"והיה כעץ שתול על מים" (ירמיה יז,ח)

עצים

עץ - Tree מושגים

parent : 7 הוא **האבא** (הורה) של 28

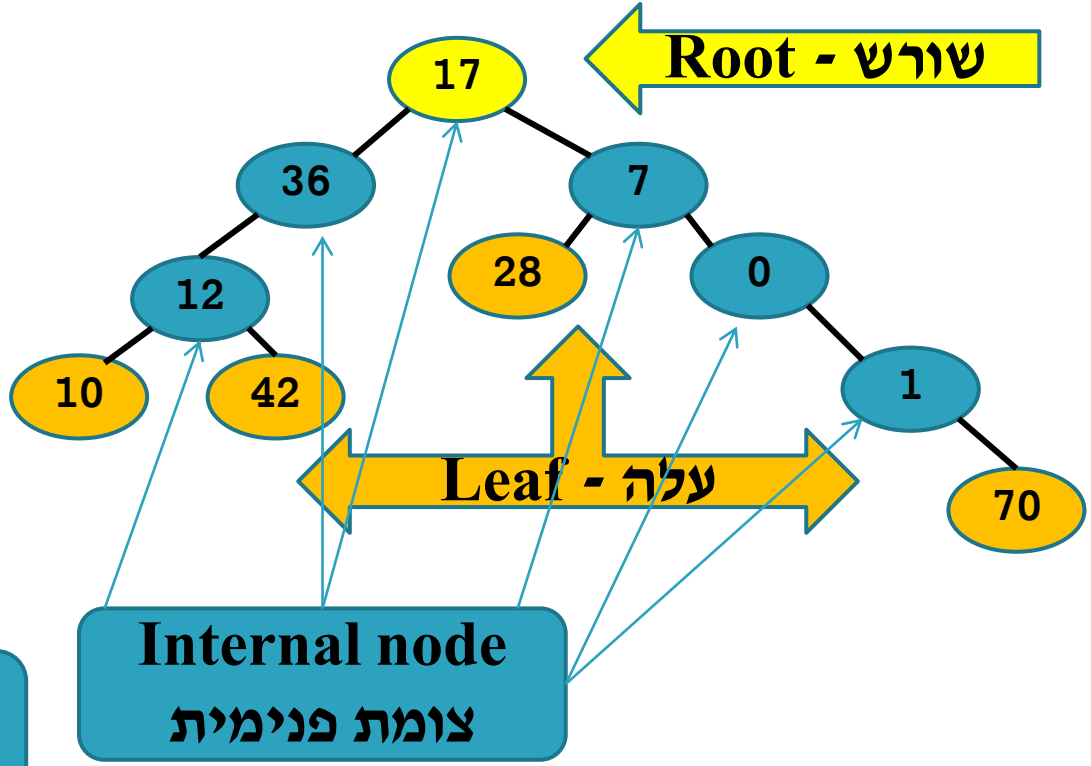
child : 28 הוא **הבן** של 7

descendent : 28 הוא **צאצא** של 17

ancestor : 17 הוא **אב-קדמון** של 28

degree : **הדרגה** (מספר הבנים) של 12 הוא 2

rooted tree: **עץ משורש** הוא עץ עם שורש אחד ובנים שכל בן הוא עץ



האם 7
אב-קדמון של 1?

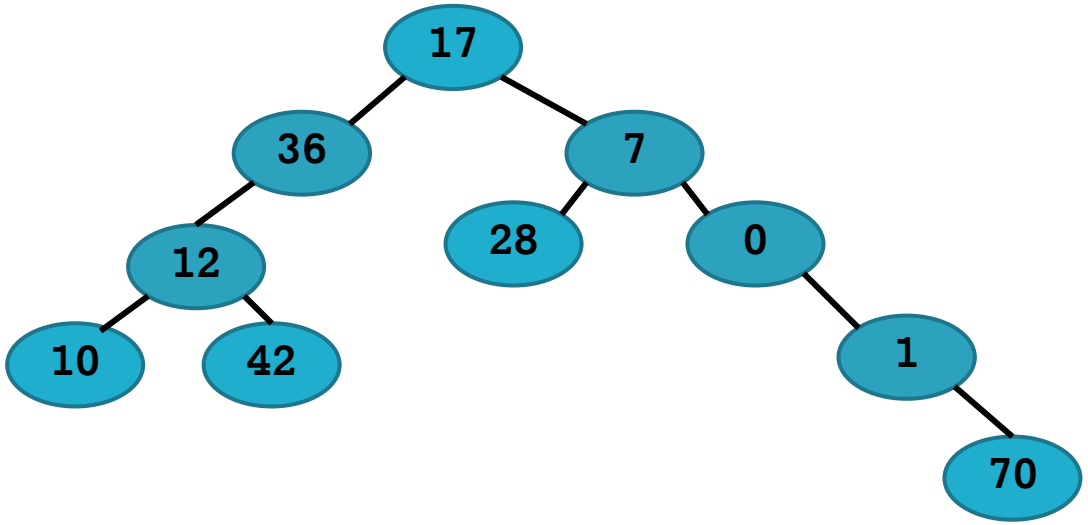
עץ - Tree מושגים

path : **מסלול** - סדרה של צמתים מהורה לצאצא

path length : **אורך מסלול** - מספר הקשתות במסלול

height : **גובה** - אורך המסלול הארוך ביותר מהשורש לעלה

depth,level : **רמה, עומק** – של צומת הוא אורך המסלול מהשורש עד לצומת



מה הגובה של העץ בדוגמא?

עץ בינארי - Binary Tree

הגדרה:

עץ ריק

או

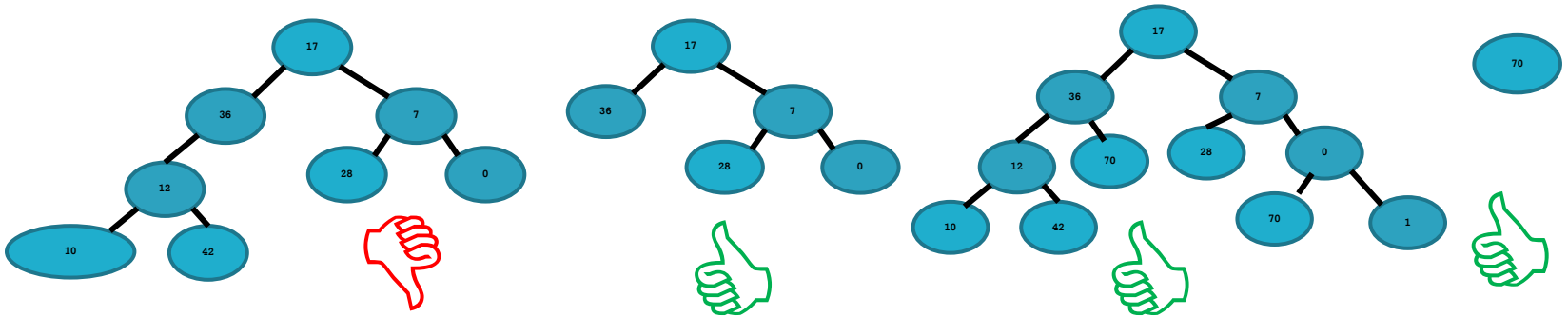
שורש ו-2 תתי עצים , בן שמאלי ובן ימיני כך

שכל תת-עץ הוא עץ בינארי

עץ בינארי - Binary Tree - סוגים

עץ בינארי מלא - full binary tree :

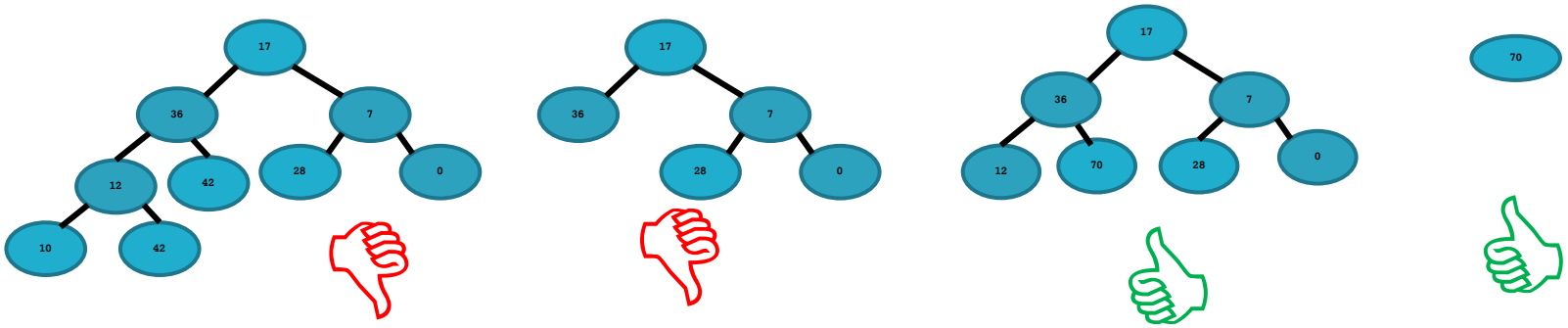
עץ בינארי T ייקרא עץ בינארי מלא אם לכל צומת פנימית יש בדיוק שני בנים לא ריקים



עץ בינארי - Binary Tree - סוגים

עץ בינארי שלם - complete binary tree:

עץ בינארי T ייקרא שלם אם הוא עץ בינארי מלא וכל העלים הם באותו עומק



complete binary tree - עץ בינארי שלם

מספר הצמתים בעץ שלם T בגובה h הוא:

$$n = 2^{h+1} - 1$$

גובה עץ שלם T עם n צמתים הוא:

$$h = \log(n+1) - 1$$

$$h = \theta(\log n)$$

ובעץ בינארי כלשהוא:

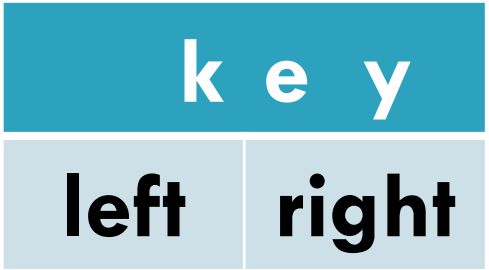
$h = O(n)$

וגם

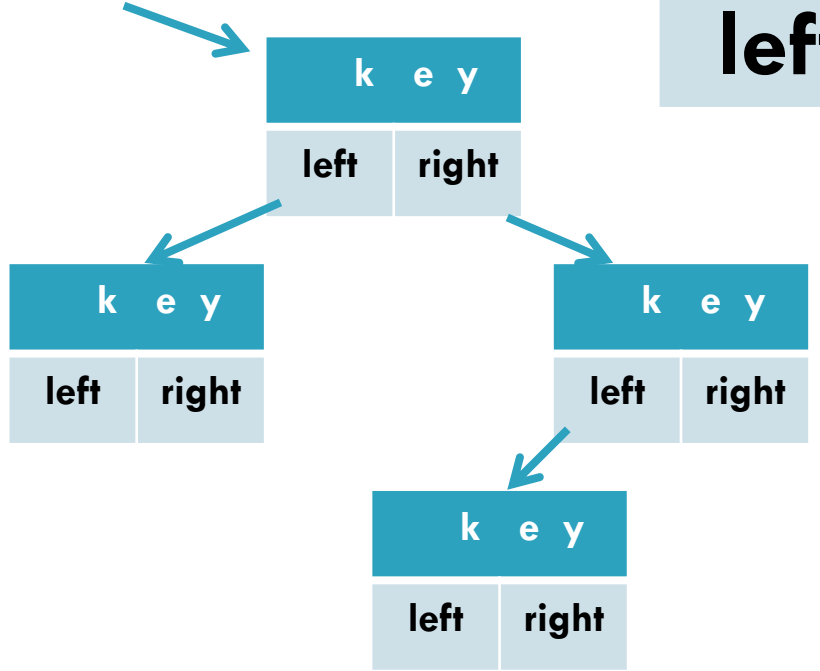
$h = \Omega(\log n)$

ייצוג של עץ בינארי

ייצוג של צומת x



ייצוג של עץ T



מעברים על עץ בינארי - Binary Tree Transversal

המשימה - לבקר בכל צומת פעם אחת בלבד

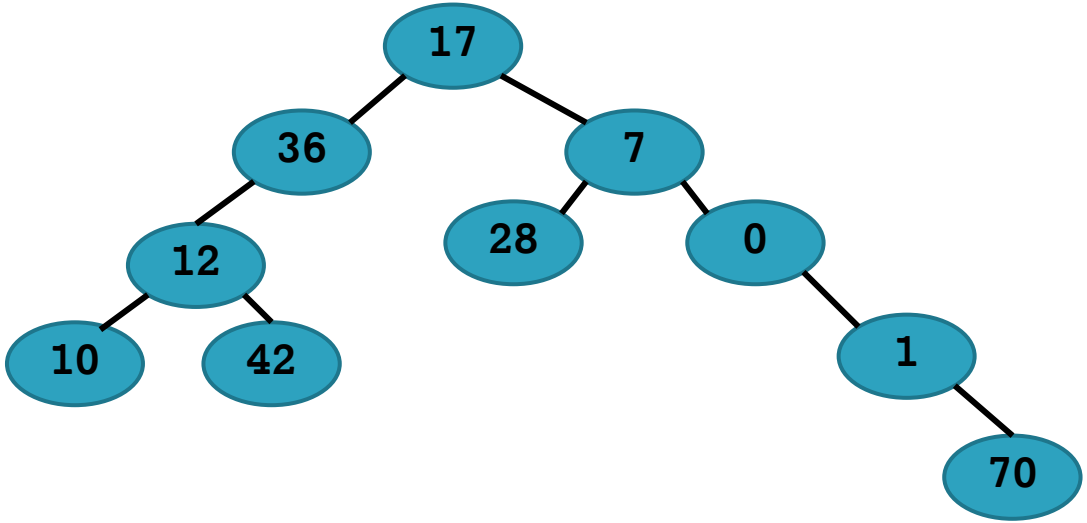
הפתרון -

מעבר עומק

מעבר רוחב

מעבר עומק - depth

תוכי - inorder

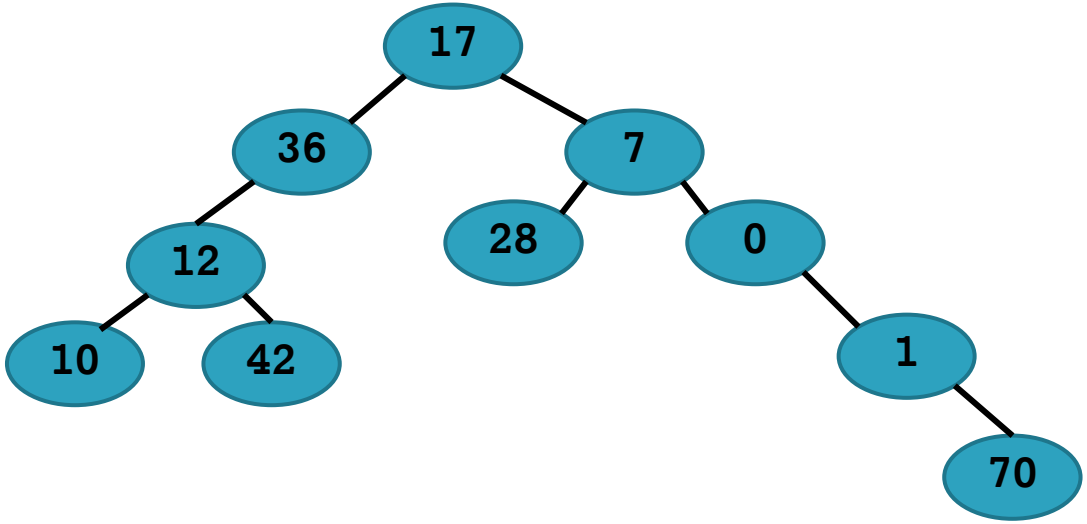


in: 10, 12, 42, 36, 17, 28, 7, 0, 1, 70

```
inorder(T)
  if not emptyTree(T)
      inorder (left(T))
      visit (root (T))
      inorder (right(T))
```

הדגמה

מעבר עומק - depth



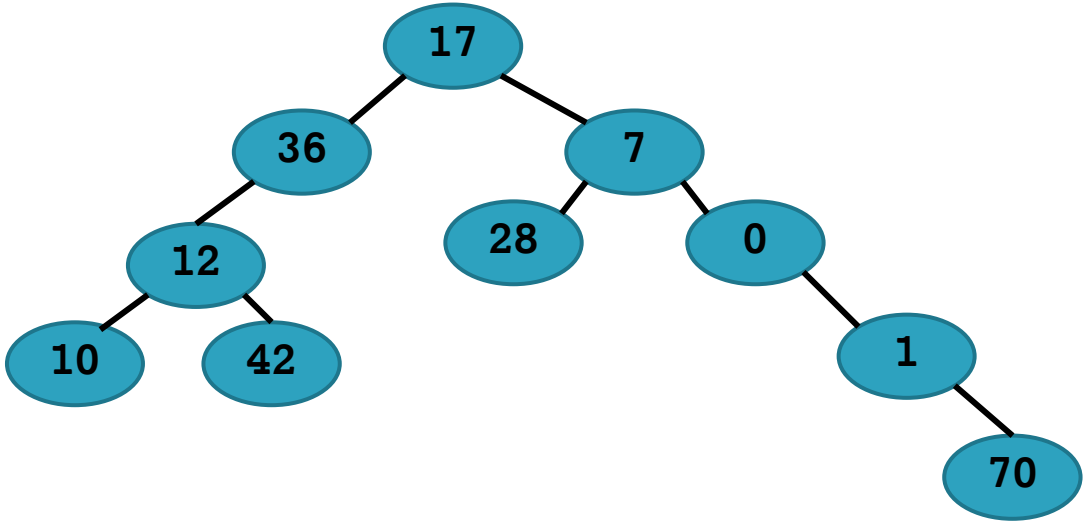
תחילי - preorder

```
preorder(T)
  if not emptyTree(T)
      visit (root (T))
       preorder (left(T))
       preorder (right(T))
```

Pre: 17, 36, 12, 10, 42, 7, 28, 0, 1, 70

מעבר עומק - depth

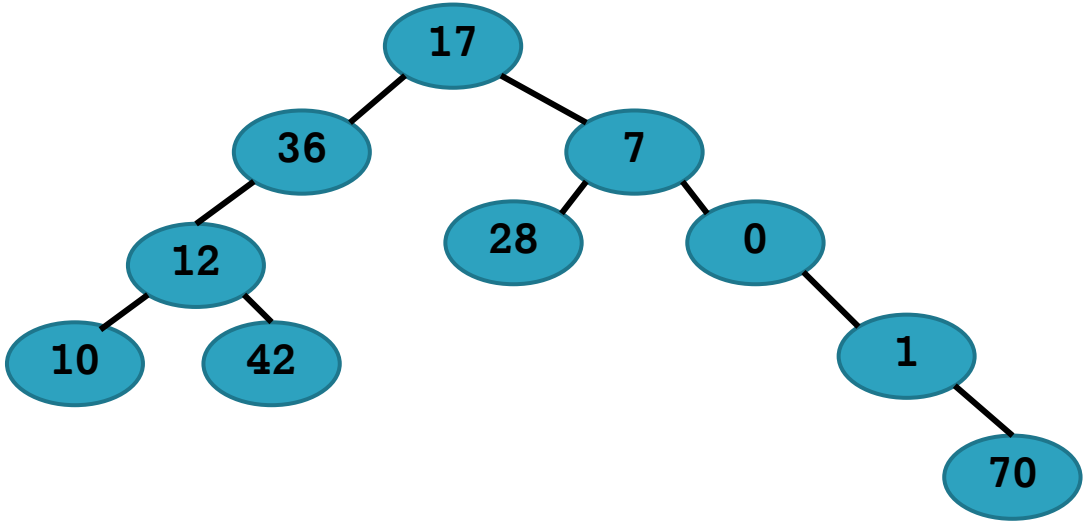
סופי - postorder




```
postorder(T)
  if not emptyTree(T)
        postorder (left(T))
        postorder (right(T))
        visit (root (T))
```

Post: 10, 42, 12, 36, 28,,70, 1, 0, 7, 17

מעבר עומק - depth



תחילי - preorder

תוכי - inorder

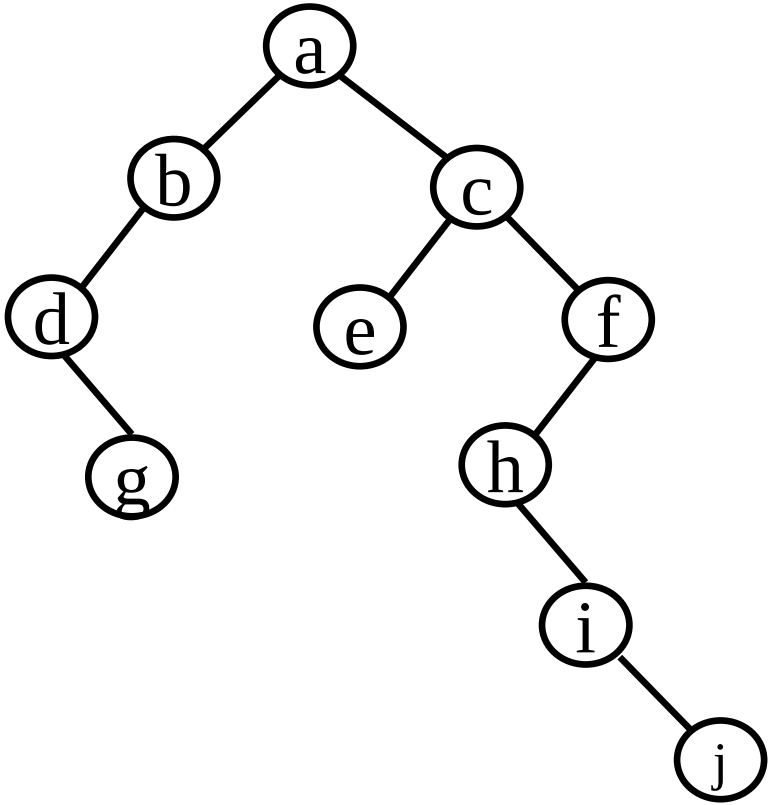
סופי - postorder

Pre: 17, 36, 12, 10, 42, 7, 28, 0, 1, 70

in: 10, 12, 42, 36, 17, 28, 7, 0, 1, 70

Post: 10, 42, 12, 36, 28,,70, 1, 0, 7, 17

תרגיל 1



עבור על העץ בשלשת השיטות

Pre: a, b, d, g, c, e, f, h, i, j

in: d, g, b, a, e, c, h, i, j, f

Post: g, d, b, e, j, i, h, f, c, a

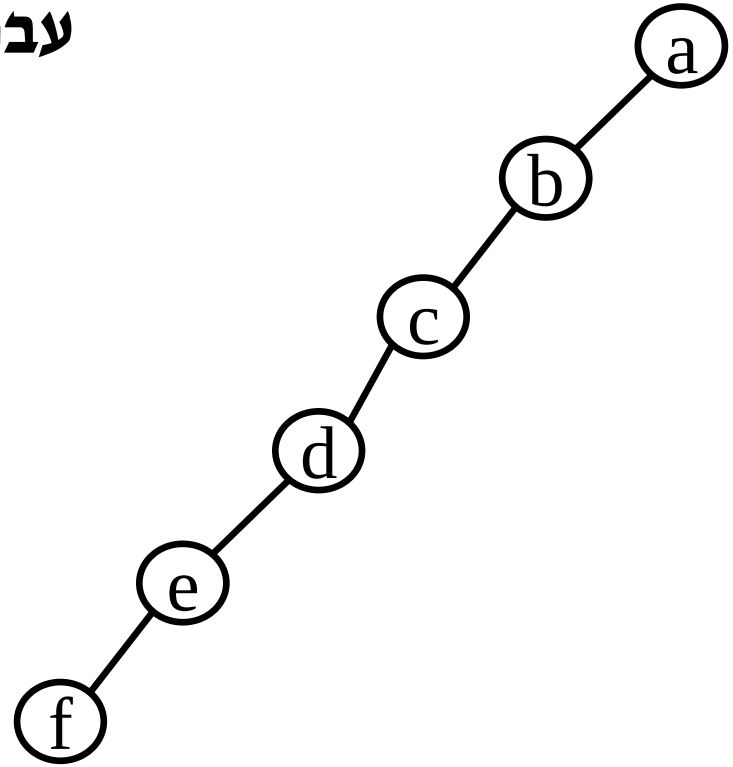
תרגיל 2

עבור על העץ בשלשת השיטות

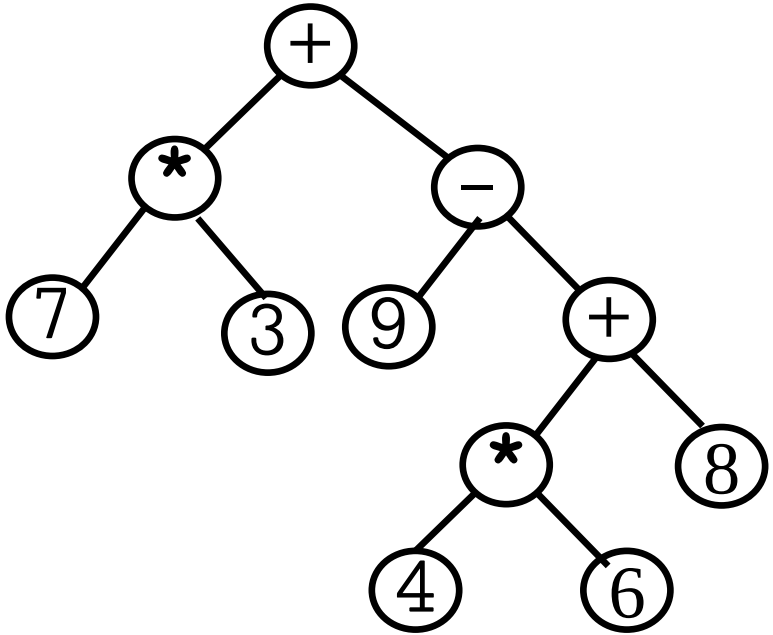
Pre: a, b, c, d, e, f

in: f, e, d, c, b, a

Post: f, e, d, c, b, a



תרגיל 3



עבור על העץ בשלשת השיטות

Pre: +, *, 7, 3, -, 9, +, *, 4, 6, 8

in: 7, *, 3, +, 9, -, 4, *, 6, +, 8

Post: 7, 3, *, 9, 4, 6, *, 8, +, -, +

7 * 3 + 9 - 4 * 6 + 8

(7 *3) + (9 -((4 *6) + 8))

תרגיל 4

כתוב אלגוריתם המקבל עץ T ומחזיר את מספר העלים בעץ T

```
countLeaves (T)
  If emptyTree(T) return 0
  If emptyTree(left(T)) && emptyTree(right(T)) return 1
  return countLeaves(left(T)) + countLeaves(right(T))
```

תרגיל 5

כתוב אלגוריתם המקבל עץ T ומחזיר את מספר הצמתים בעץ T

```
countNodes(T)
  If emptyTree(T) return 0
  return countNodes(left(T)) + countNodes(right(T)) + 1
```

תרגיל 6

כתוב אלגוריתם המקבל עץ T וערך x ומחזיר 'אמת' אם x נמצא בעץ T אחרת תחזיר 'שקר'

```
exist (T,x)
  If emptyTree(T) return false
  if key(T) == x return true
  return exist(left(T),x) || exist(right(T),x)
```

תרגיל 7

כתוב אלגוריתם המקבל עץ T ומחזיר 'אמת' אם קיימים בעץ T צמתים המכילים x ו-y, כך ש-y הוא צאצא של x אחרת תחזיר 'שקר'

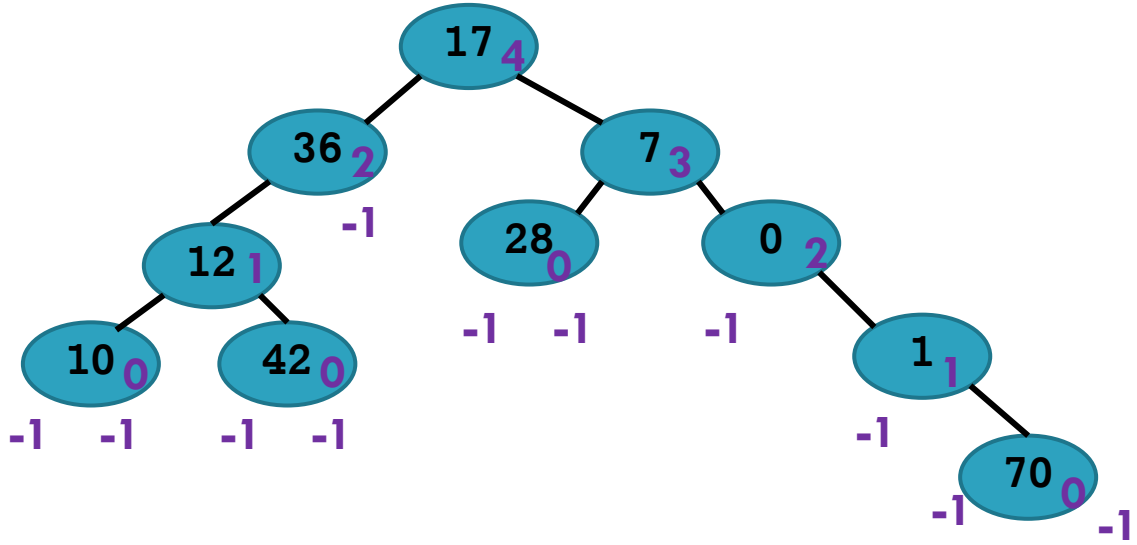
```
descendent (T,x,y)
  If emptyTree(T) return false
  if key(T) == x return exist(left(T),y) || exist(right(T),y)
  return descendent(left(T),x,y) || descendent(right(T),x,y)
```

תרגיל 8

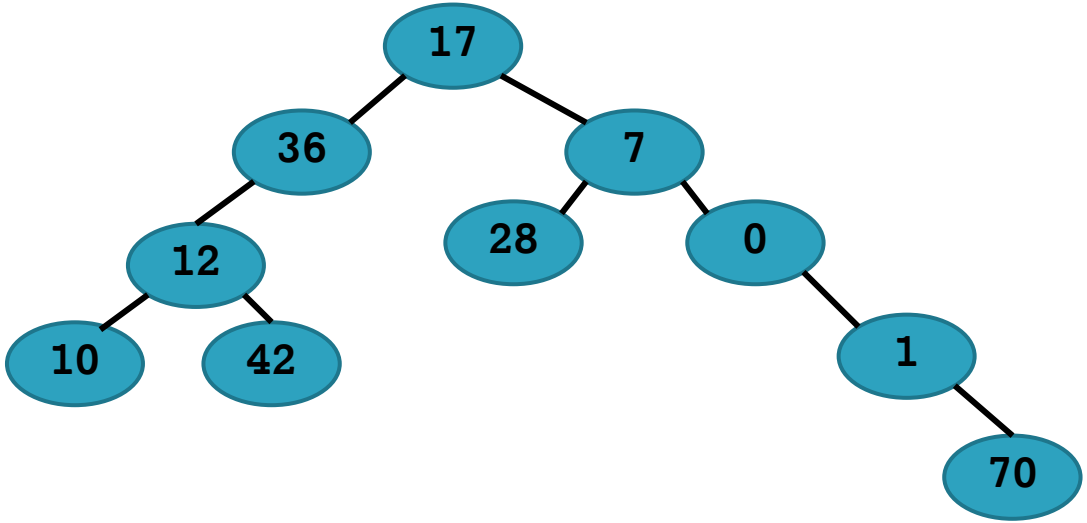
כתוב אלגוריתם המקבל עץ T ומחזיר את גובהו

height (T)

```
If emptyTree(T) return -1
return max( height(left(T)) , height (right(T)) ) + 1
```



מעבר רוחב - breadth



17, 36, 7, 12, 28, 0, 10, 42, 1, 70

למי יש רעיון?