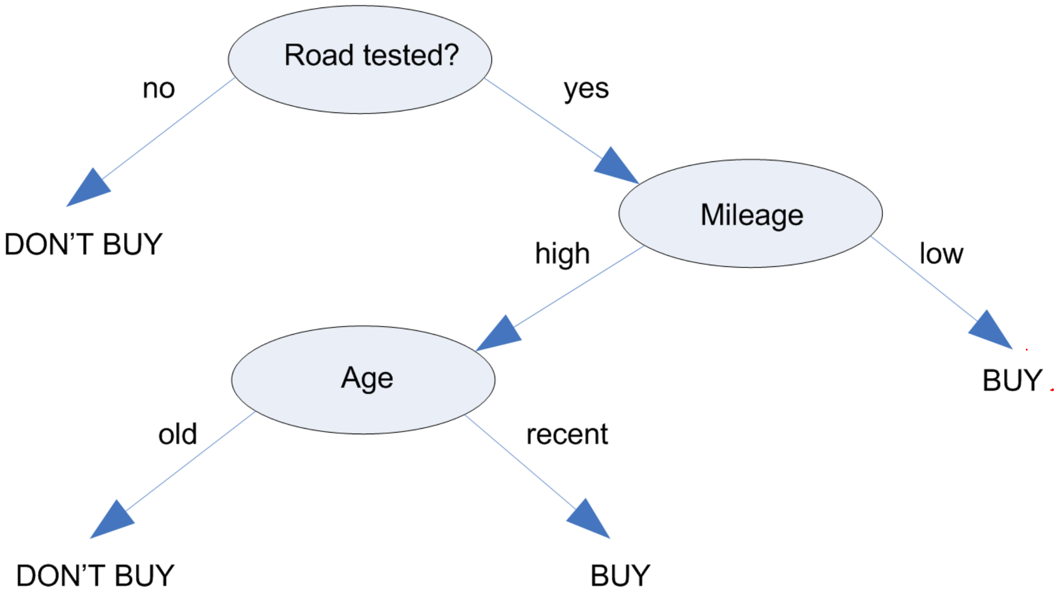
דוגמא לעץ בינארי

- עץ בינארי שמתאר ביטוי אריתמטי.
- הצמתים הפנימיים הם אופרטורים והעלים הם מספרים/משתנים



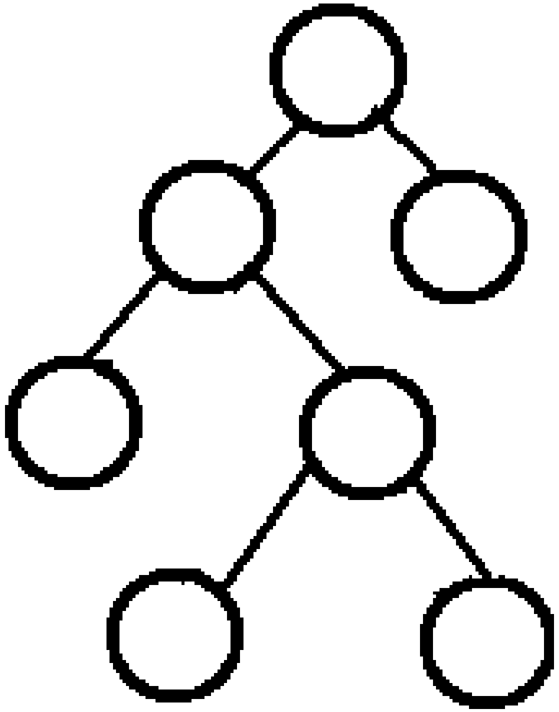
דוגמא לעץ בינארי

- עץ החלטות (קניית רכב)
- הצמתים הפנימיים הם שאלות והעלים הם החלטות

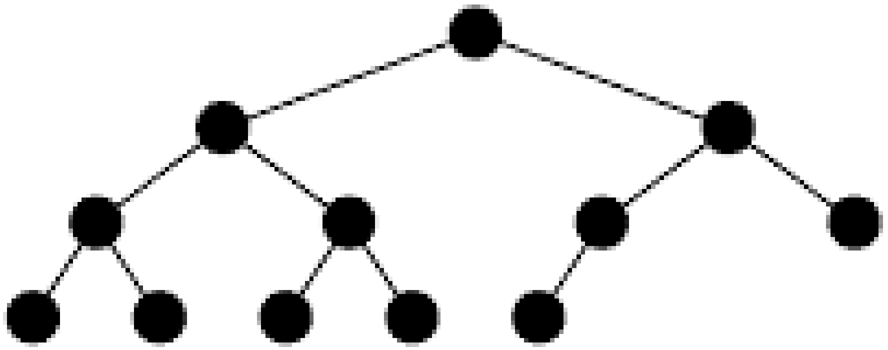


עץ בינארי מלא (Full Binary Tree)

- עץ בינארי מלא הוא עץ שבו לכל צומת יש 0 או 2 בנים (אין צומת עם דרגה 1)



עץ מלא



עץ לא מלא

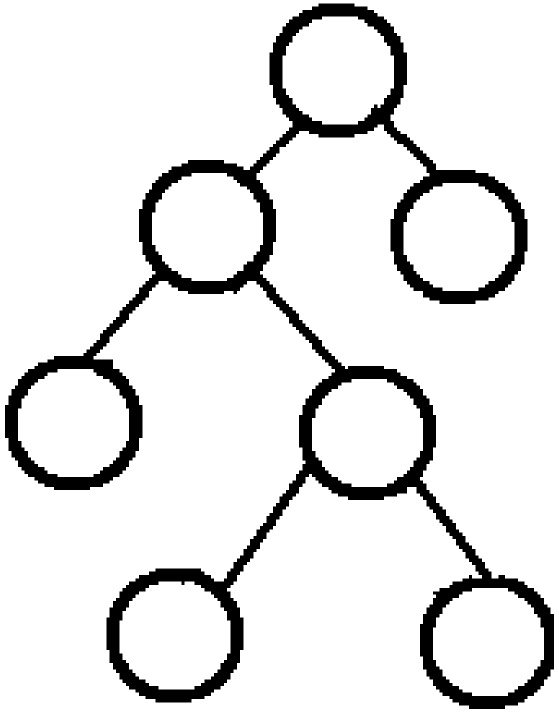
עץ בינארי מלא

משפט: יהי T עץ בינארי מלא.

נסמן את מספר העלים בעץ T ב– $l(\mathrm{T})$

נסמן את מספר הצמתים הפנימיים בעץ T ב- $m(T)$

בעץ בינארי מלא T מתקיים $l(T) = m(T) + 1$

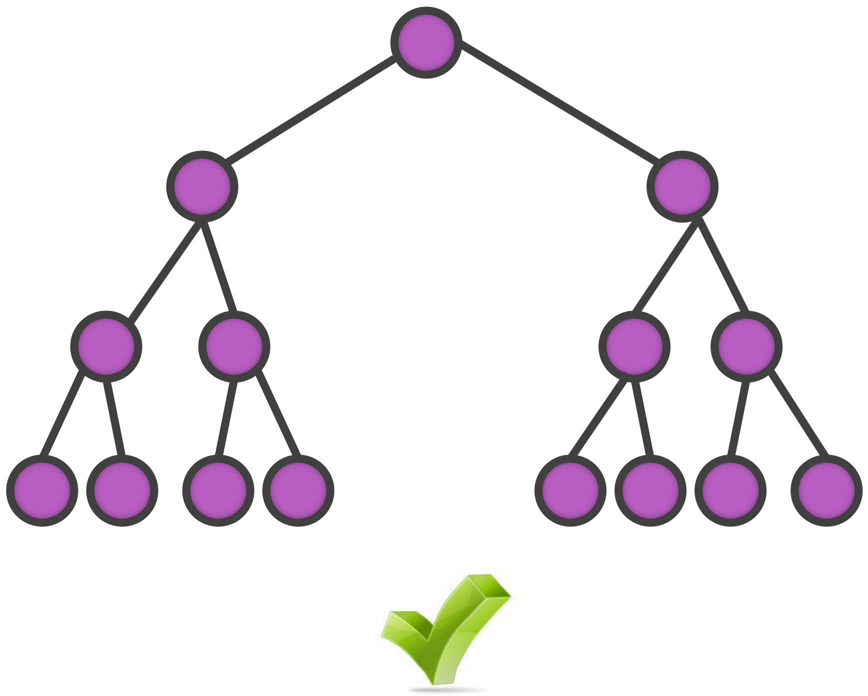


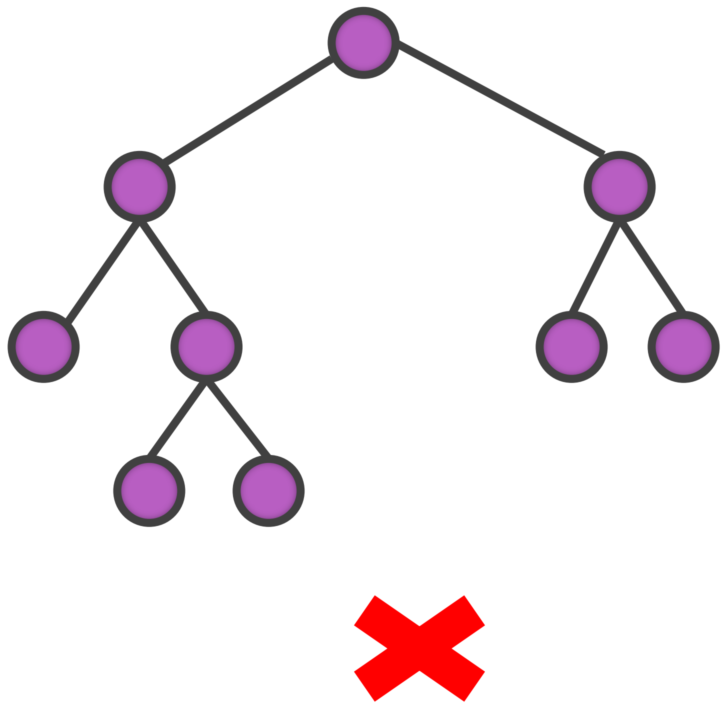
משפט: בעץ בינארי מלא T מתקיים $l(T) = m(T) + 1$

הוכחה:

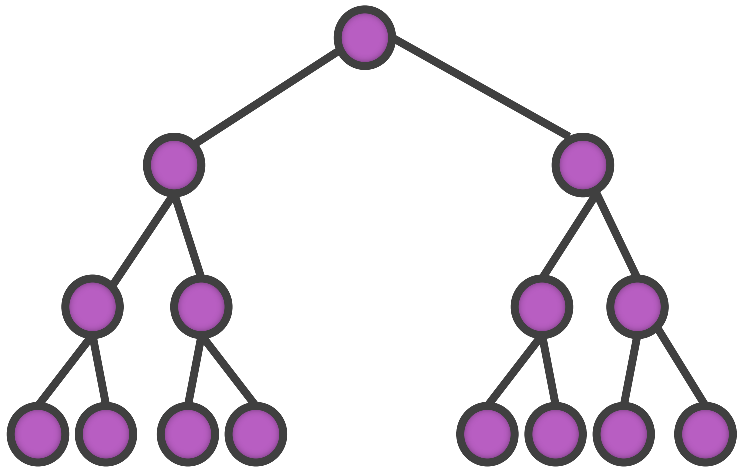
עץ בינארי מושלם (Perfect Binary Tree)

עץ בינארי מושלם - עץ בינארי מלא בו כל עלים באותו העומק





עץ בינארי מושלם



משפט: בעץ בינארי מושלם בגובה h

- מספר הצמתים הפנימיים הוא $2^h - 1$
- מספר העלים הוא 2^h
- מספר הכולל של צמתים הוא $2^{h+1} - 1$

הוכחה:

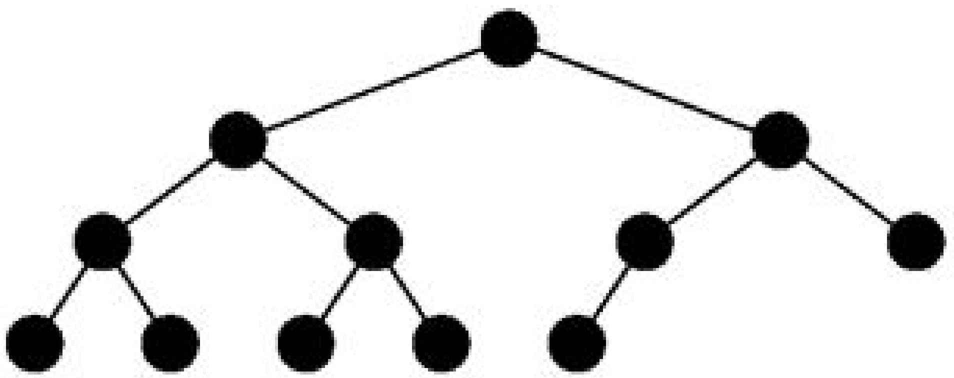
משפט: בעץ בינארי מושלם בגובה h

- מספר הצמתים הפנימיים הוא $2^h - 1$
- מספר העלים הוא 2^h
- מספר הכולל של צמתים הוא $2^{h+1} - 1$

הוכחה (המשך):

עץ בינארי שלם (complete binary tree)

- **עץ בינארי שלם (complete binary tree)** הוא עץ בו כל רמה, פרט לאחרונה היא מלאה בה כל הצמתים מרוכזים מצד שמאל



משפט: **גובהו של עץ בינארי שלם בעל n צמתים הוא $\lfloor \log n \rfloor$**

הוכחה:

משפט: **גובהו של עץ בינארי שלם בעל n צמתים הוא $\lfloor \log n \rfloor$**

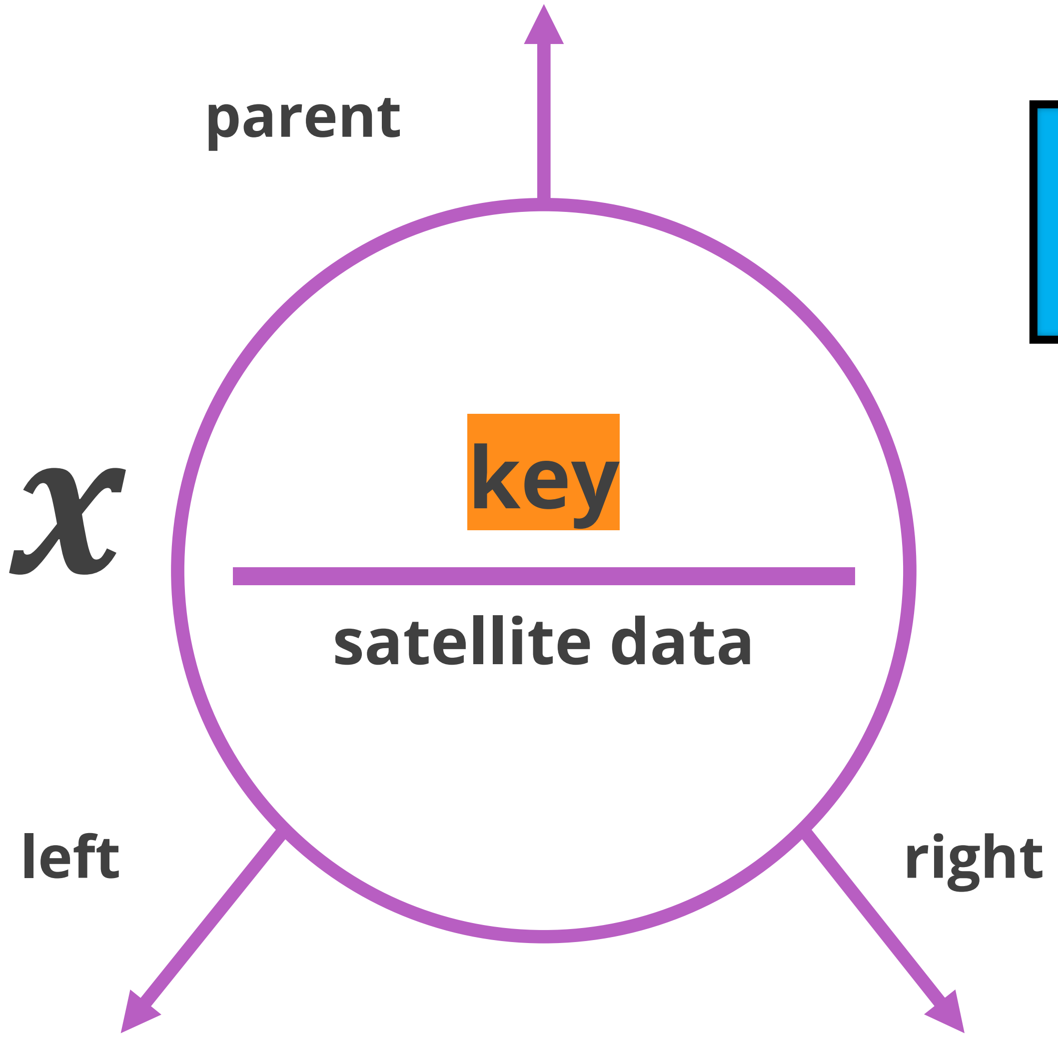
הוכחה (המשך):

ייצוג של עץ בינארי

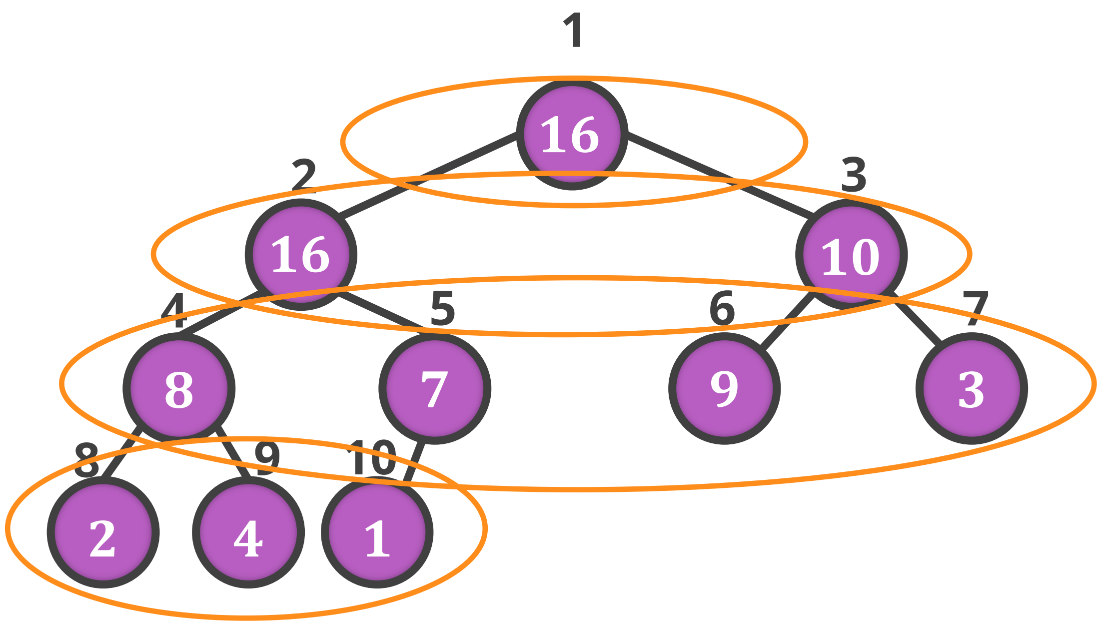
- ייצוג של עץ בינארי
  1. כמבנה מקושר
  2. באמצעות מערך

ייצוג של עץ בינארי כמבנה מקושר

העץ נתון על ידי מצביע לשורש $T.root$



ייצוג עץ בינארי באמצעות מערך



	1	2	3	4	5	6	7	8	9	10
A	16	16	10	8	7	9	3	2	4	1

נממש עץ באמצעות מערך

שורש העץ – A [1]

בהינתן אינדקס i של הצומת

left(i) return $2i$ right(i) return $2i + 1$	parent(i) return $\lfloor i/2 \rfloor$

יתרונות וחסרונות של כל אחד מהייצוגים

ייצוג של עץ בינארי באמצעות מערך	ייצוג של עץ בינארי כמבנה מקושר

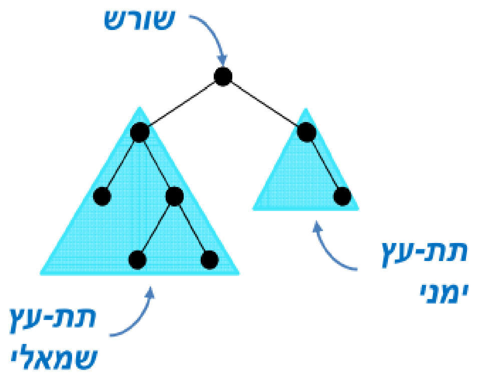
אלגוריתמי סריקה לעצים בינאריים

- **סריקה (traversal)** – מעבר שיטתי על צמתים וצלעות של העץ

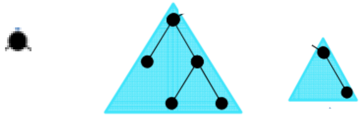
Preorder

- מבקרים קודם בקודקוד ואחריו בתת העץ השמאלי ולבסוף בתת העץ הימני

Preorder(x)
    if ($x \neq Null$)
        print $x.key$
        Preorder($x.left$)
        Preorder($x.right$)

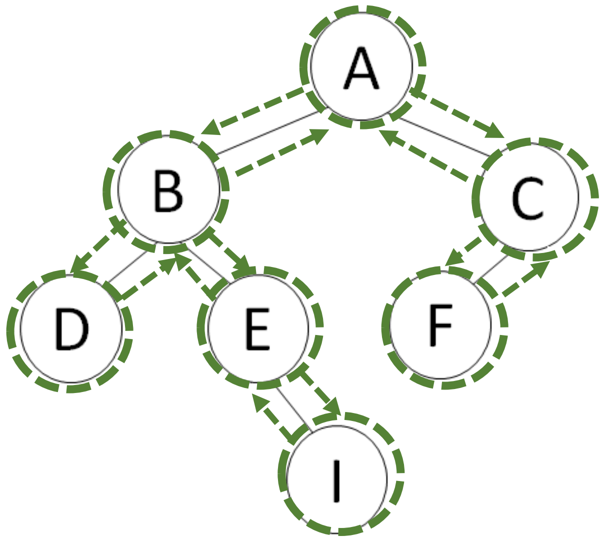


pre-order – שורש, תת עץ שמאלי, תת עץ ימני



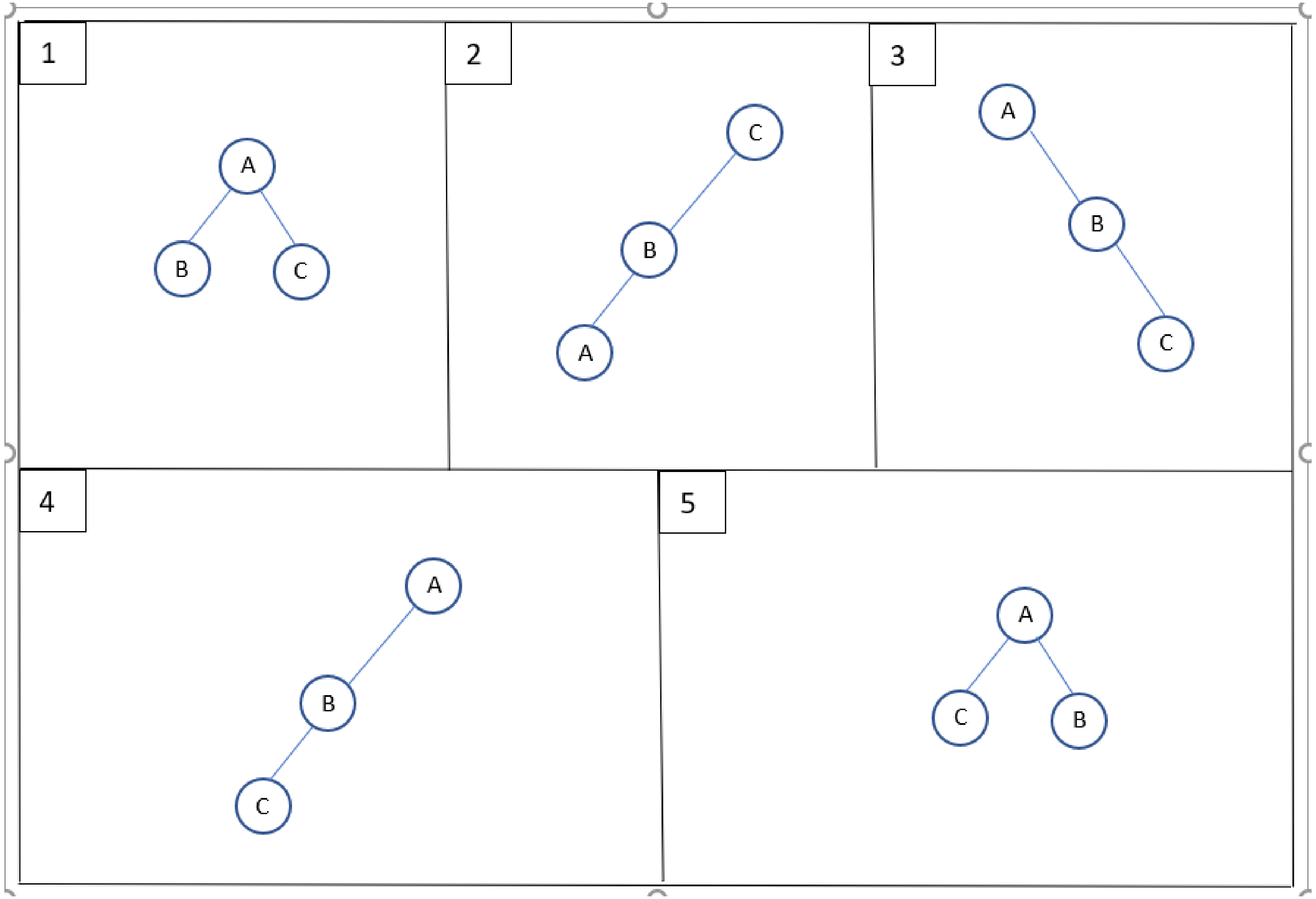
Preorder

- מה יהיה הפלט של סריקת Preorder על העץ הזה?



A, B, D, E, I, C,

שאלה: סמנו את כל העצים שסריקת ה-preorder שלהם היא סדרת הצמתים (A B C משמאל לימין).

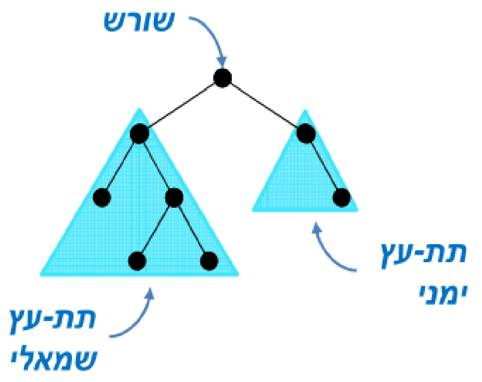


a. 1
b. 2
c. 3
d. 4
e. 5

Inorder

- עבור כל צומת x, מבקרים קודם בתת העץ השמאלי של x, אחריו בצומת x ולבסוף בתת העץ הימני

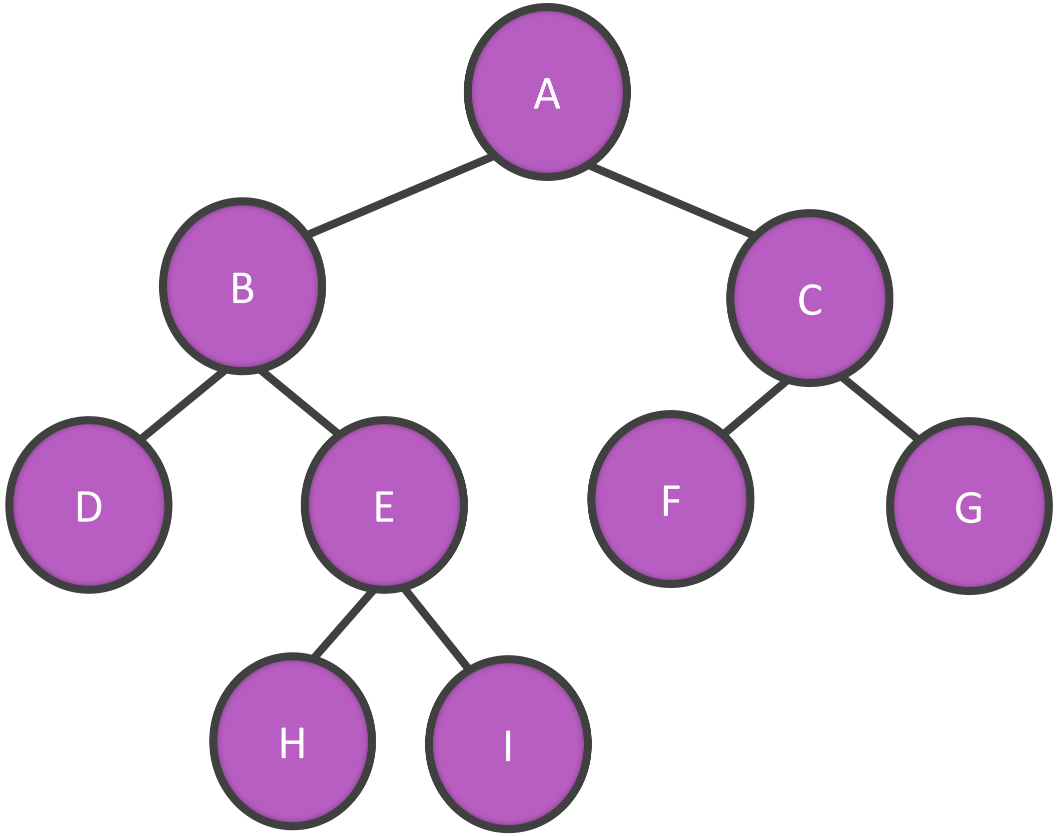
```
Inorder(x)
        if (x ≠ Null)
                Inorder(x.left)
                print x.key
                Inorder(x.right)
```



in-order – תת עץ שמאלי, שורש, תת עץ ימני

Inorder

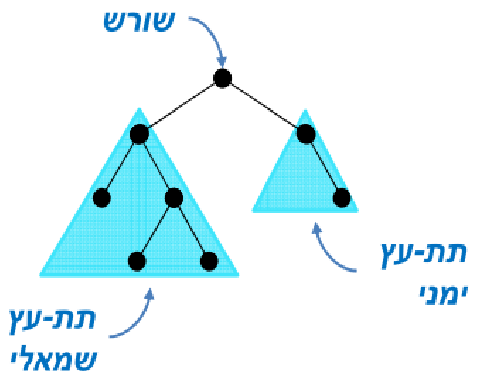
- מה יהיה הפלט של Inorder על העץ הזה?



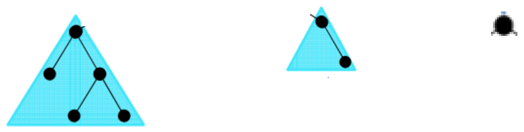
Postorder

- מבקרים קודם בתת העץ השמאלי ואחריו בתת העץ הימני ולבסוף בצומת

Postorder(x)
    if ($x \neq Null$)
        Postorder($x.left$)
        Postorder($x.right$)
        print x.key

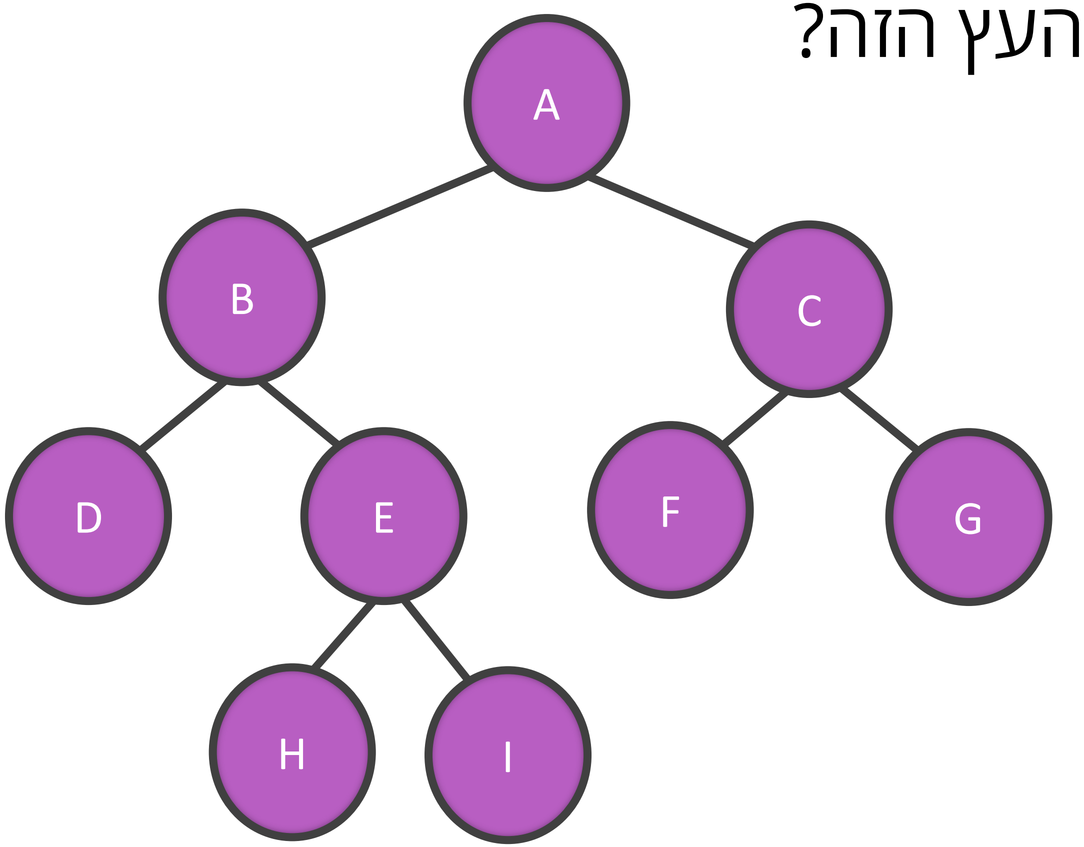


post-order – תת עץ שמאלי, תת עץ ימני, שורש



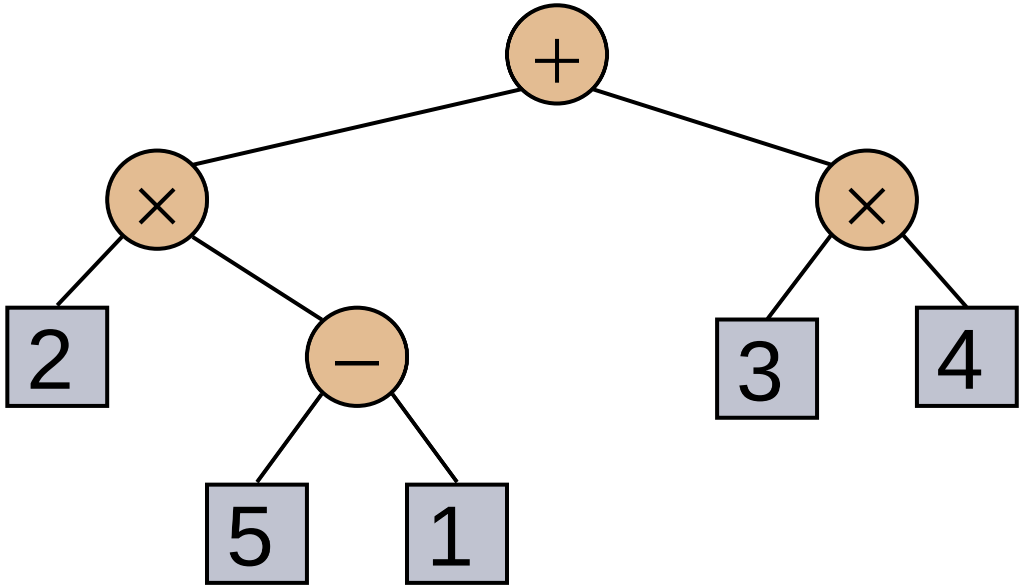
Postorder

- מה יהיה הפלט של Postorder על העץ הזה?



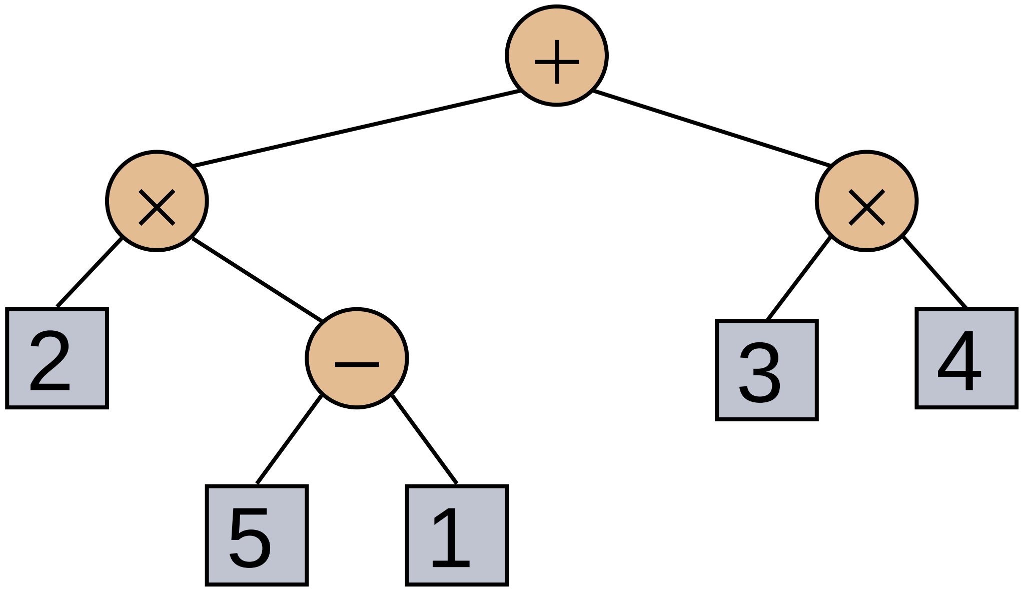
Inorder - ביטוי אריטמטי

- ניתן להשתמש ברעיון של Inorder כדי לרשום ביטוי אריתמטי שמיוצג ע"י עץ בינארי.
- לפני קריאה לעץ השמאלי נדפיס '(' ואחרי קריאה לעץ הימני נדפיס ')'



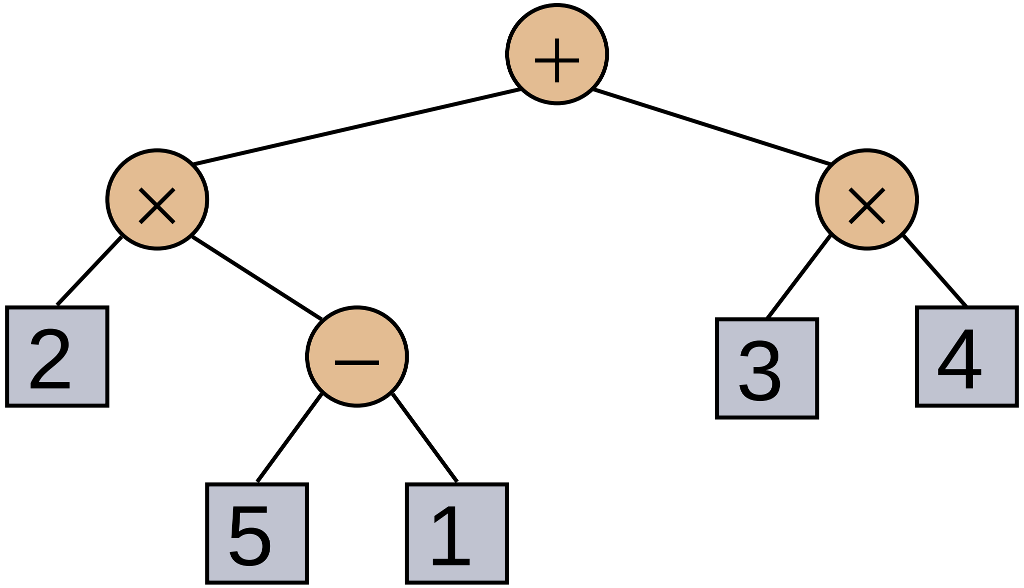
ביטוי אריתמטי - Inorder

Inorder(x)
if ($x \neq Null$)
    if($x.left \neq$Null)
        print "("
        Inorder($x.left$)
    print $x.key$
    if($x.right \neq Null$)
        Inorder($x.right$)
        print ")"



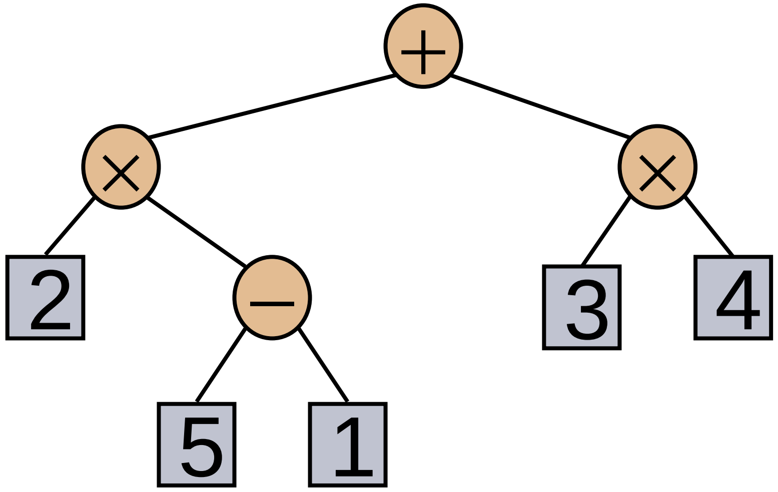
ביטוי אריתמטי - Inorder

- איך תשנו את האלג על מנת לחשב ביטוי אריתמטי?



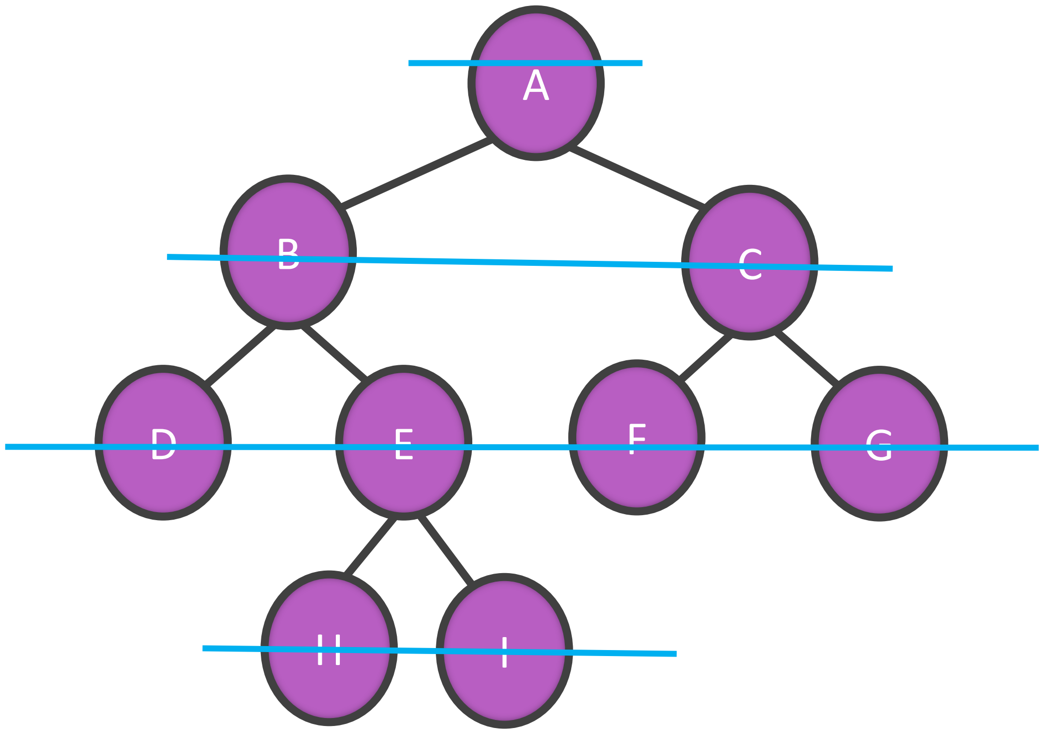
ביטוי אריתמטי - Inorder

Calculate (x)
        if ($x \neq Null$)
                if(x is $NOT\ leaf$)
                        $resL$ = Calculate($x.left$)
                        $resR$ = Calculate($x.right$)
                        $op = x.key$
                        return $resL$ op $resR$
                else
                        return $x.key$



סריקה לפי רמות (Levelorder)

- מבקרים בקודוקודים לפי הרמות מלמעלה למטה. בכל רמה ההתקדמות היא משמאל לימין.



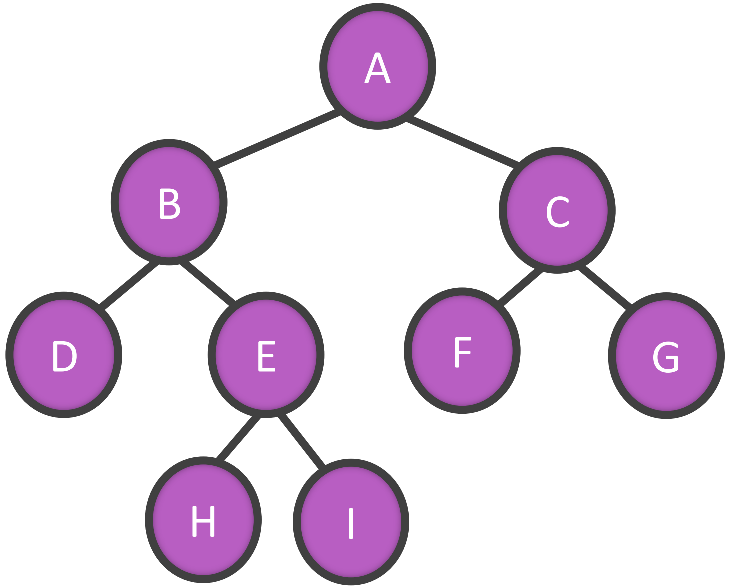
A B C D E F G H I

- באיזה מבנה נתונים כדאי להשתמש?

סריקה לפי רמות (Level-order)

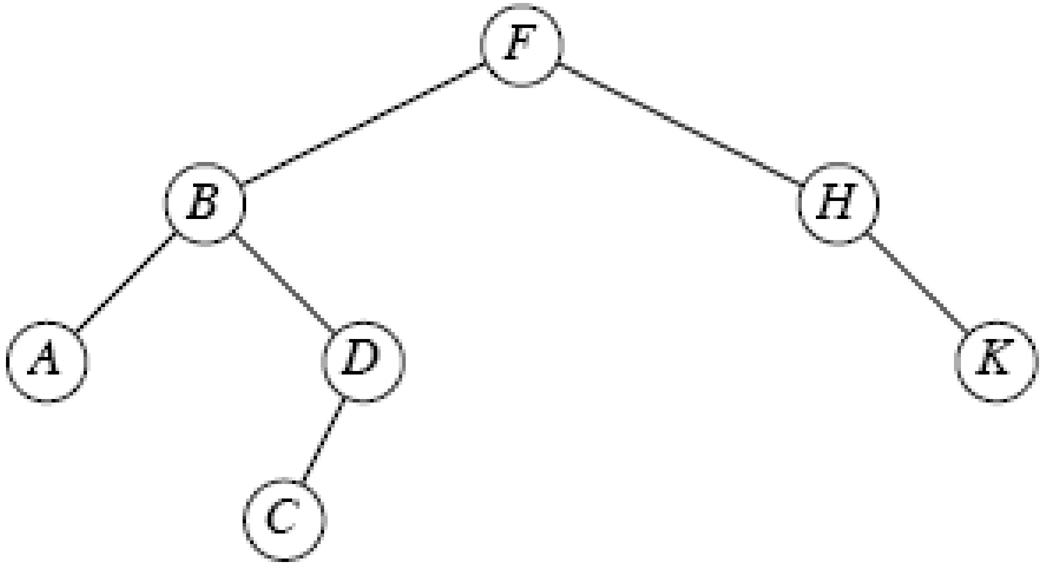
- מבקרים בקודקודים לפי הרמות מלמעלה למטה.

Level-order(Tree T)
Q ←create empty queue
x ←root(T)
while($x \neq Null$)
    print $x.key$
    enqueue($Q, x.left$)
    enqueue($Q, x.right$)
    $x \leftarrow$ dequeue(Q)

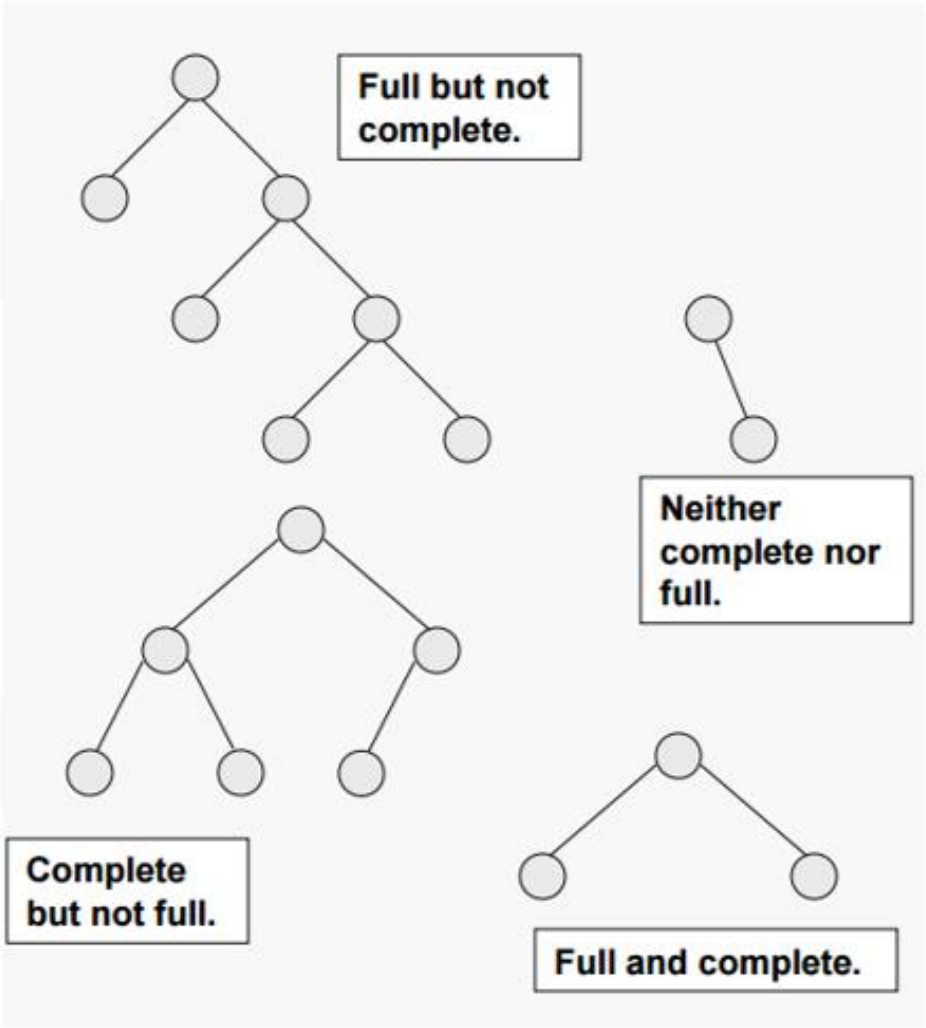


Level-order

- מה יהיה הפלט של Level-order על העץ הזה?



סיכום


Full but not complete.
Neither complete nor full.
Complete but not full.
Full and complete.

סיכום

- עץ - תכונות והגדרות
- אלגוריתמי סריקה בעץ
- ייצוגים שונים של העץ
- שימוש נכון באלגוריתמי הסריקה השונים