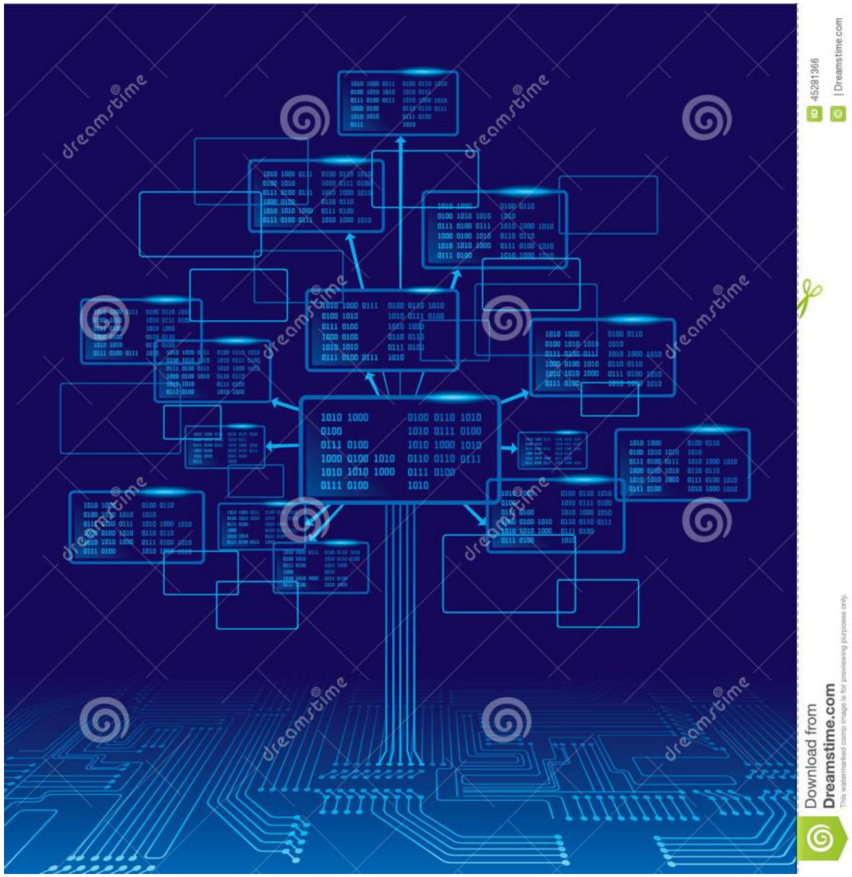
עצי חיפוש



ד"ר נעה לוינשטיין ©
עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

מילון dictionary

מבנה נתונים מילון

שומר אוסף איברים. לכל איבר מלבד האינפורמציה יש מפתח ייחודי.
על אוסף המפתחות מוגדר סדר ליניארי.

לדוגמא: מספרים טבעיים, מילים - סדר לקסיקוגרפי

עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

מילון dictionary

מבנה נתונים מילון

שומר אוסף איברים. לכל איבר מלבד האינפורמציה יש מפתח ייחודי.
על אוסף המפתחות מוגדר סדר ליניארי.

לדוגמא: מספרים טבעיים, מילים –סדר לקסיקוגרפי

הפעולות

- איתחול יוצר מילון ריק create-dictionary()
- הכנסת איבר מוסיף ל- D איבר שמפתחו k insert(k,D)
- הוצאת איבר מסיר מ- D איבר שמפתחו k delete(k,D)
- חיפוש find(k,D)

מחזיר מצביע לאיבר ב-D שמפתחו k או null אם לא נמצא.

מילון dictionary

פעולות נוספות

- עוקב successor(k,D)

מחזיר מצביע לאיבר ב-D שמפתחו עוקב ל- k (האיבר ב-D בעל המפתח הקטן ביותר שגדול מ-k) או null אם לא קיים כזה.

- קודם predecessor(k,D)

מחזיר מצביע לאיבר ב-D שמפתחו קודם ל- k או null אם לא קיים כזה.

- מינימום min(D)

מחזיר את המפתח המינימלי ב- D

- מקסימום max(D)

מחזיר את המפתח המכסימלי ב- D

מילון dictionary

פעולות נוספות

- שרשור concatenate(D1,D2)

ההנחה שכל המפתחות ב-D1 קטנים מהמפתחות ב-D2

- פיצול split(k,D)

מפצל את D ל- D1 שבו כל האיברים בעלי מפתח הקטן מ- k

וב- D2 כל האיברים בעלי מפתח הגדול מ- k.

אפשר לממש בעזרת רשימות מקושרות (ממוינות או לא ממוינות)

אפשר לממש בעזרת רשימות מקושרות (ממוינות או לא ממוינות)

לא יעיל !

סיבוכיות זמן של הפעולות: הכנסה, הוצאה, חיפוש

במקרה גרוע: $\Theta(n)$

במקרה ממוצע: $\Theta(n)$

בכל מקרה: $O(n)$

אפשר לממש בעזרת רשימות מקושרות (ממוינות או לא ממוינות)

לא יעיל !

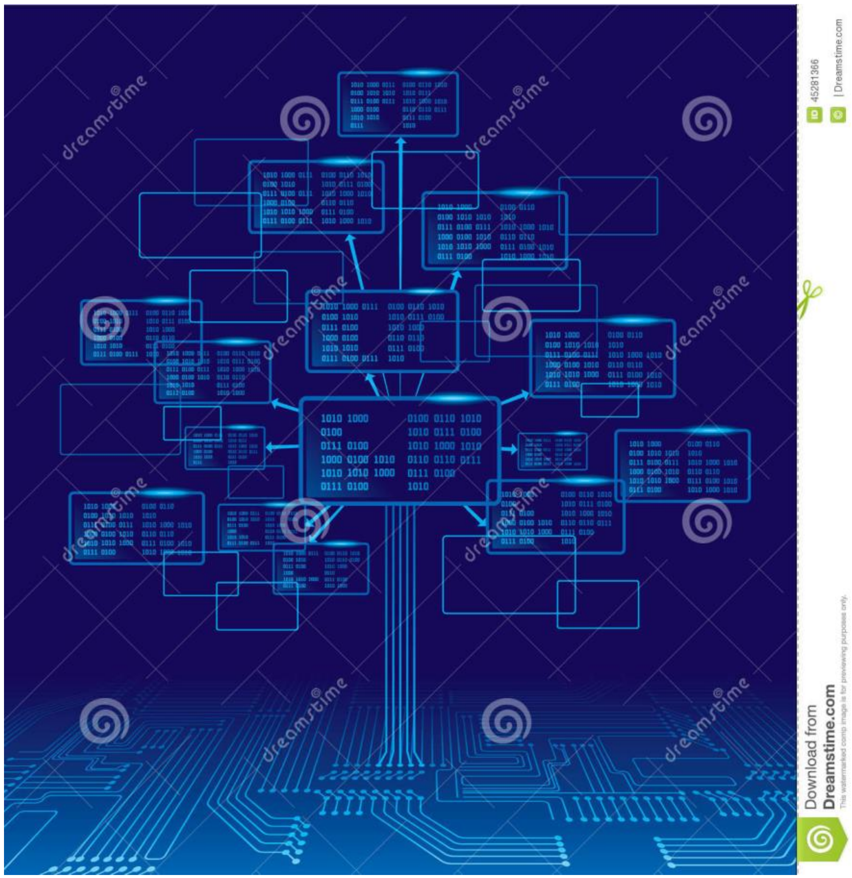
סיבוכיות זמן של הפעולות: הכנסה, הוצאה, חיפוש

במקרה גרוע: $\Theta(n)$

במקרה ממוצע: $\Theta(n)$

בכל מקרה: $O(n)$

מה לגבי מימוש בעזרת עצים??



עץ חיפוש בינארי

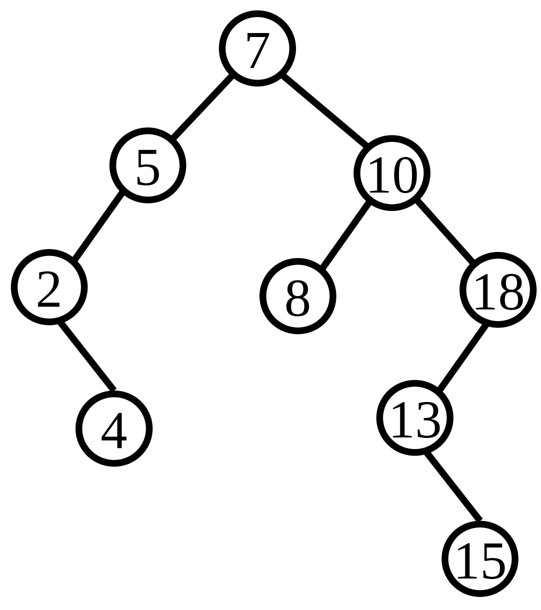
עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

עץ חיפוש בינארי

עץ חיפוש בינארי הוא עץ בינארי המקיים:

לכל צומת x,

- כל המפתחות בתת העץ השמאלי קטנים מהמפתח של x.
- כל המפתחות בתת העץ הימני גדולים מהמפתח של x.



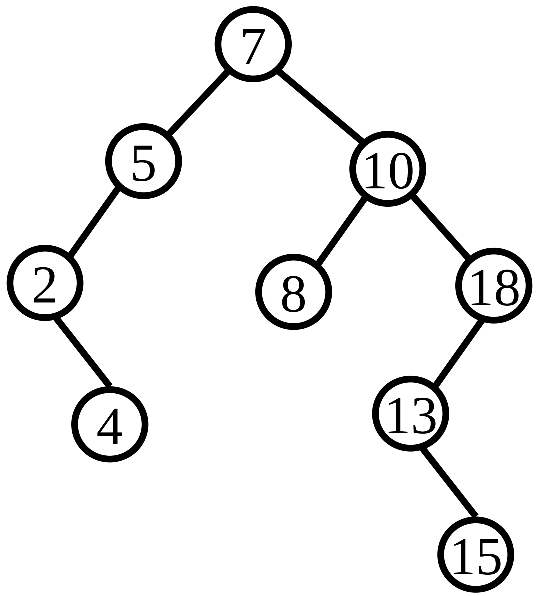
הנחה: המפתחות ייחודיים.

עץ חיפוש בינארי

עץ חיפוש בינארי הוא עץ בינארי המקיים:

לכל צומת x,

- כל המפתחות בתת העץ השמאלי קטנים מהמפתח של x.
- כל המפתחות בתת העץ הימני גדולים מהמפתח של x.



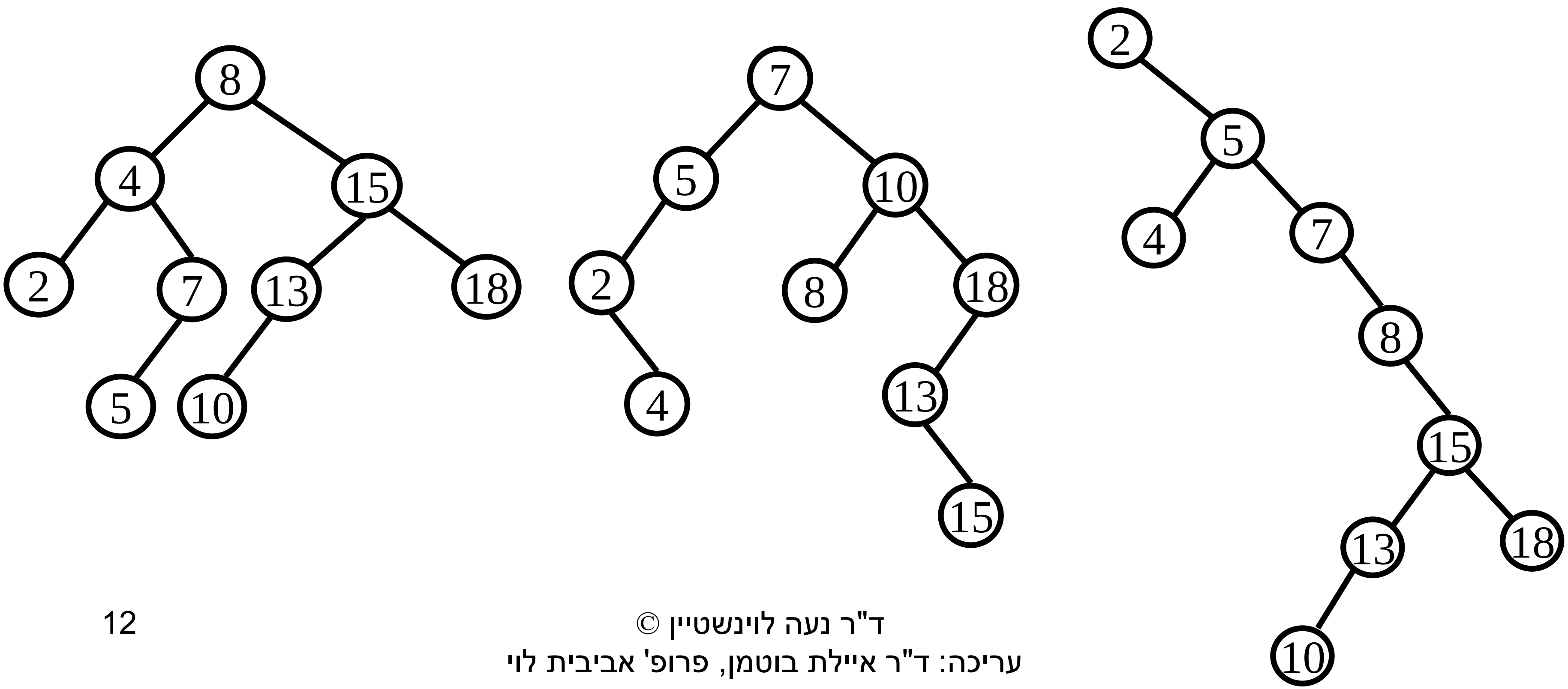
הנחה: המפתחות ייחודיים.

עץ חיפוש בינארי

עץ חיפוש בינארי הוא עץ בינארי המקיים:

לכל צומת x,

- כל המפתחות בתת העץ השמאלי קטנים מהמפתח של x.
- כל המפתחות בתת העץ הימני גדולים מהמפתח של x.



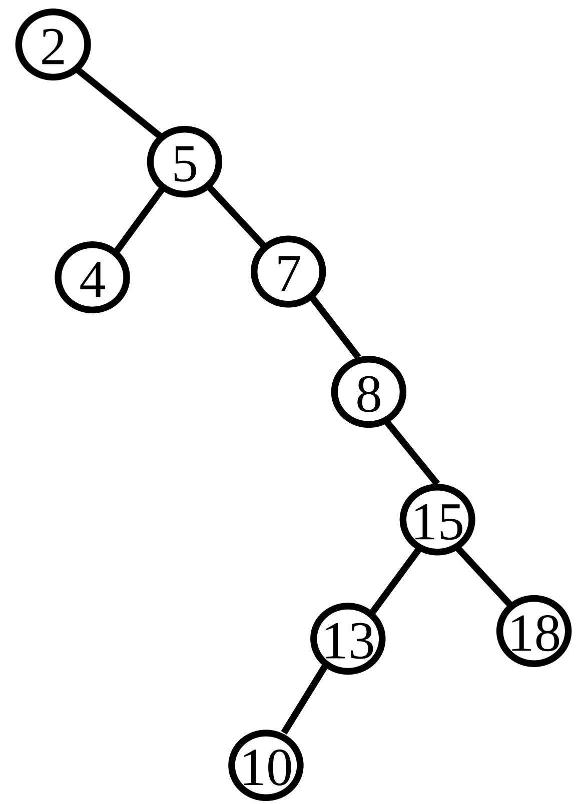
הנחה: המפתחות ייחודיים.

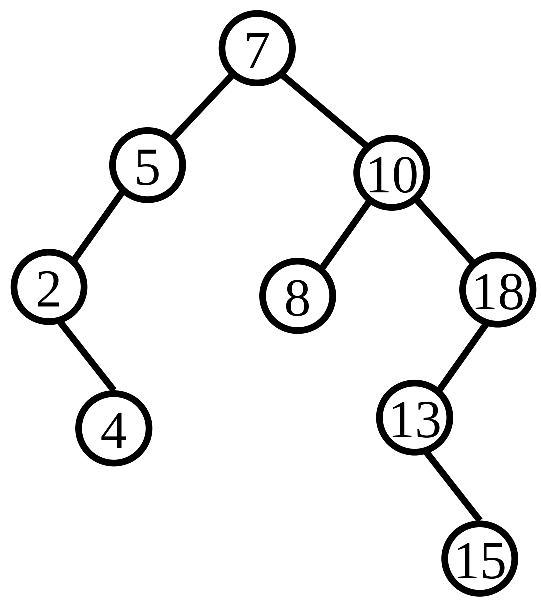
עץ חיפוש בינארי

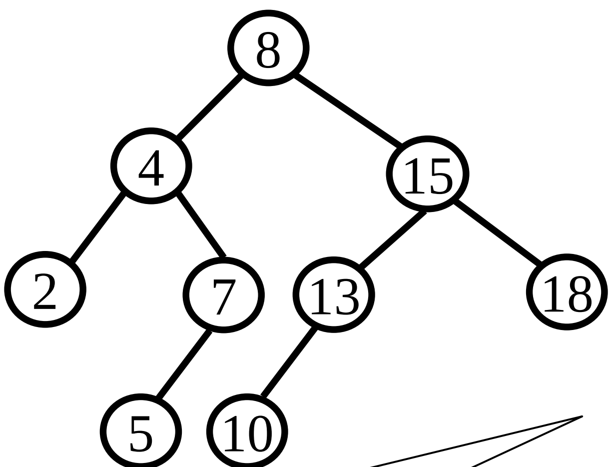
עץ חיפוש בינארי הוא עץ בינארי המקיים:

לכל צומת x,

- כל המפתחות בתת העץ השמאלי קטנים מהמפתח של x.
- כל המפתחות בתת העץ הימני גדולים מהמפתח של x.



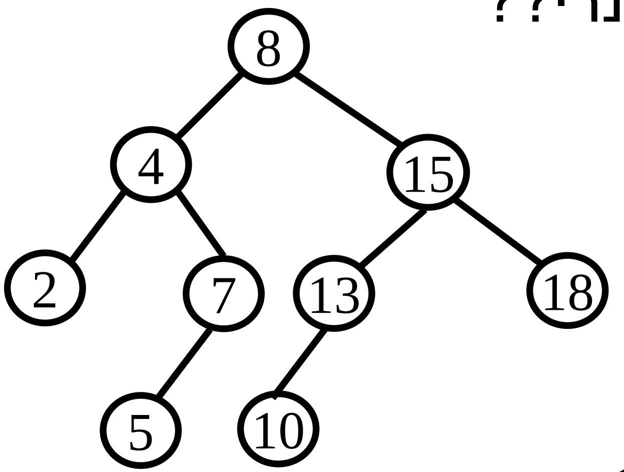


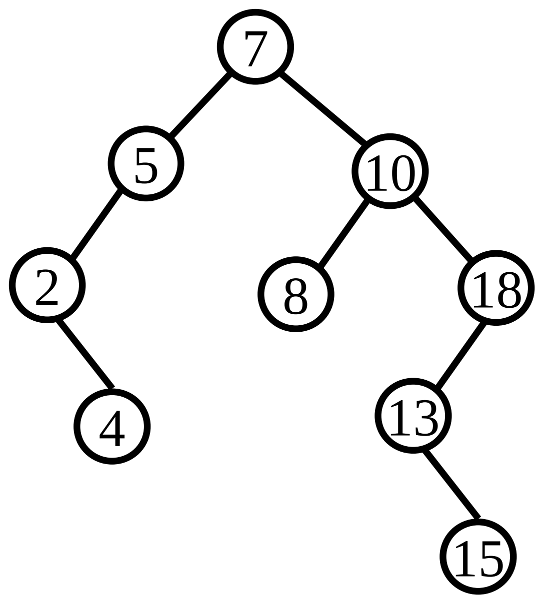


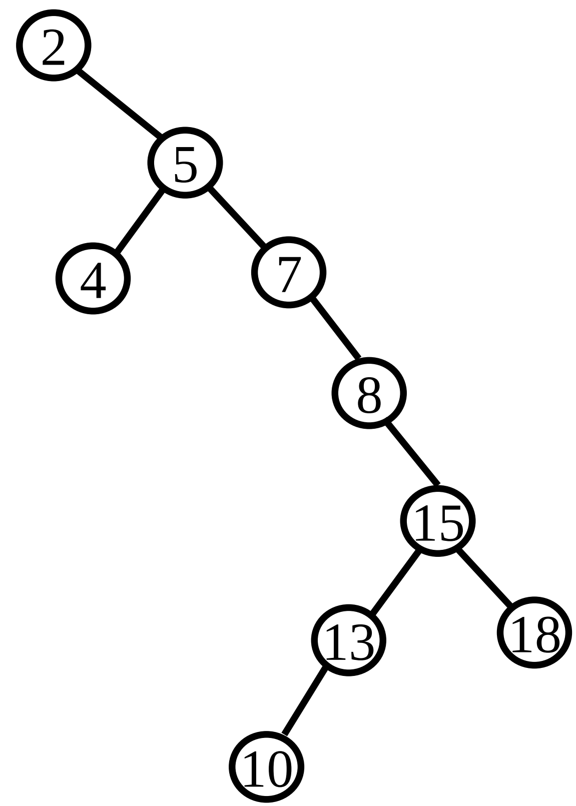
מה משותף ומה שונה בין עצי החיפוש ?

עץ חיפוש בינארי

מה נקבל בסריקת inorder של עץ חיפוש בינרי??







מימוש הפעולות

חיפוש

Search(k,T)

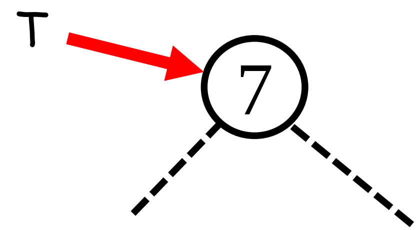
מחזיר מצביע לאיבר ב- T שמפתחו k או null אם לא נמצא.

מימוש הפעולות

חיפוש

Search(5,T)

מחזיר מצביע לאיבר ב- T שמפתחו k או null אם לא נמצא.

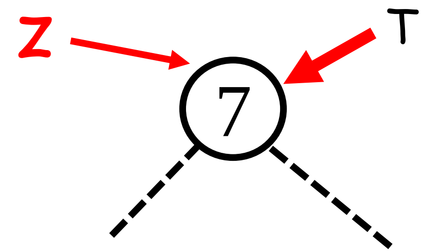


מימוש הפעולות

חיפוש

Search(5,T)

מחזיר מצביע לאיבר ב- T שמפתחו k או null אם לא נמצא.

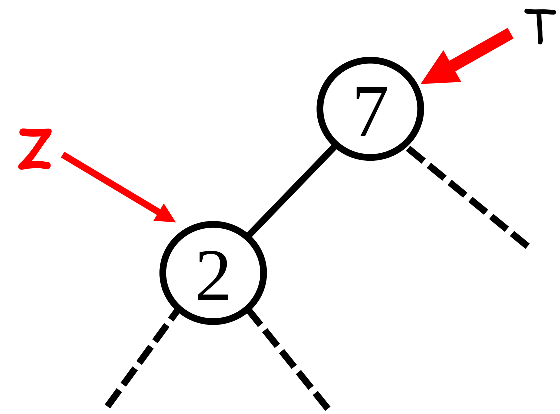


מימוש הפעולות

חיפוש

Search(5,T)

מחזיר מצביע לאיבר ב- T שמפתחו k או null אם לא נמצא.

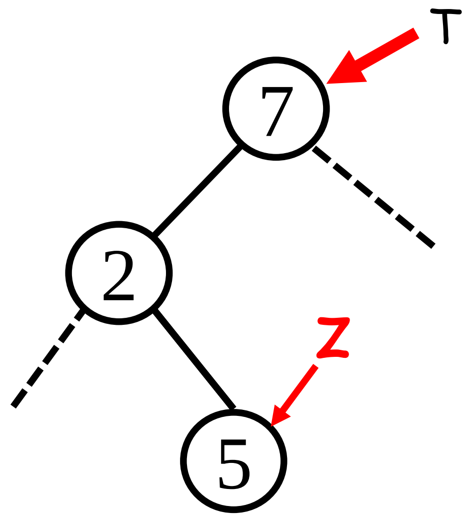


מימוש הפעולות

חיפוש

Search(5,T)

מחזיר מצביע לאיבר ב- T שמפתחו k או null אם לא נמצא.

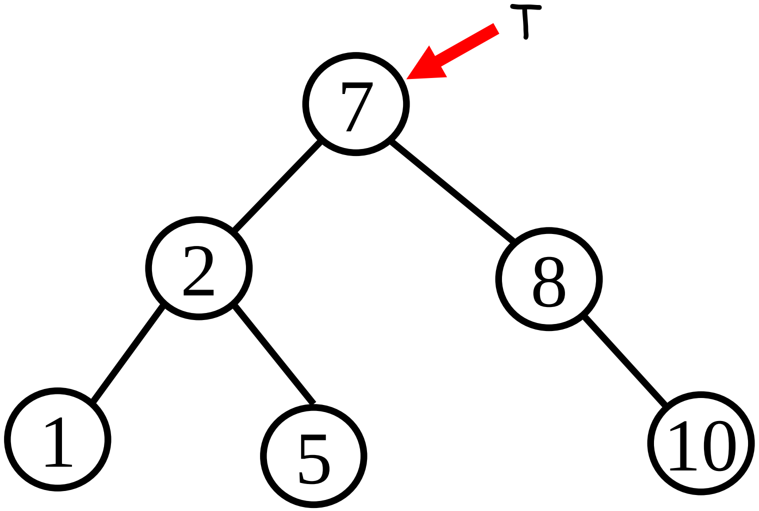


מימוש הפעולות

חיפוש

Search(k,T)

מחזיר מצביע לאיבר ב- T שמפתחו k או null אם לא נמצא.

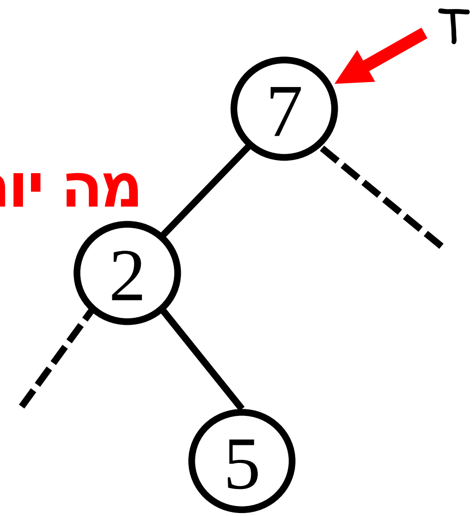


מימוש הפעולות

חיפוש

Search(6,T)

מחזיר מצביע לאיבר ב- T שמפתחו k או null אם לא נמצא.



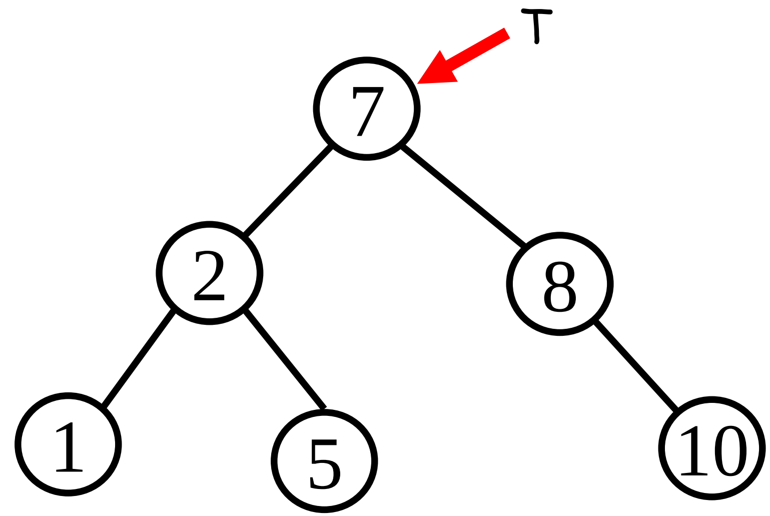
מה יוחזר אם מחפשים ערך שאינו נמצא בעץ ?

עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

מימוש הפעולות

חיפוש קודם או עוקב

מחזיר מצביע לאיבר ב- T שמפתחו k ואם k לא נמצא - **העוקב או הקודם**.

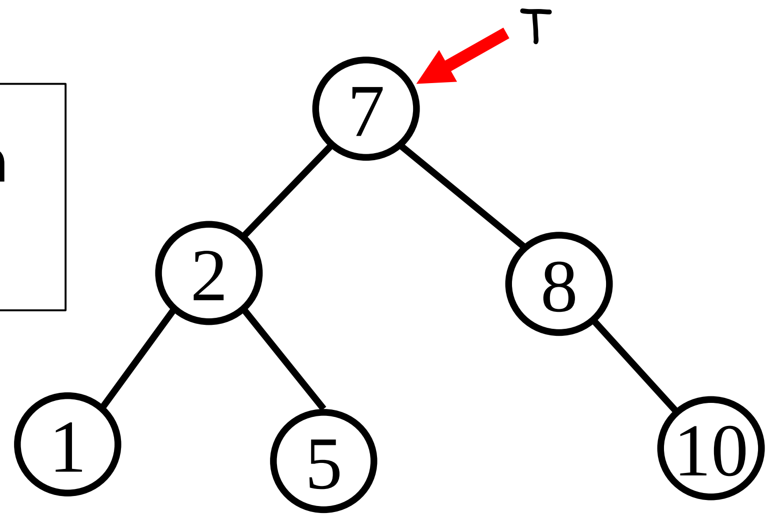


מימוש הפעולות

חיפוש קודם או עוקב

מחזיר מצביע לאיבר ב- T שמפתחו k ואם k לא נמצא - **העוקב או הקודם**.

האיבר הקטן ביותר הגדול מ-k

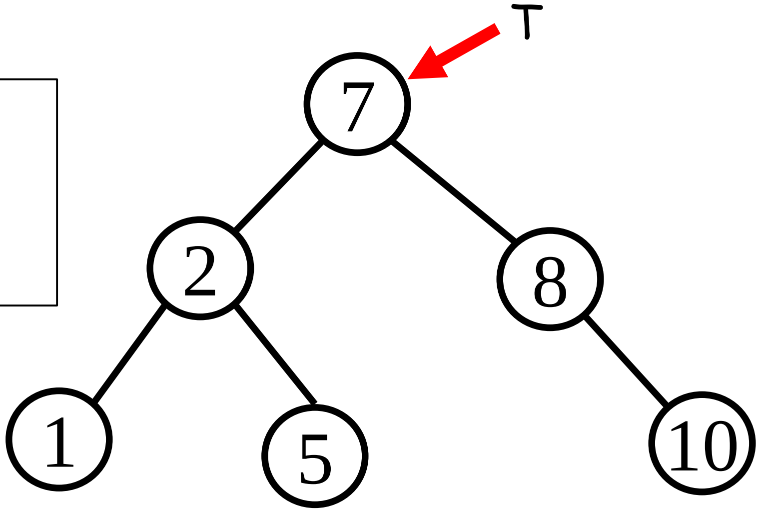


מימוש הפעולות

חיפוש קודם או עוקב

מחזיר מצביע לאיבר ב- T שמפתחו k ואם k לא נמצא - **העוקב או הקודם**.

האיבר הקטן ביותר הגדול מ-k
העוקב של 6 הוא 7

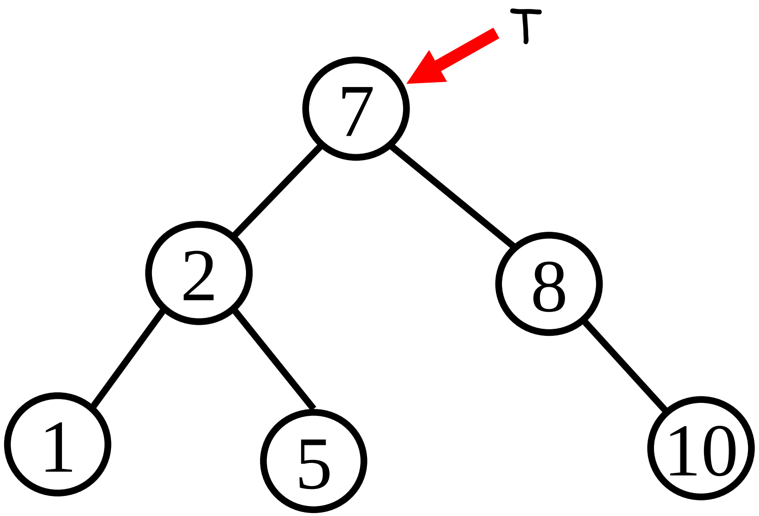


מימוש הפעולות

חיפוש קודם או עוקב

מחזיר מצביע לאיבר ב- T שמפתחו k ואם k לא נמצא - **העוקב או הקודם**.

האיבר הגדול ביותר הקטן מ-k

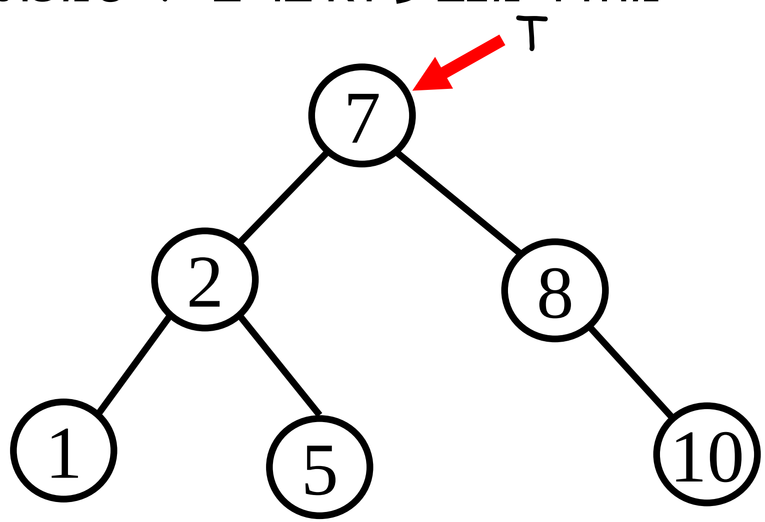


מימוש הפעולות

חיפוש קודם או עוקב

מחזיר מצביע לאיבר ב- T שמפתחו k ואם k לא נמצא - **העוקב או הקודם**.

האיבר הגדול ביותר הקטן מ-k
הקודם של 6 הוא 5

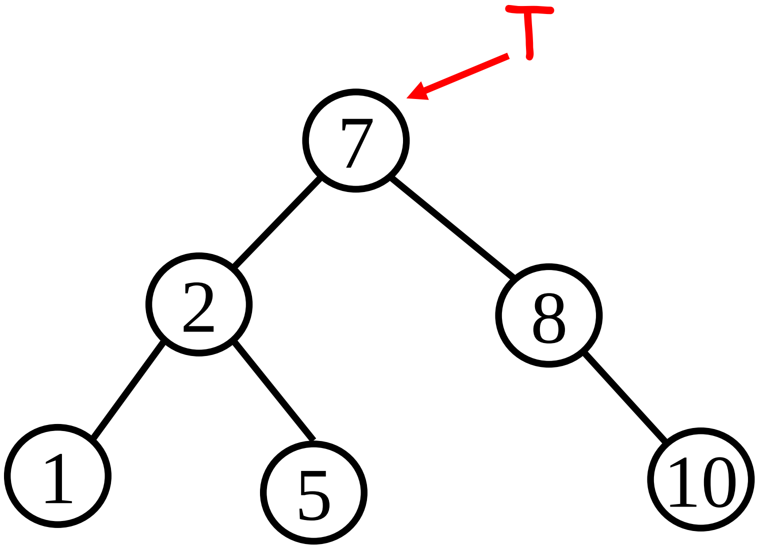


הכנסה

insert(k,T)

מוסיף לעץ T איבר שמפתחו k

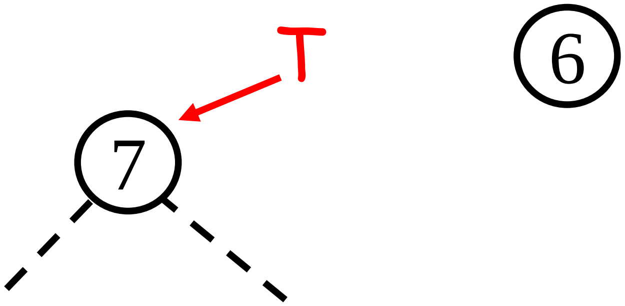
נתעלם מאינפורמציה נוספת שהאיבר מכיל



insert(k,T)

הכנסה

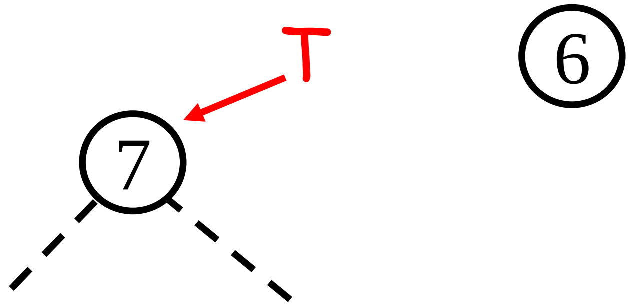
מוסיף לעץ T איבר שמפתחו k
נתעלם מאינפורמציה נוספת שהאיבר מכיל



הכנסה

insert(k,T)

מוסיף לעץ T איבר שמפתחו k
נתעלם מאינפורמציה נוספת שהאיבר מכיל



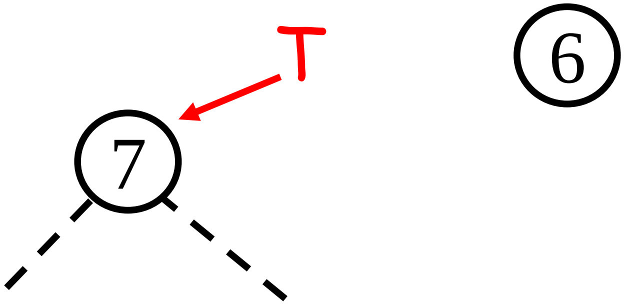
מה השלב הראשון ?

הכנסה

insert(k,T)

מוסיף לעץ T איבר שמפתחו k

נתעלם מאינפורמציה נוספת שהאיבר מכיל



מה השלב הראשון ?

חיפוש!

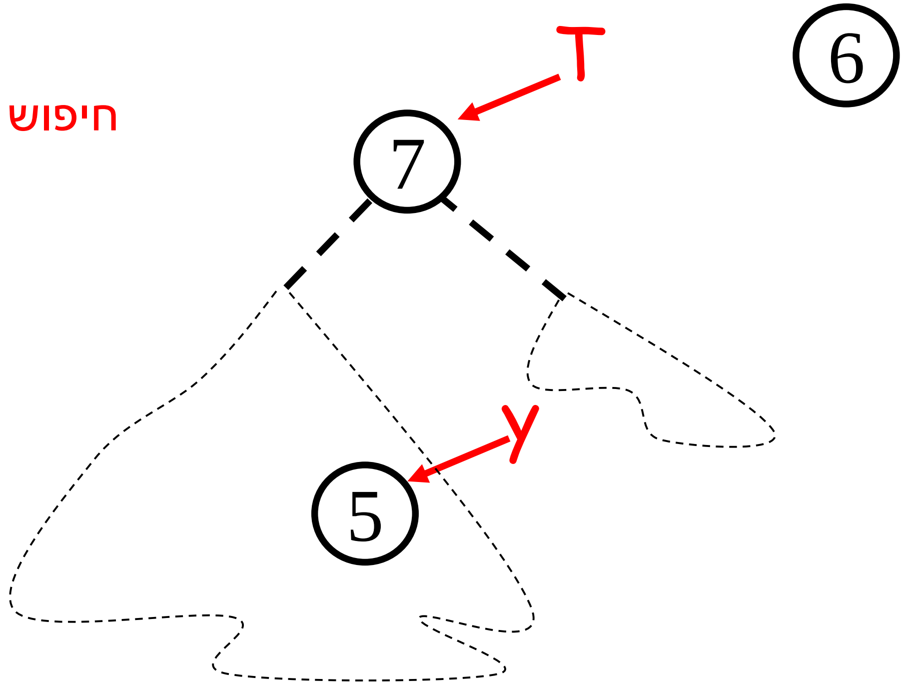
איזה סוג חיפוש ?

הכנסה

insert(k,T)

מוסיף לעץ T איבר שמפתחו k
נתעלם מאינפורמציה נוספת שהאיבר מכיל

חיפוש עם מציאת העוקב או הקודם



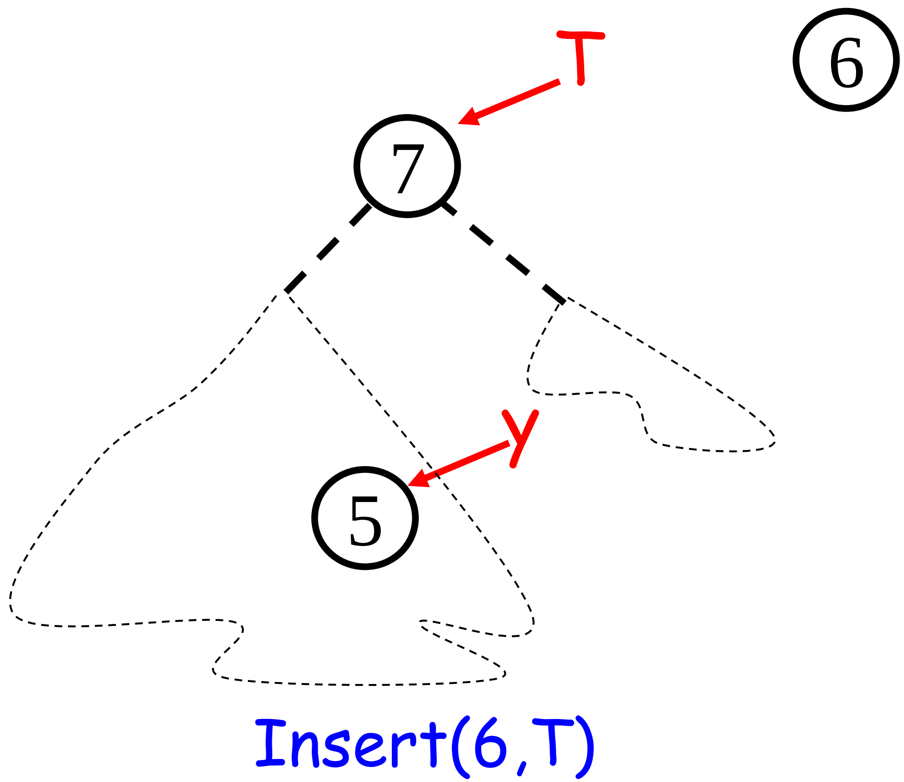
Insert(6,T)

הכנסה

insert(k,T)

מוסיף לעץ T איבר שמפתחו k
נתעלם מאינפורמציה נוספת שהאיבר מכיל

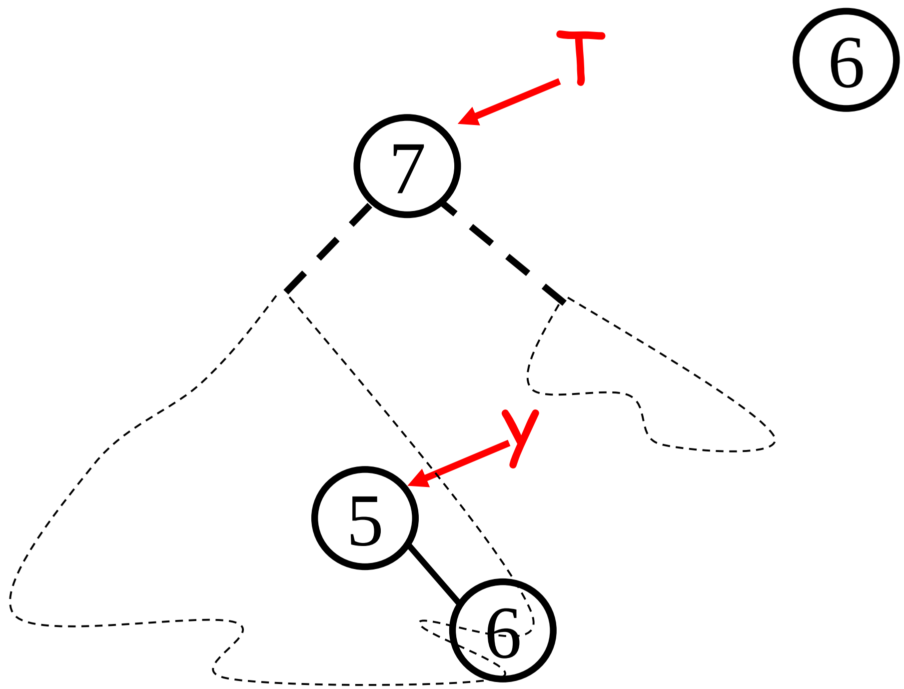
ומה השלב הבא?



insert(k,T)

הכנסה

מוסיף לעץ T איבר שמפתחו k
נתעלם מאינפורמציה נוספת שהאיבר מכיל



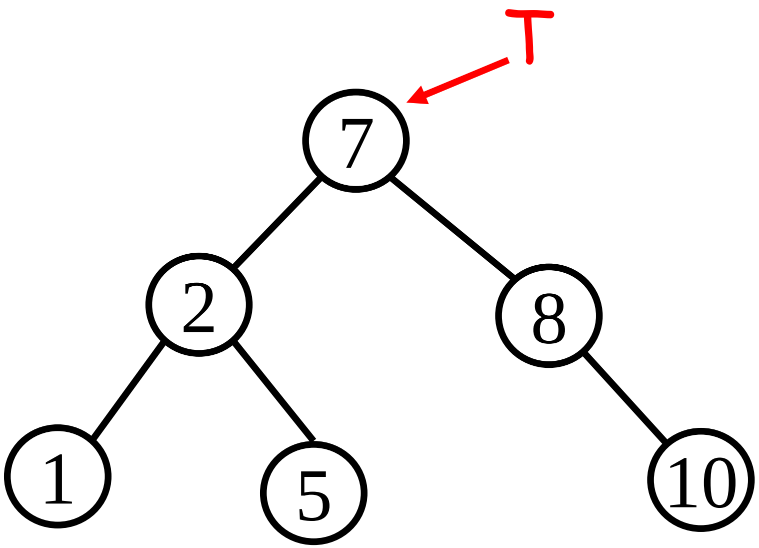
Insert(6,T)

insert(k,T)

הכנסה

מוסיף לעץ T איבר שמפתחו k

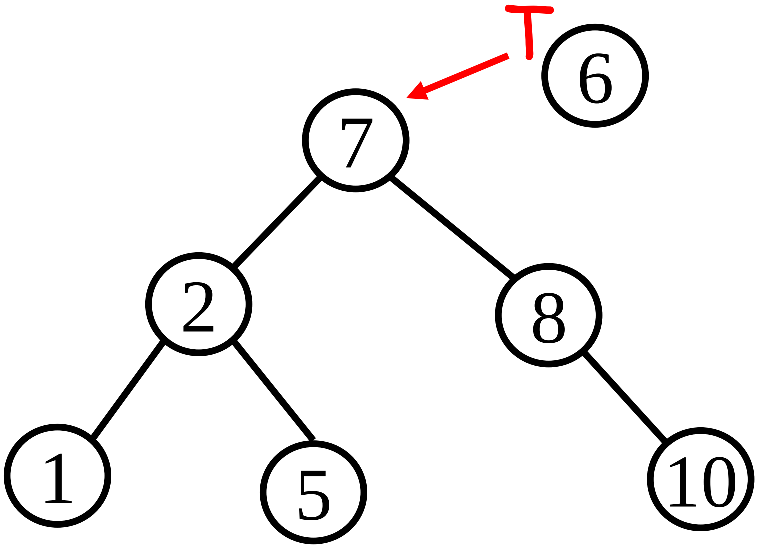
נתעלם מאינפורמציה נוספת שהאיבר מכיל



insert(k,T)

הכנסה

מוסיף לעץ T איבר שמפתחו k
נתעלם מאינפורמציה נוספת שהאיבר מכיל

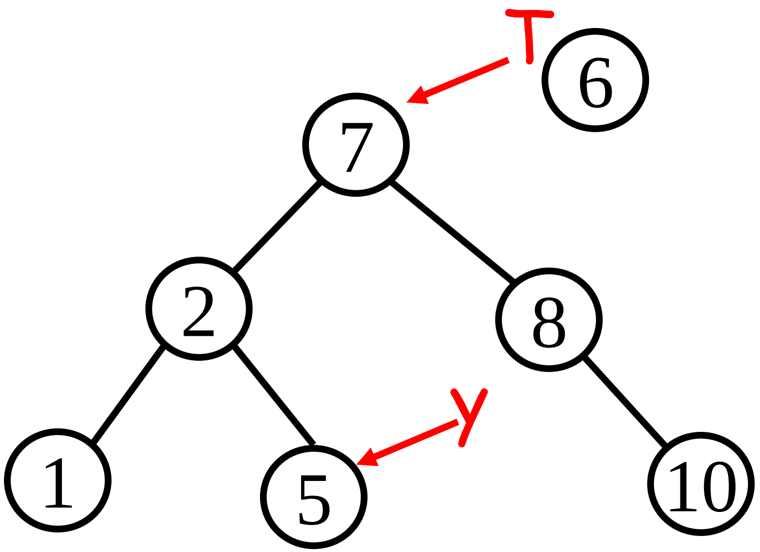


Insert(6,T)

insert(k,T)

הכנסה

מוסיף לעץ T איבר שמפתחו k
נתעלם מאינפורמציה נוספת שהאיבר מכיל



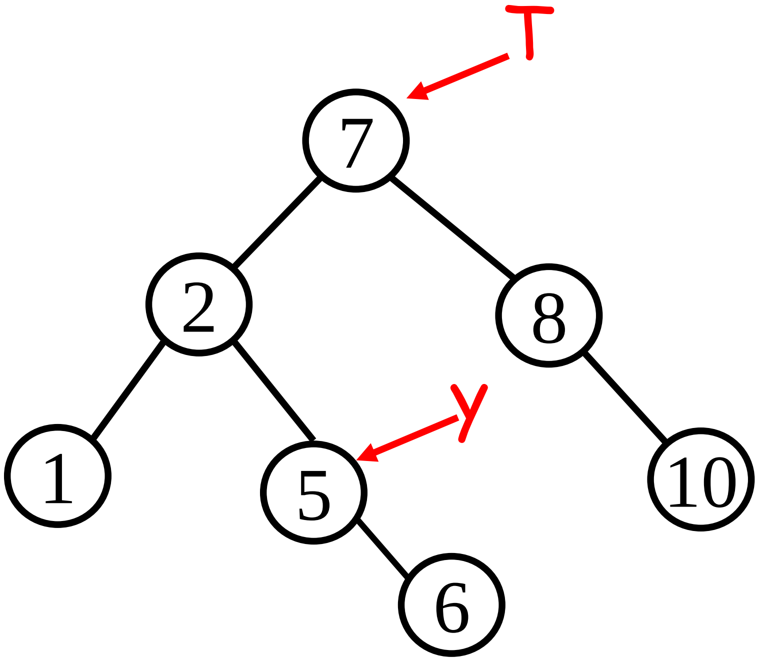
Insert(6,T)

הכנסה

insert(k,T)

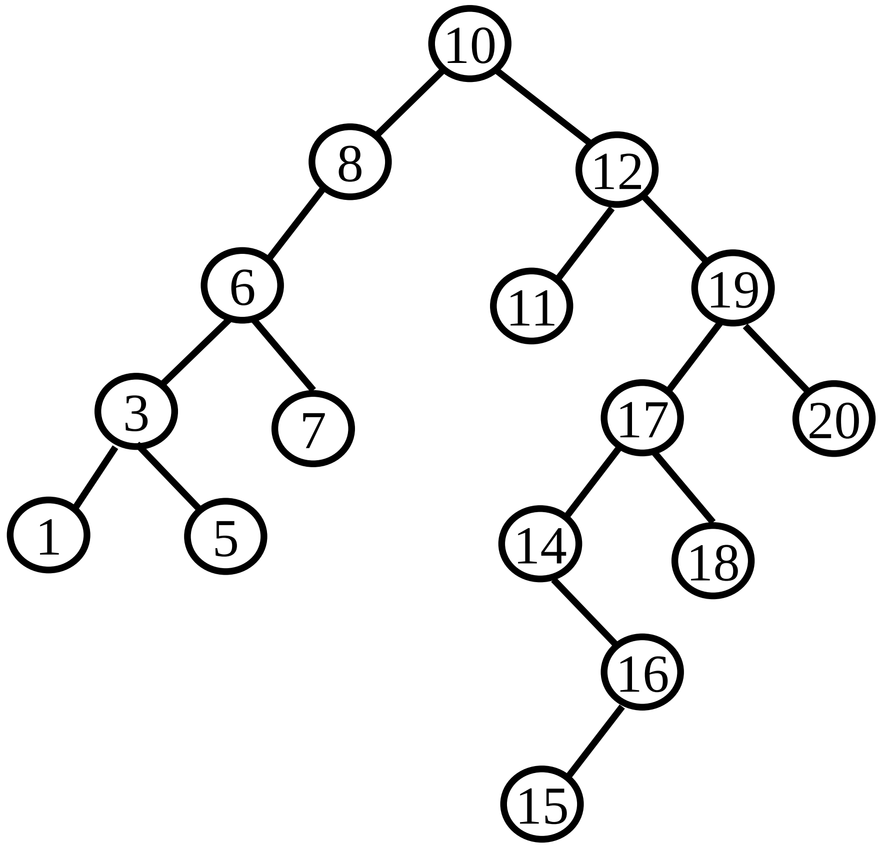
מוסיף לעץ T איבר שמפתחו k

נתעלם מאינפורמציה נוספת שהאיבר מכיל

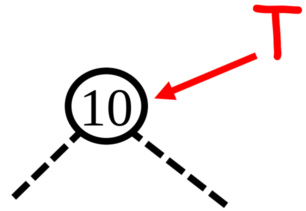


Insert(6,T)

delete(5,T)

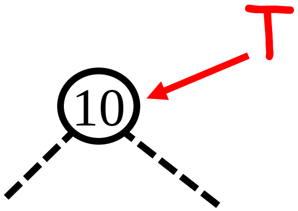


delete(5,T)



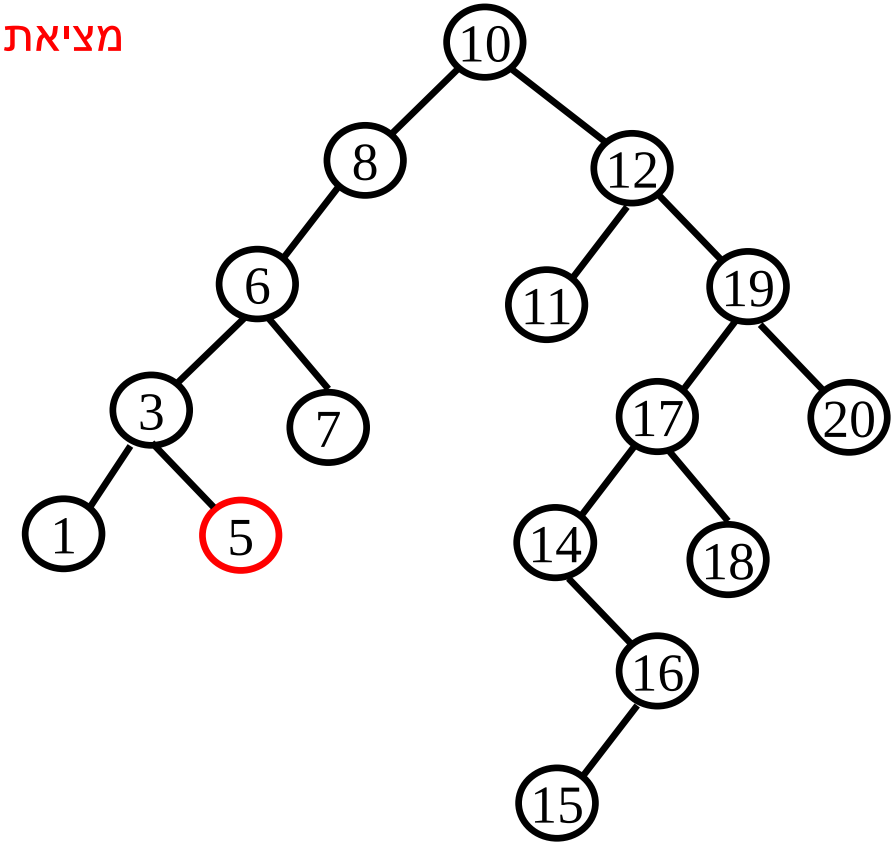
delete(5,T)

מה השלב הראשון ?

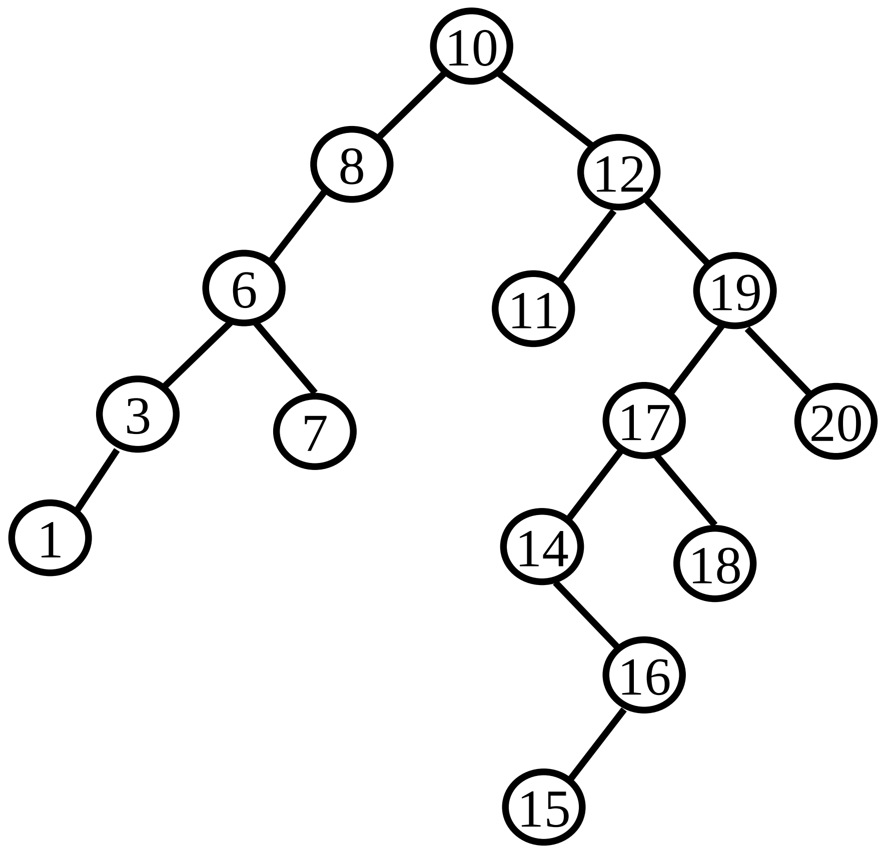


delete(5,T)

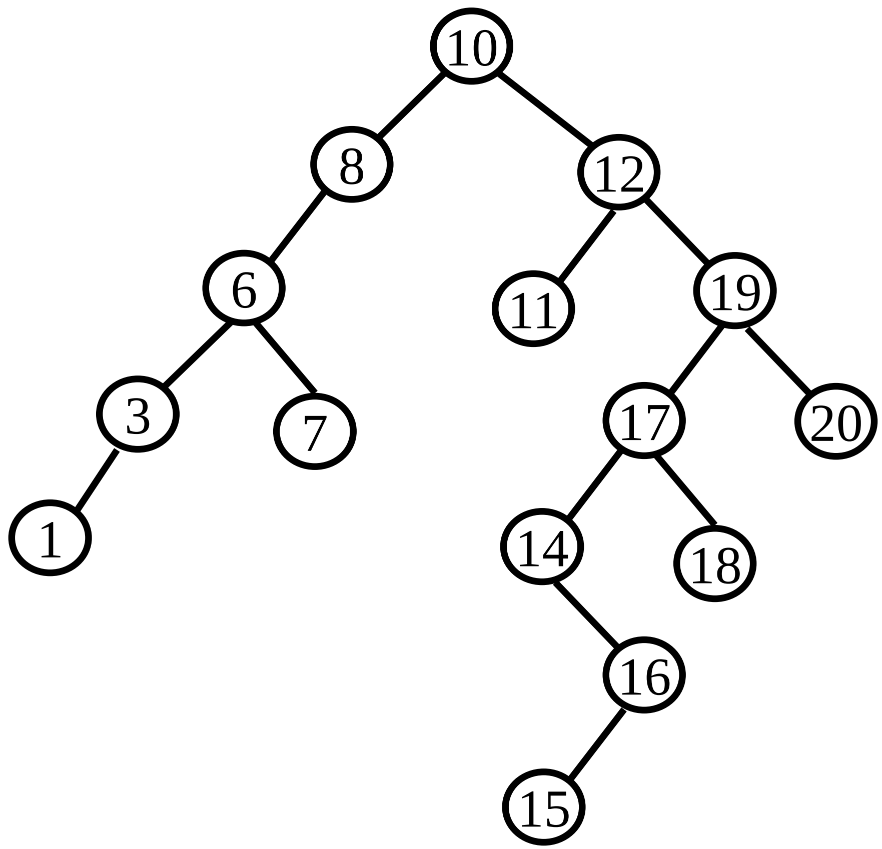
מציאת האיבר



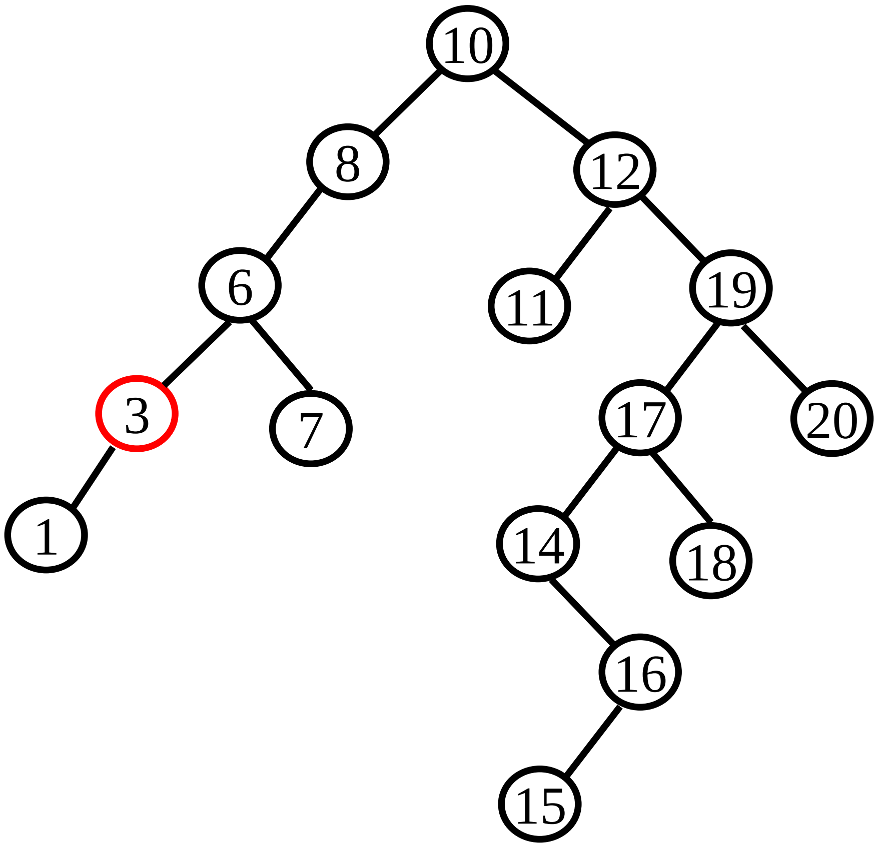
delete(5,T)



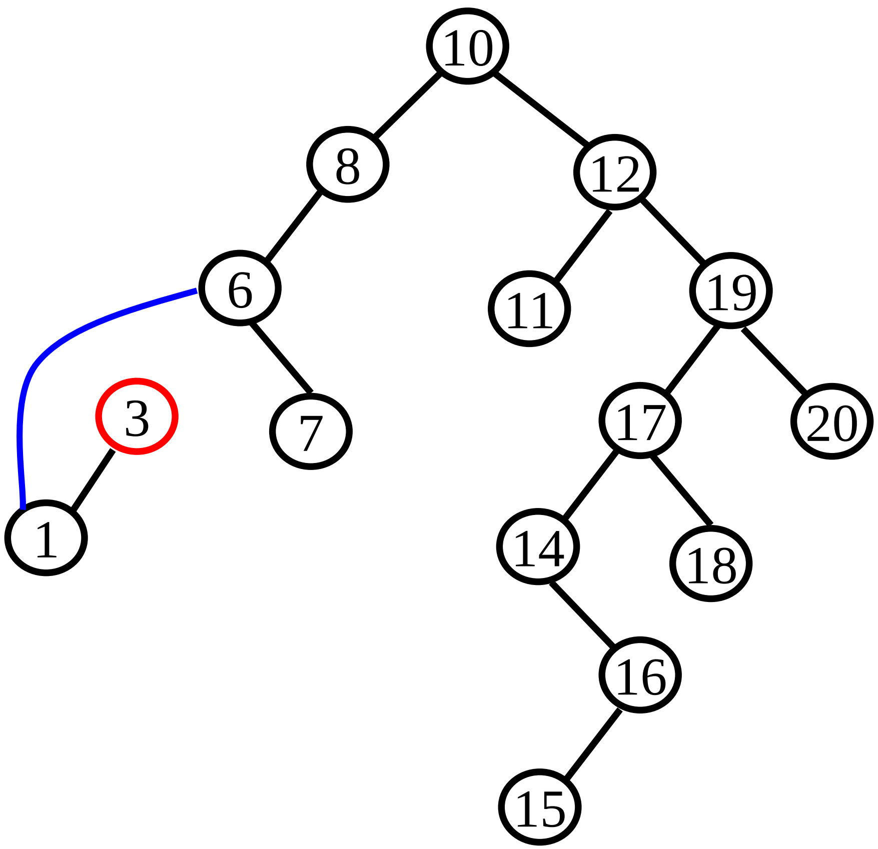
delete(3,T)



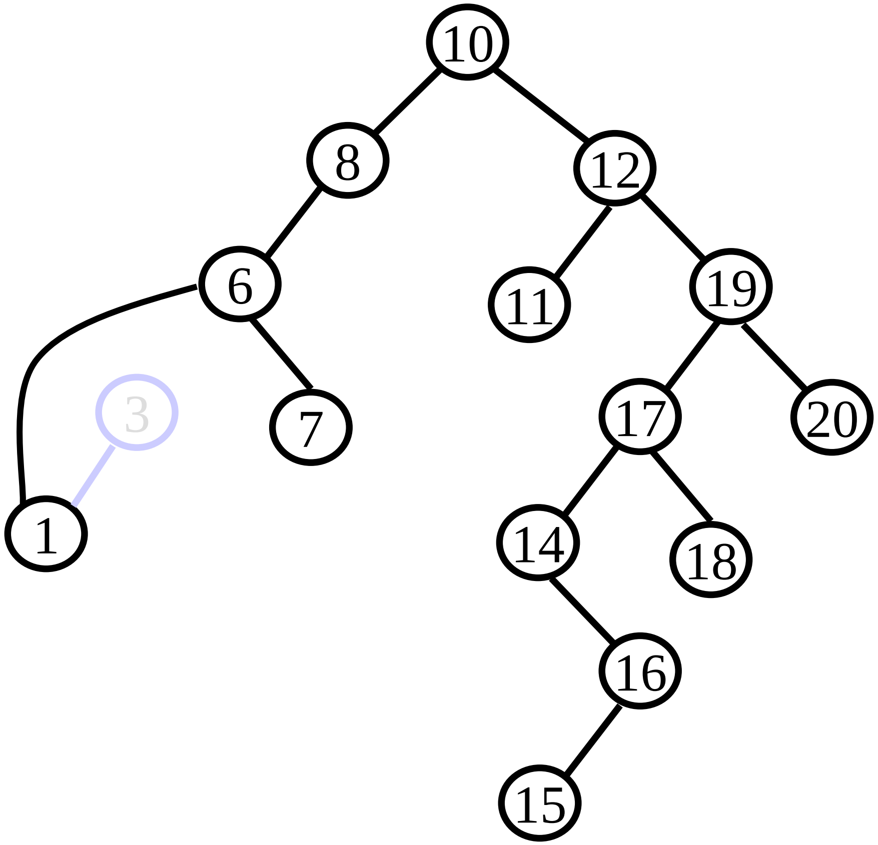
delete(3,T)



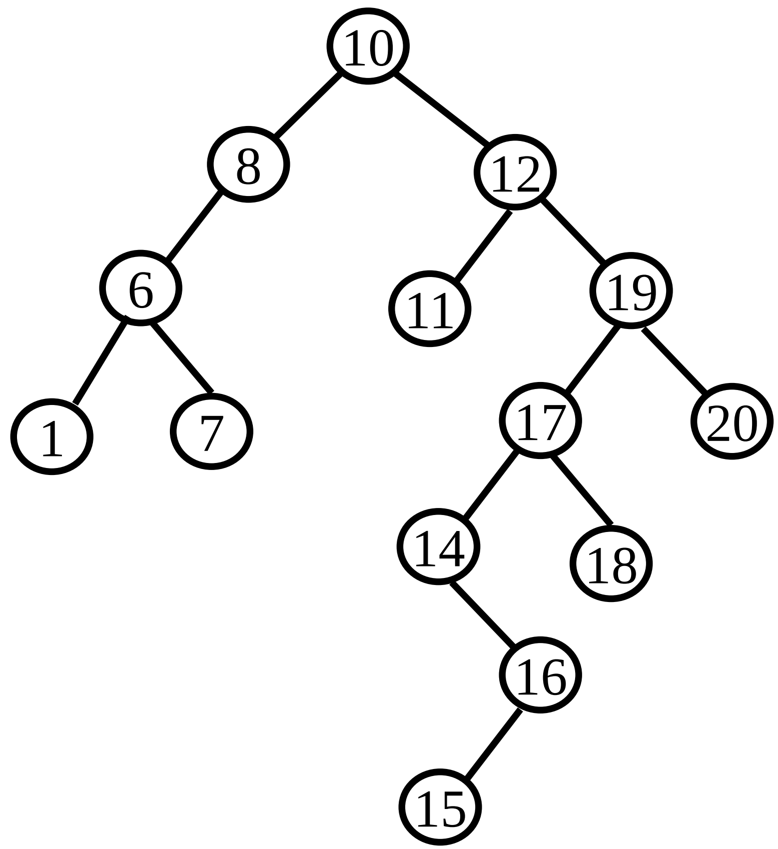
delete(3,T)



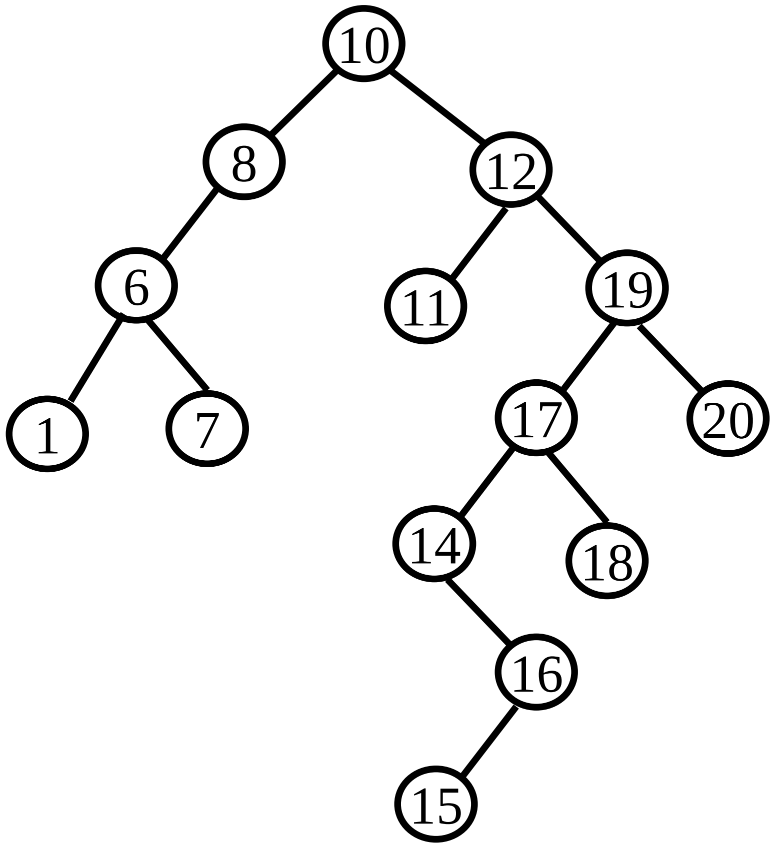
delete(3,T)



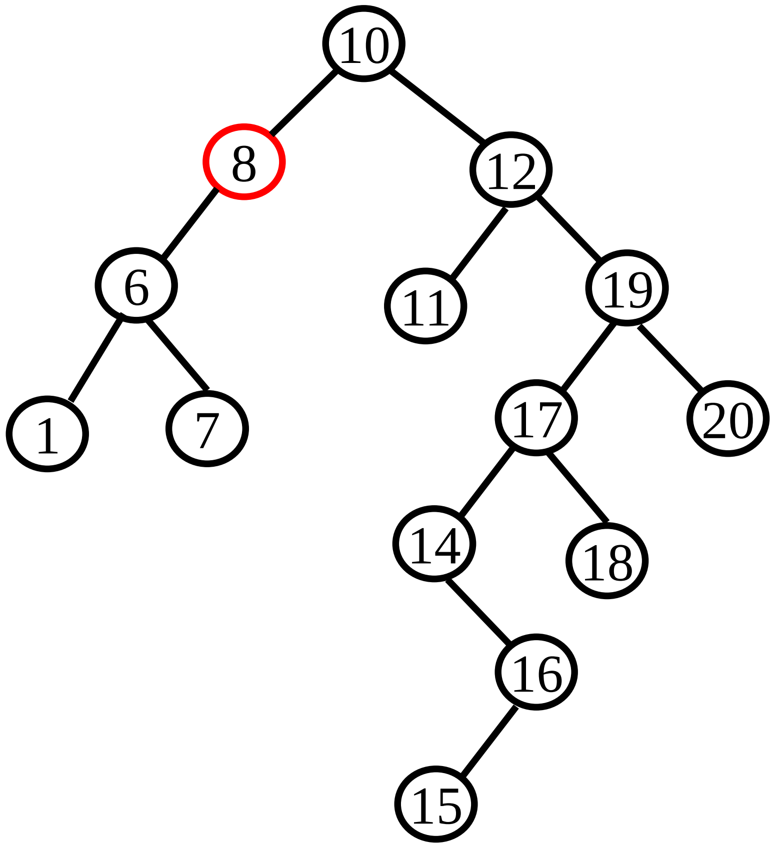
delete(3,T)



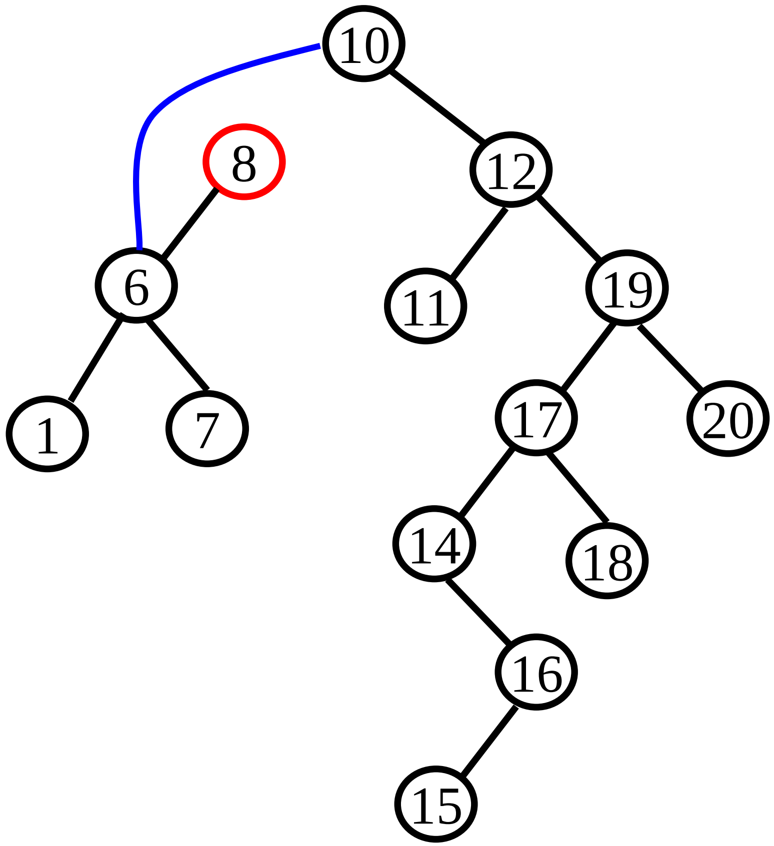
delete(8,T)



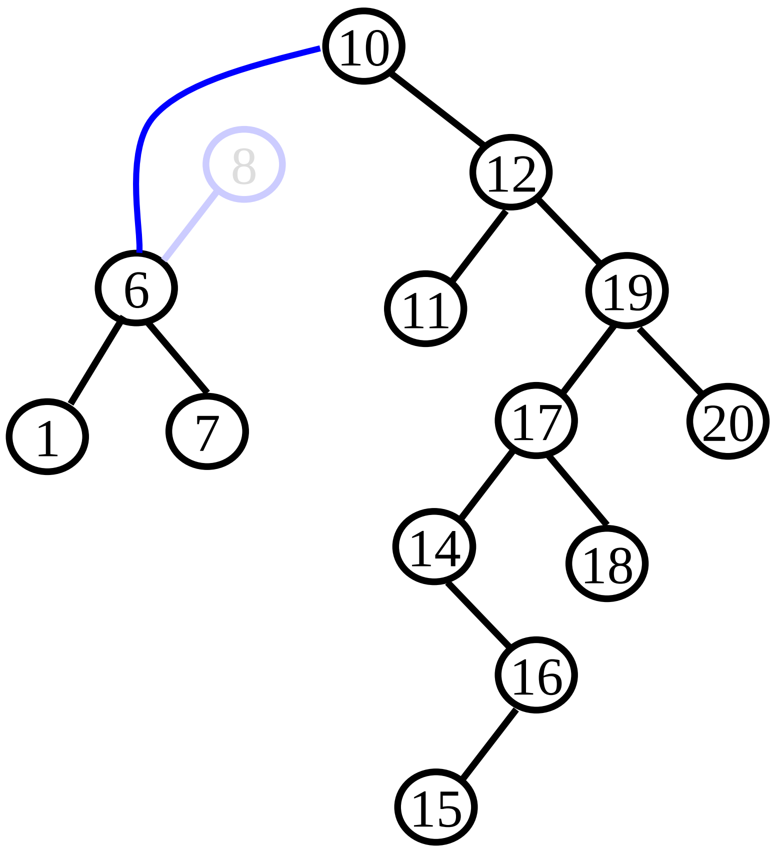
delete(8,T)



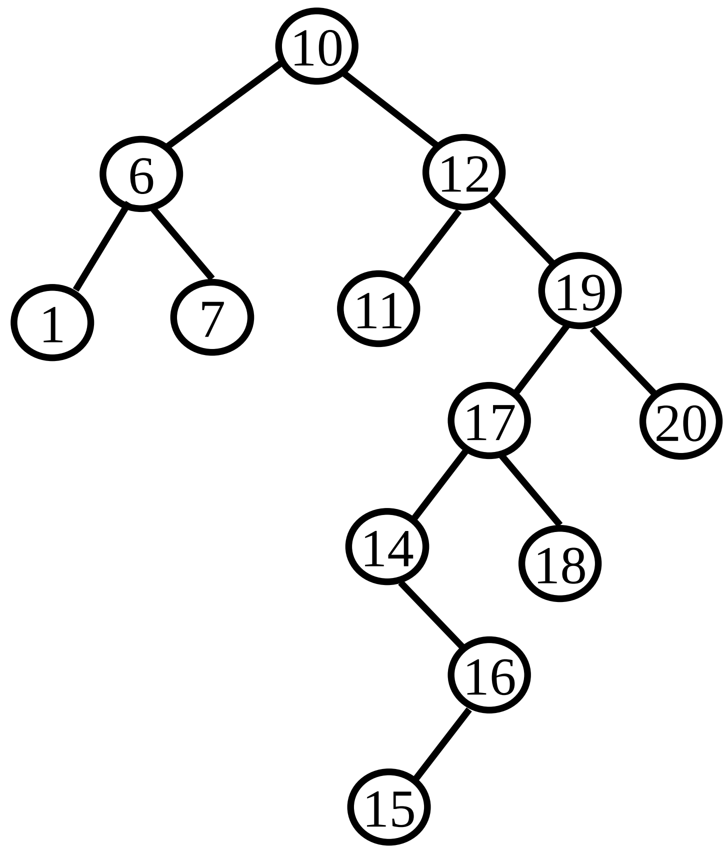
delete(8,T)



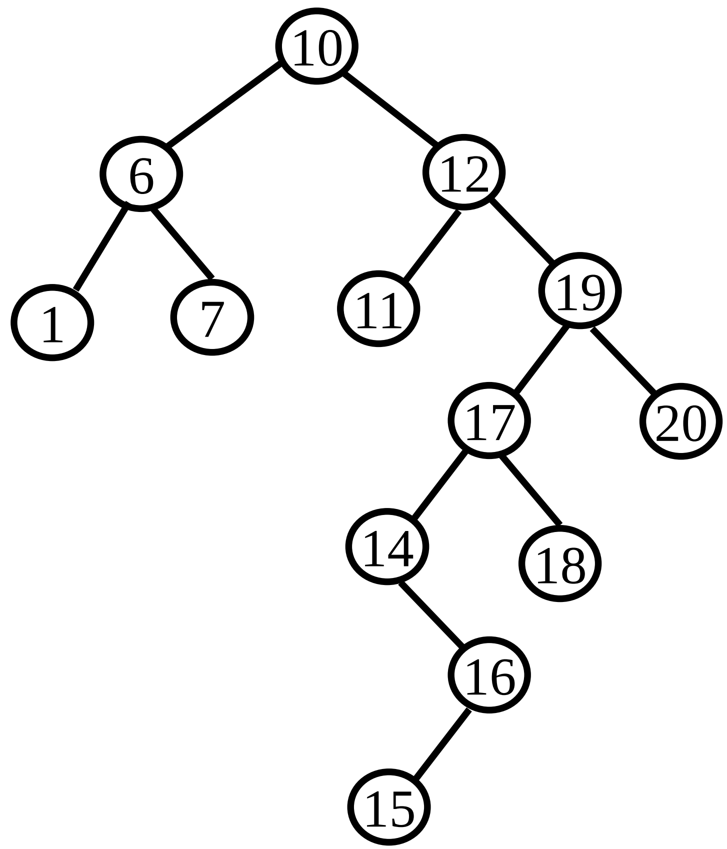
delete(8,T)



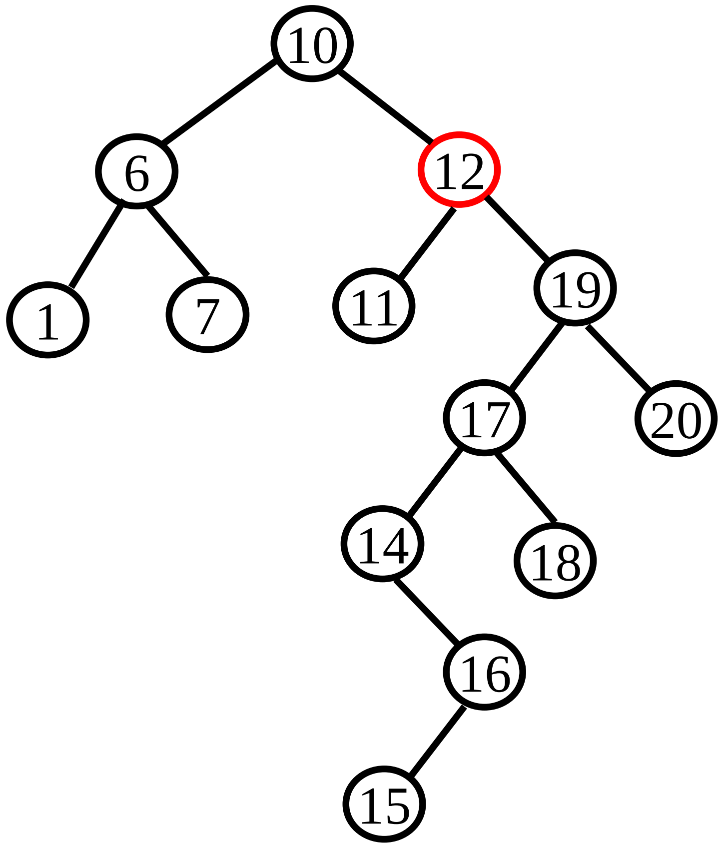
delete(8,T)



delete(12,T)

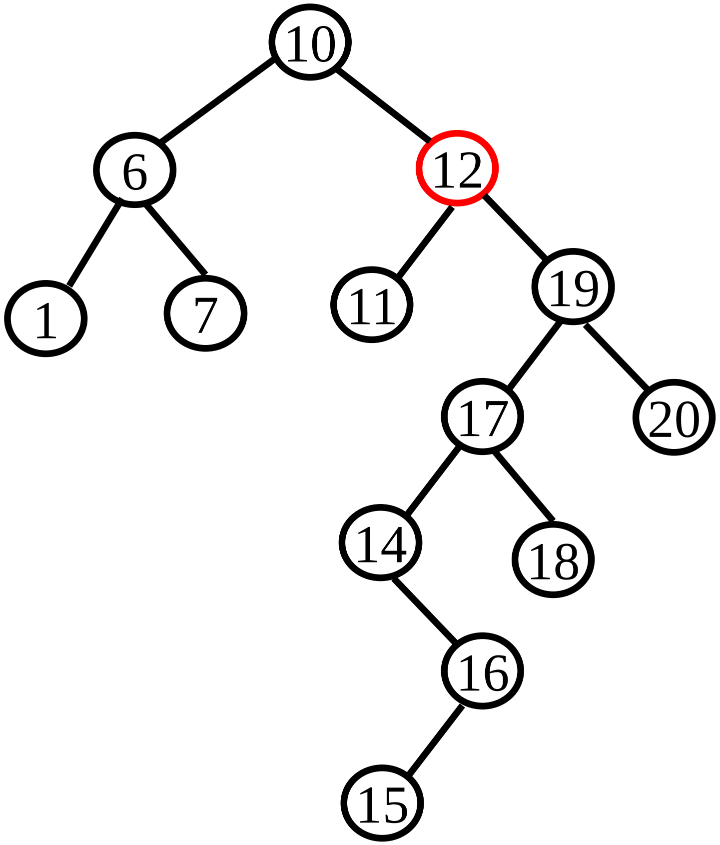


delete(12,T)



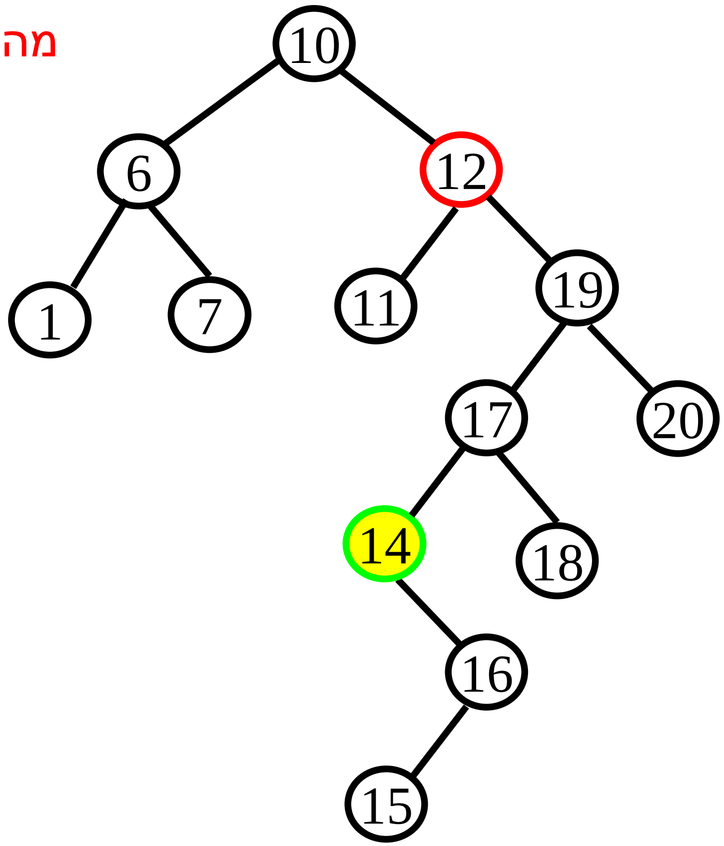
delete(12,T)

מה הבעיה ?



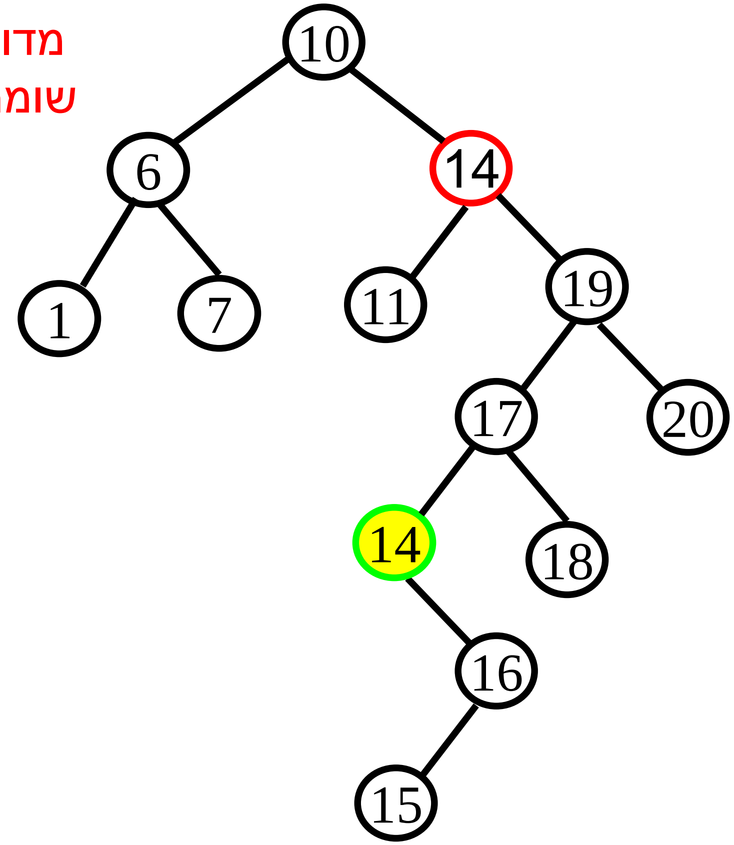
delete(12,T)

מה התכונה של 14 ביחס ל-12?



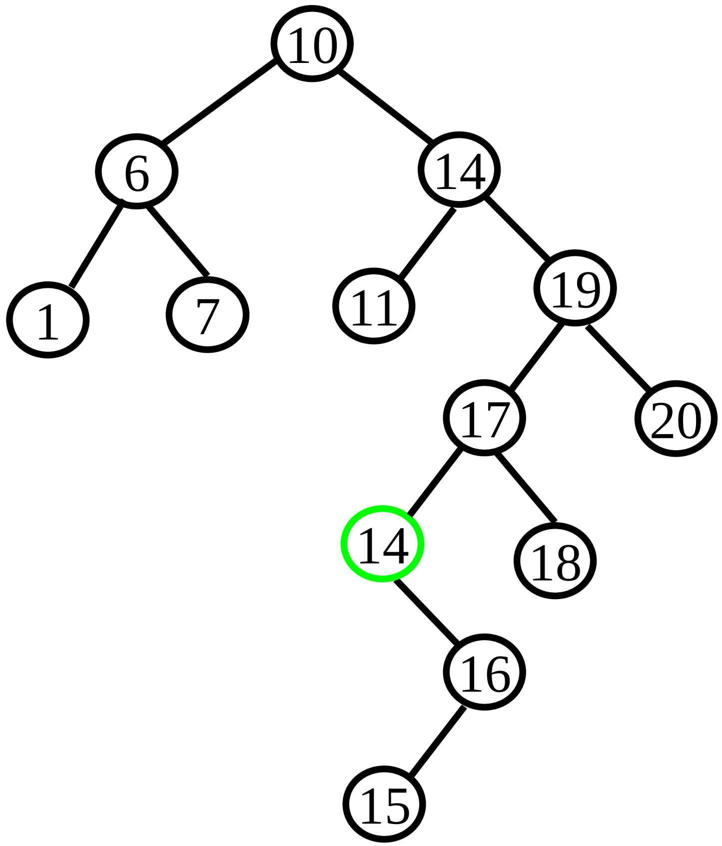
delete(12,T)

מדוע "שתילת" 14 במקום 12
שומרת על תכונות עץ חיפוש?

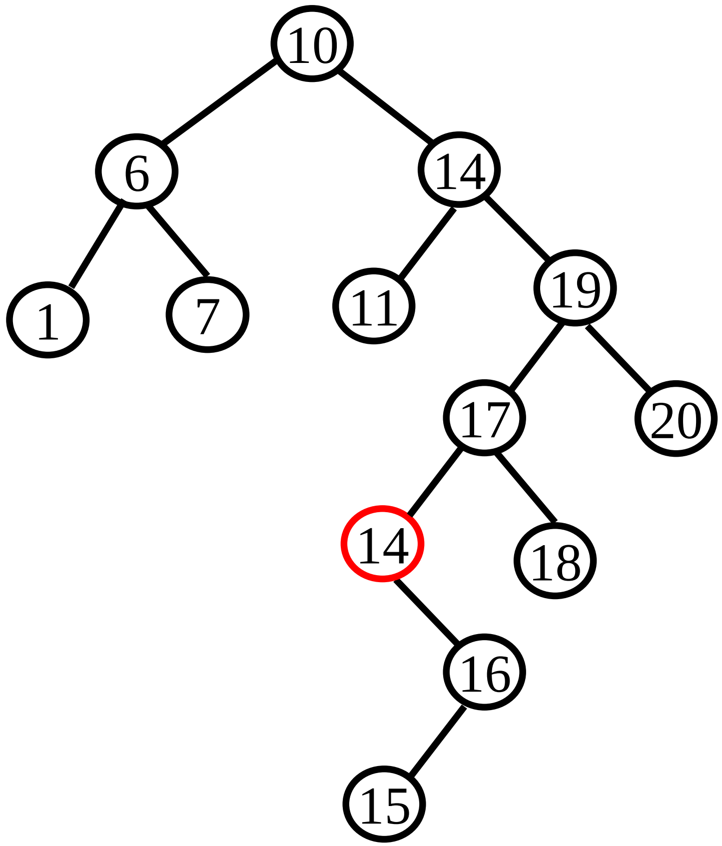


delete(12,T)

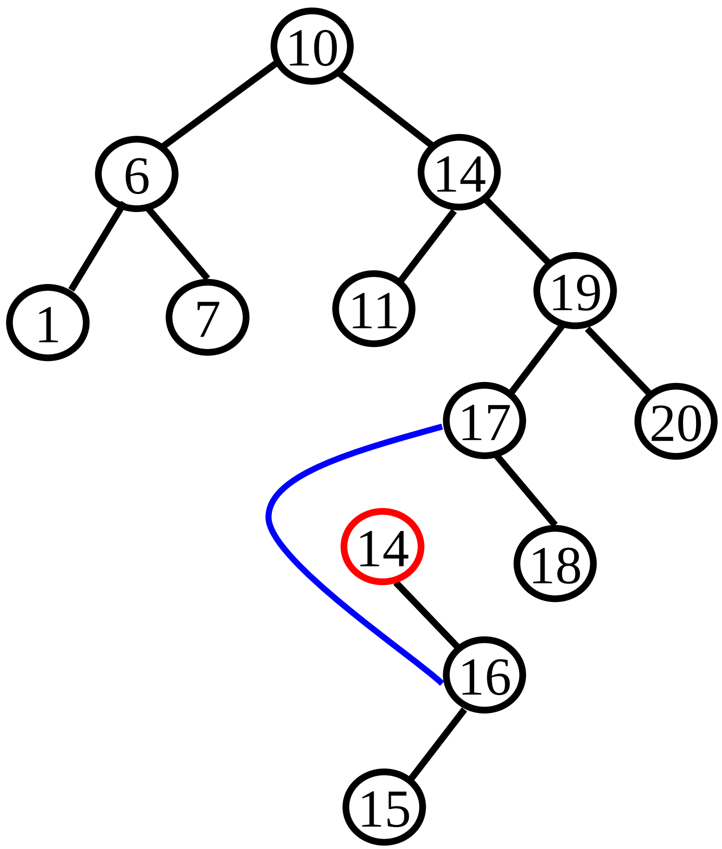
מה נותר לעשות?



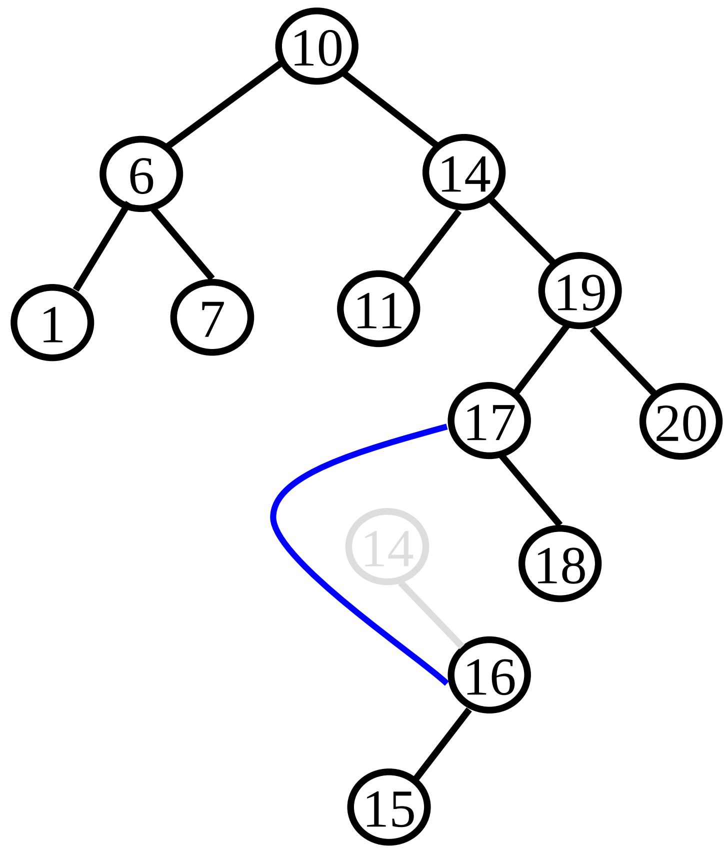
delete(12,T)



delete(12,T)

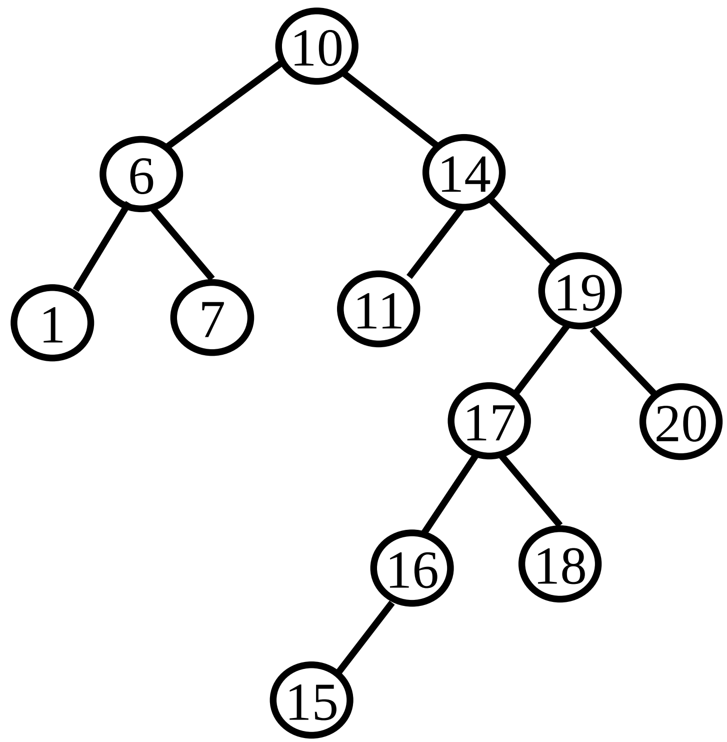


delete(12,T)



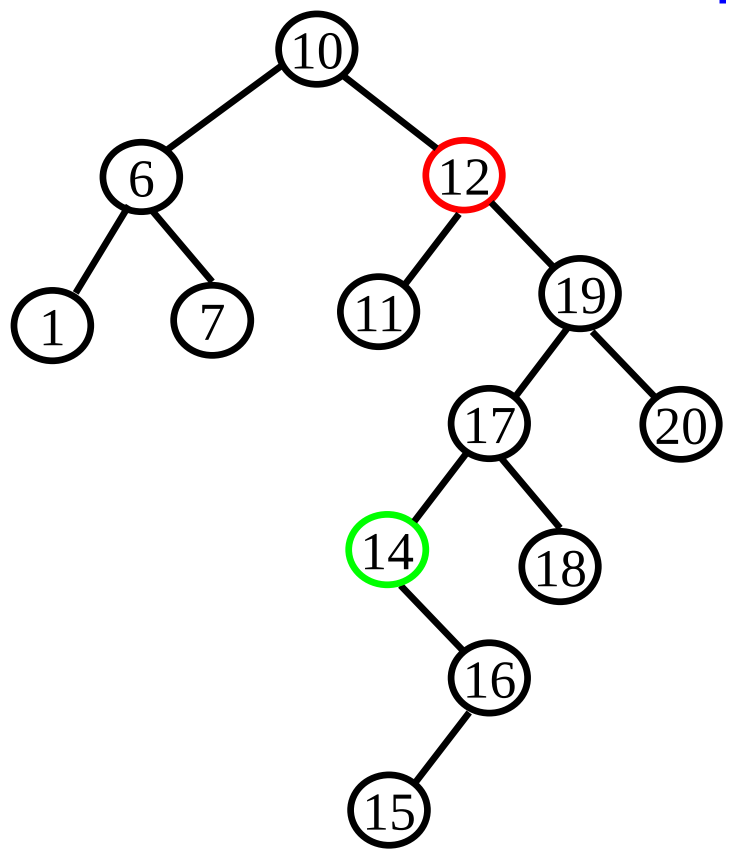
עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

delete(12,T)



delete(12,T)

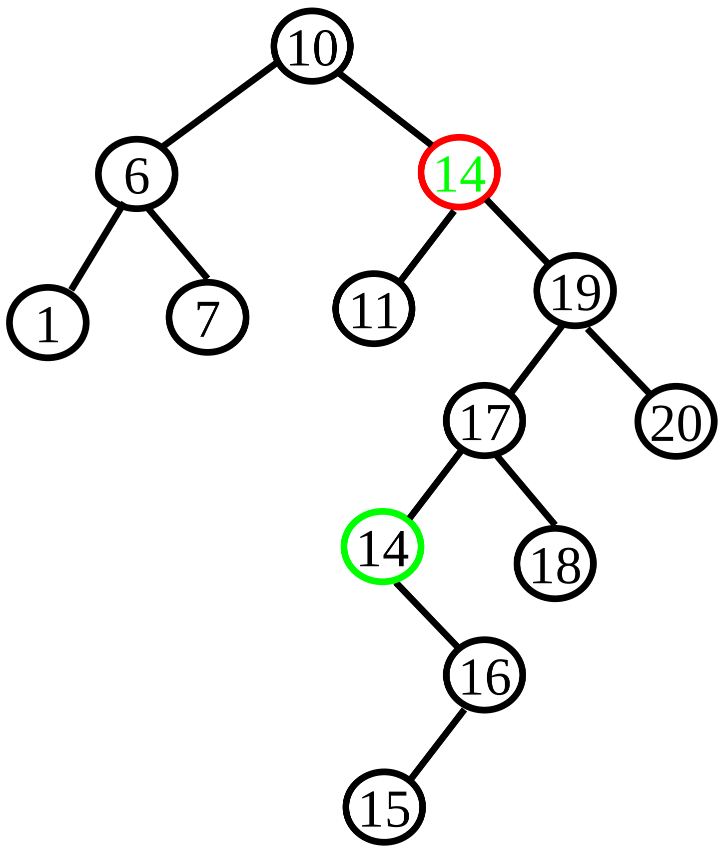
מכיוון שלצומת הנוכחי יש שני בנים, העוקב שלו יהיה המינימום בתת העץ הימני שלו



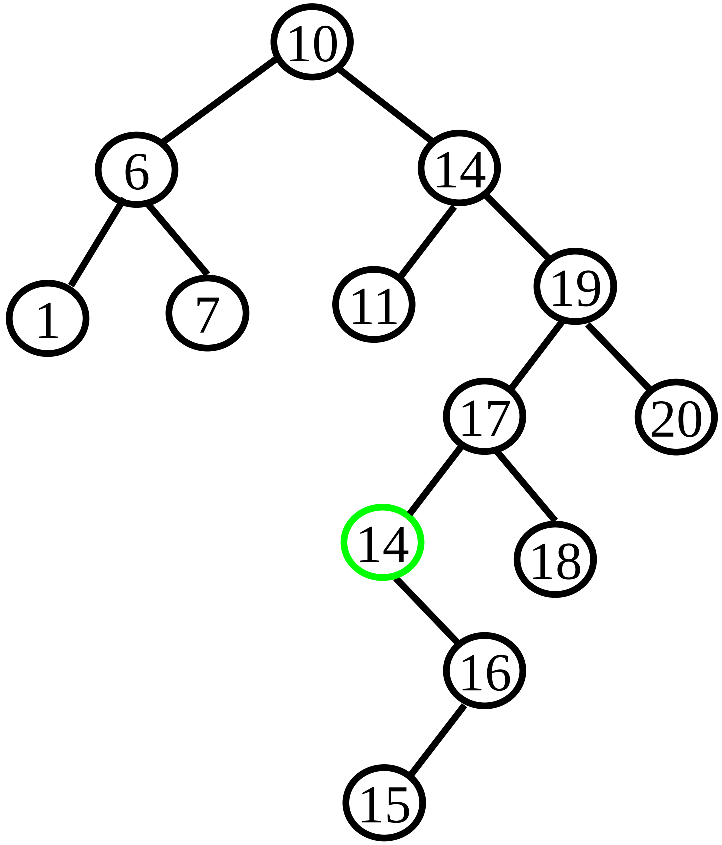
לעוקב בהכרח אין בן שמאלי.
(אילו היה ל-14 בן שמאלי אז הוא היה קטן מ-14 וגדול מ-12, אבל אז 14 לא היה העוקב של 12...)

עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

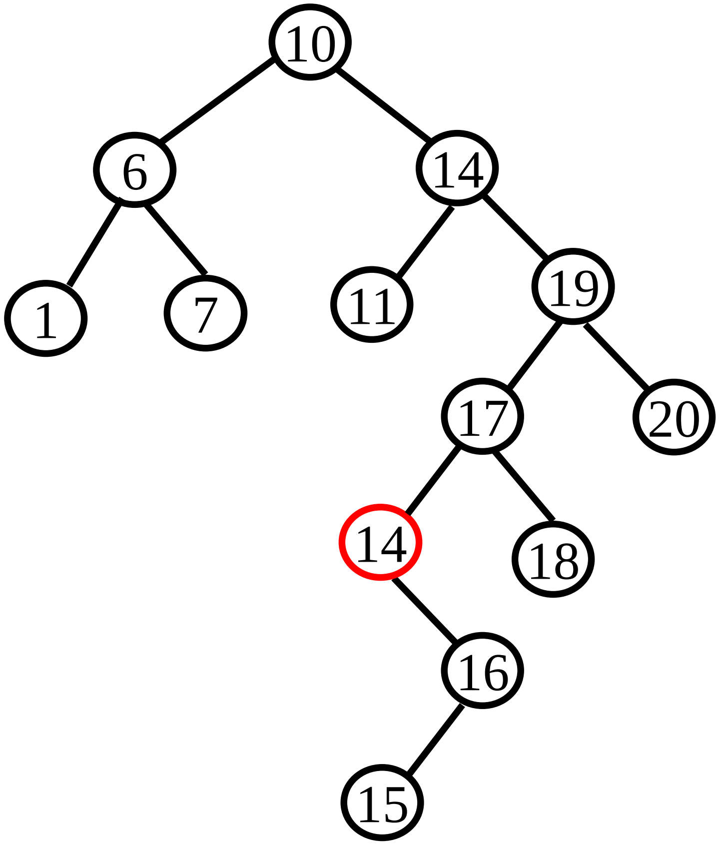
delete(12,T)



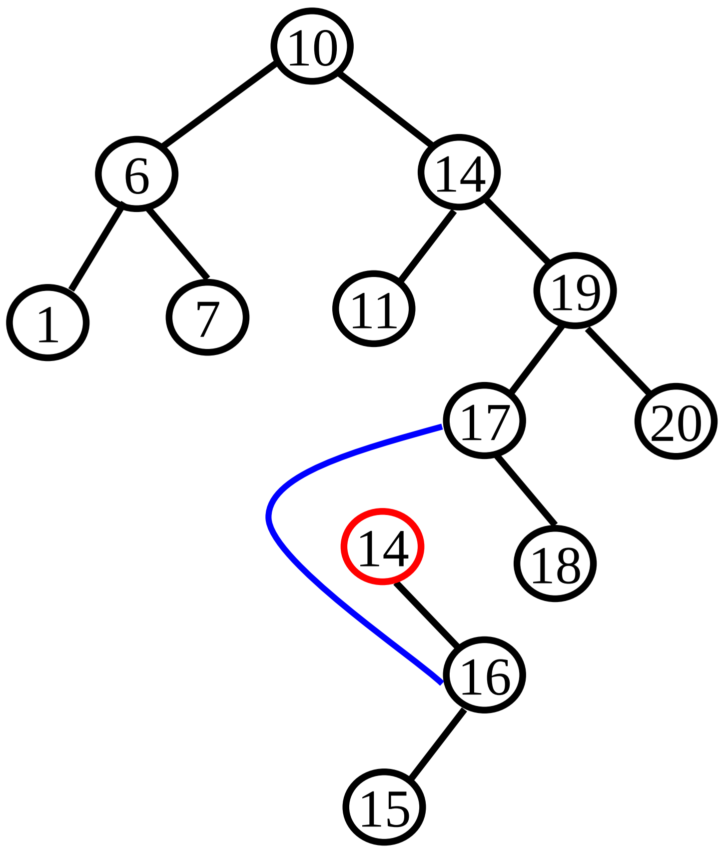
delete(12,T)



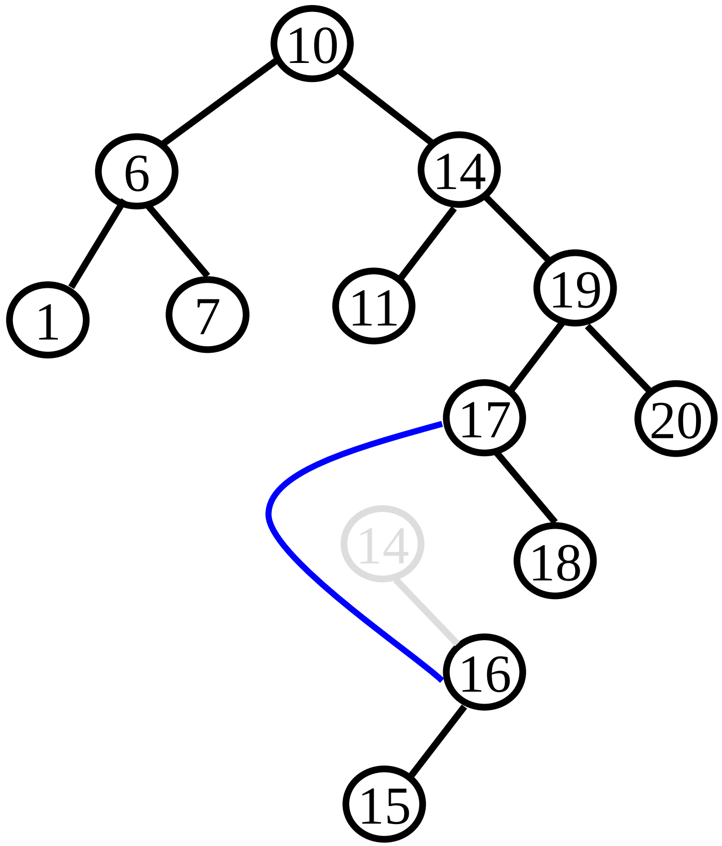
delete(12,T)



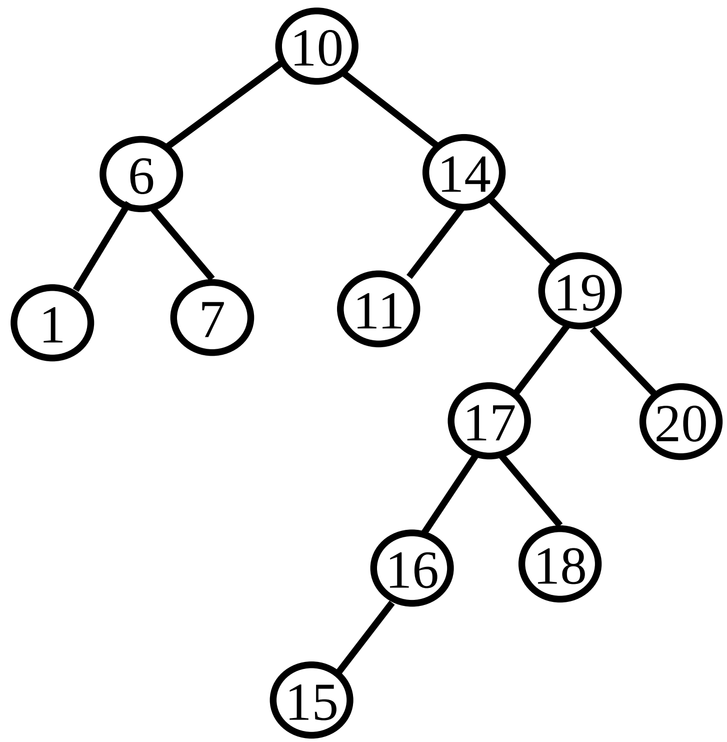
delete(12,T)



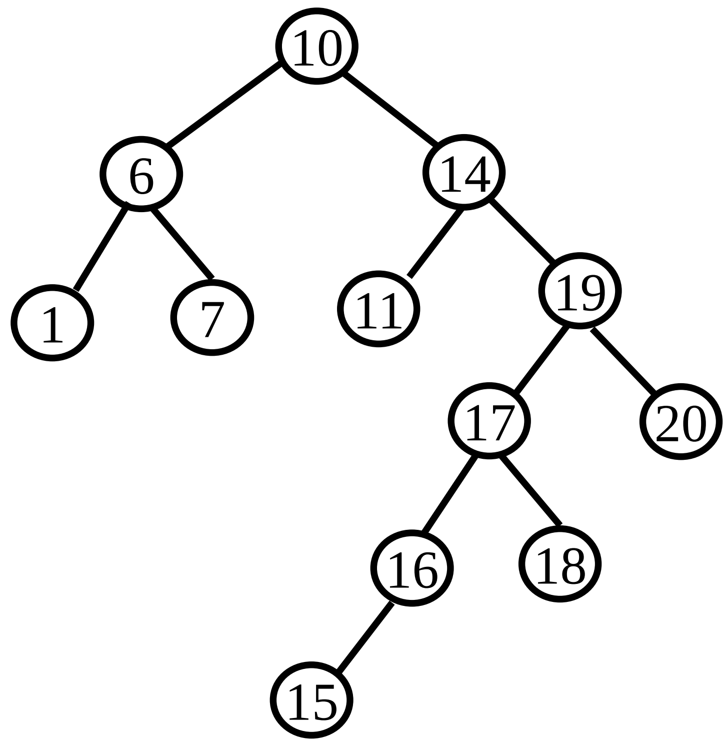
delete(12,T)



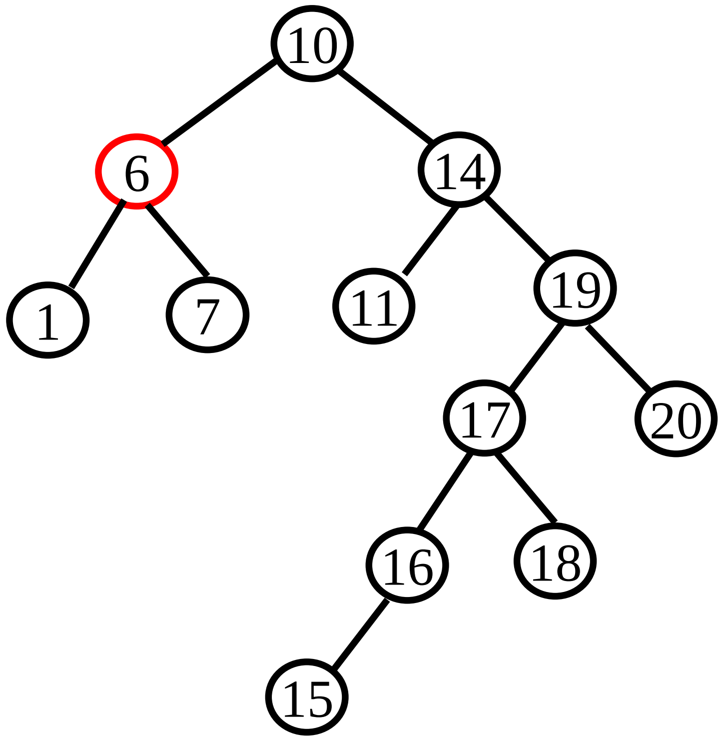
delete(12,T)



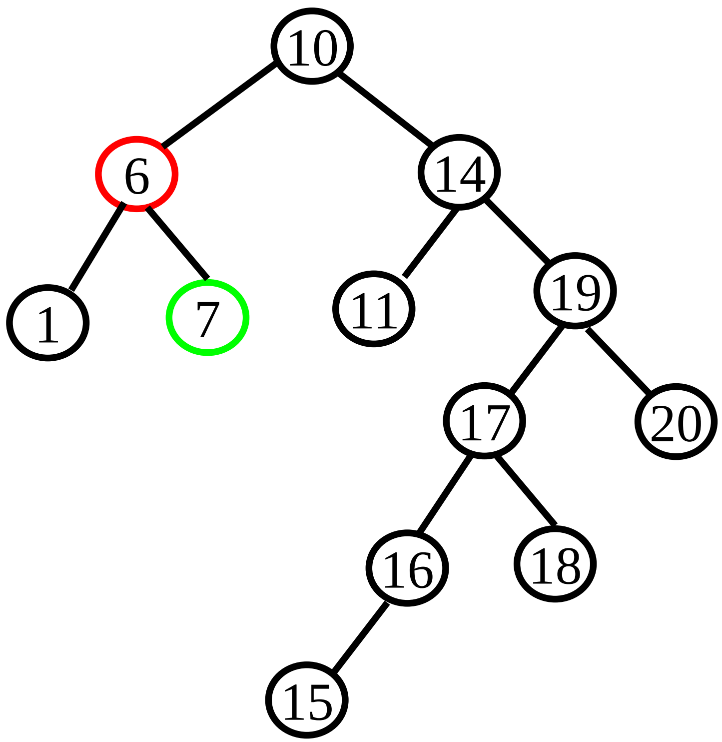
delete(6,T)



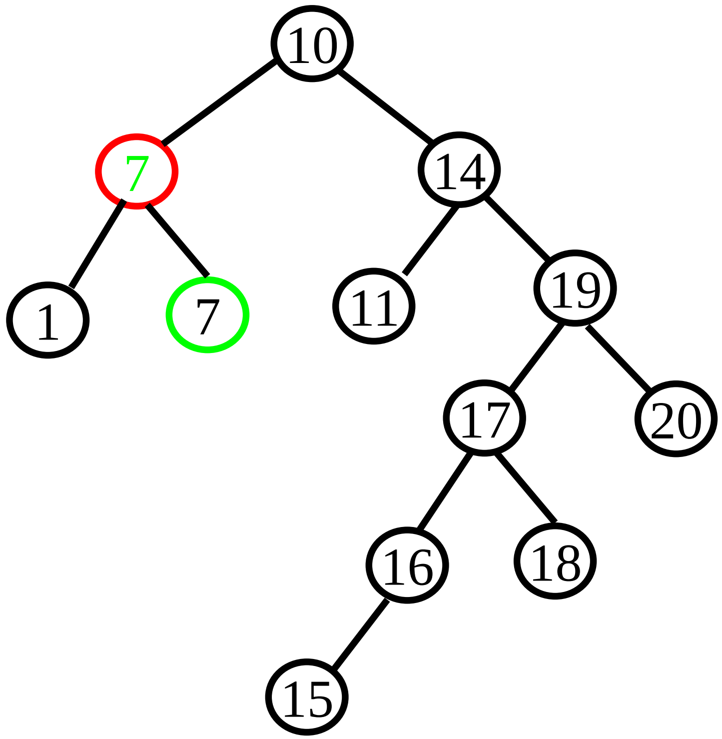
delete(6,T)



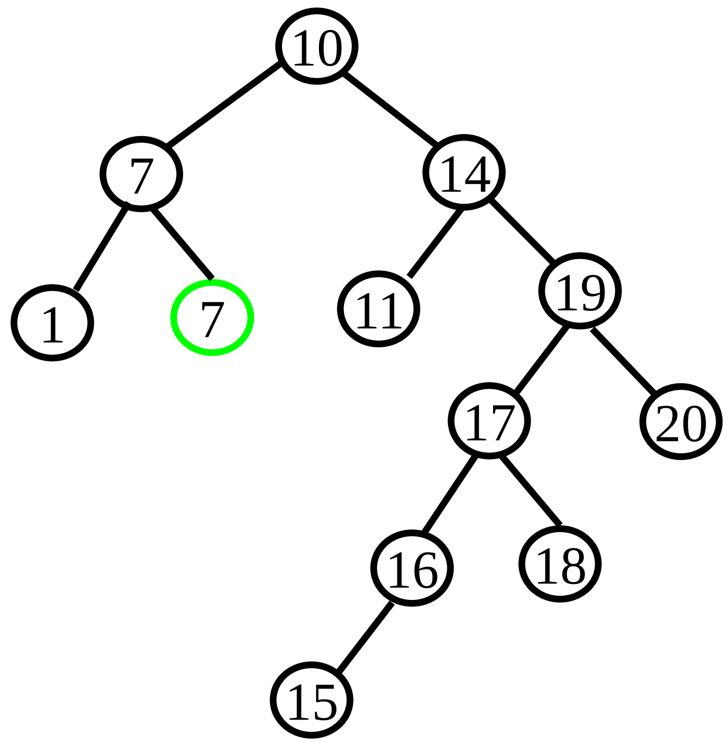
delete(6,T)



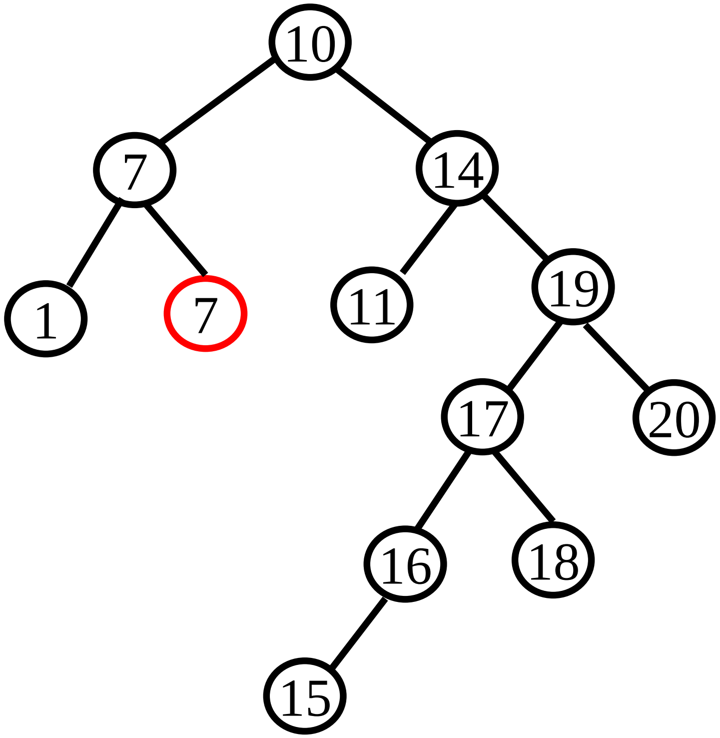
delete(6,T)



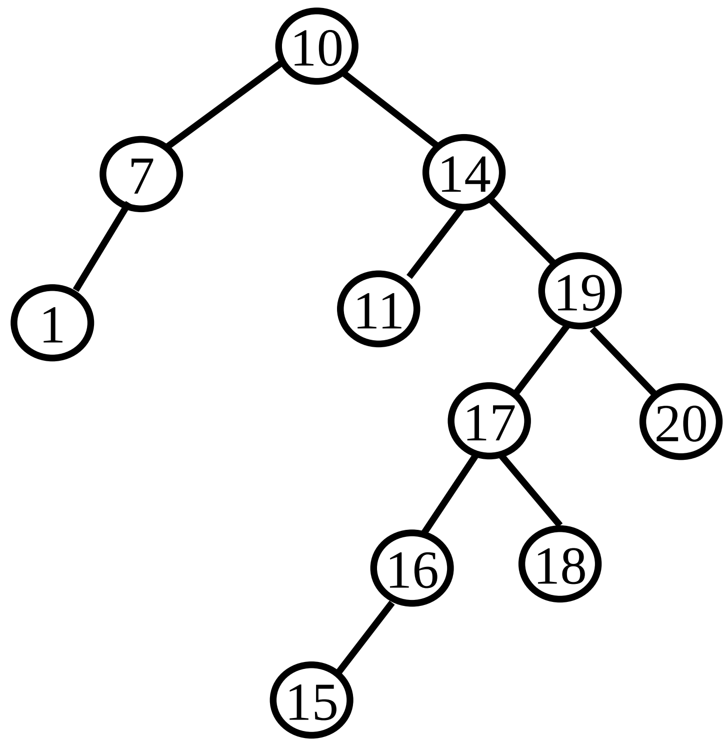
delete(6,T)



delete(6,T)



delete(6,T)



דוגמא לבניית עץ חיפוש

התחל בעץ ריק

בכל שלב הכנס לעץ איבר אחד, יש לקרוא את הנתונים משמאל לימין.

5,2,6,3,7,3,1,8,9

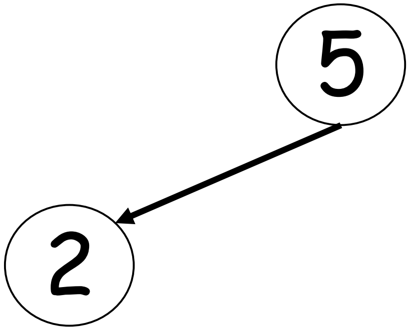
5,2,6,3,7,3,1,8,9

5

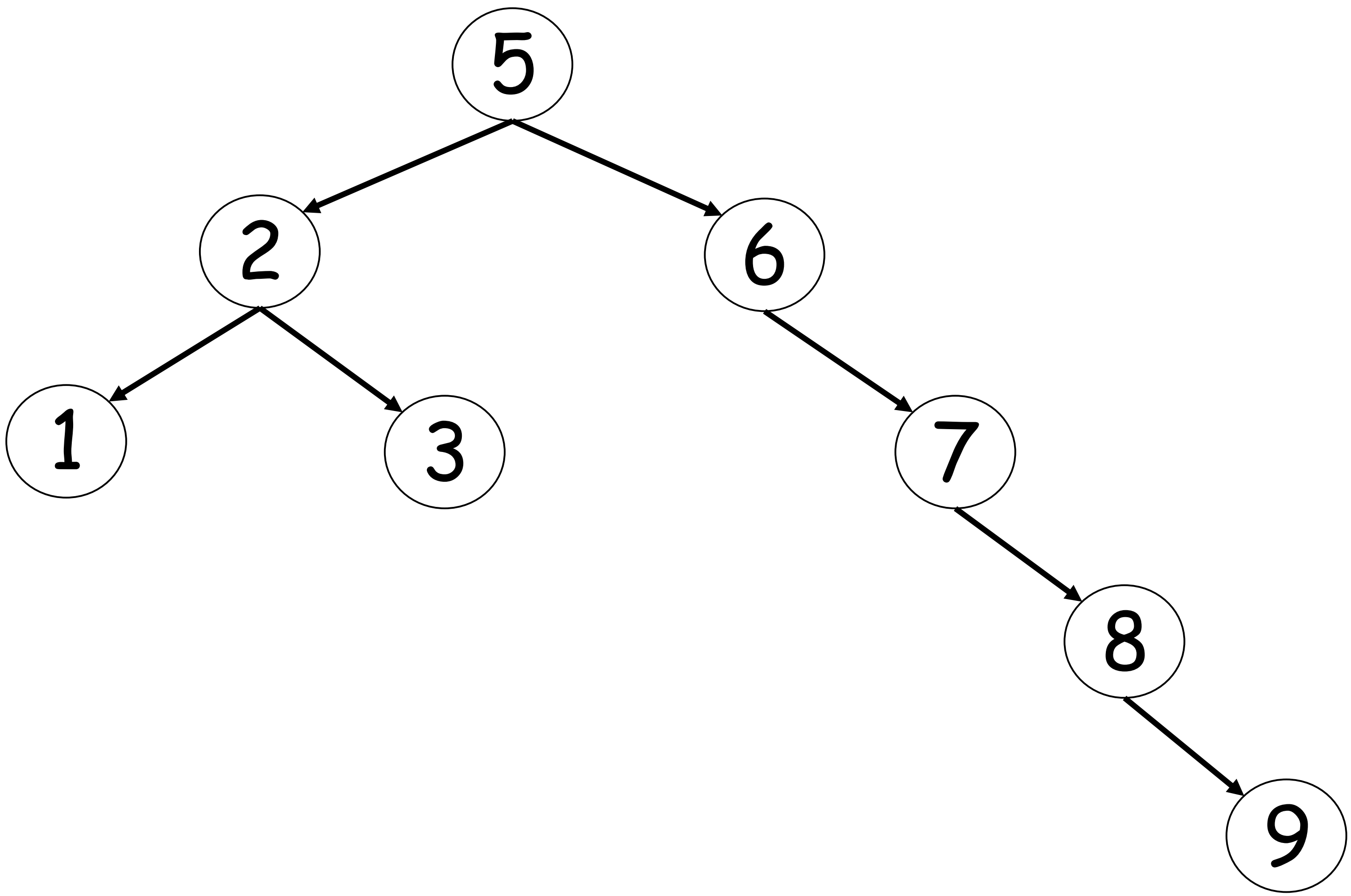
5,2,6,3,7,1,8,9

5

5,2,6,3,7,1,8,9



5,2,6,3,7,1,8,9



ניתוח זמן

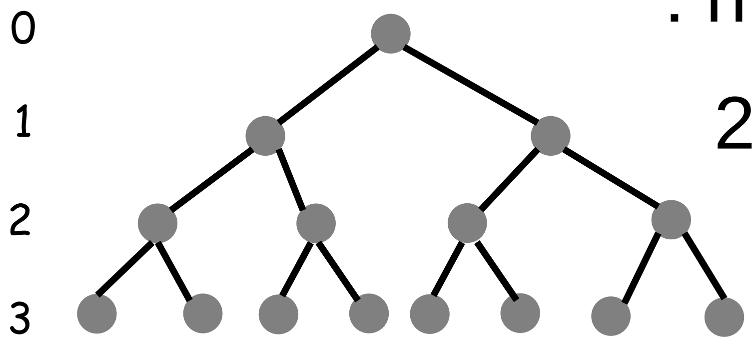
עלות של כל אחת מהפעולות: חיפוש
הכנסה
הוצאה
היא $O(h)$ כאשר h - גובה העץ.

ניתוח זמן

עלות של כל אחת מהפעולות: חיפוש
הכנסה
הוצאה
היא O(h) כאשר h - גובה העץ.

מהו הגובה של עץ חיפוש בינרי?

גובה עץ בינרי



מספר הצמתים בעץ בינרי **שלם** בגובה h :

$$2^0 + 2^1 + 2^2 + 2^3 + \ldots + 2^h = 2^{h+1}-1$$

הגובה של עץ בינרי **שלם** בעל n צמתים: $\log(n+1)-1$

בעץ בינרי **שלם** מתקיים: $h=\theta(\log n)$

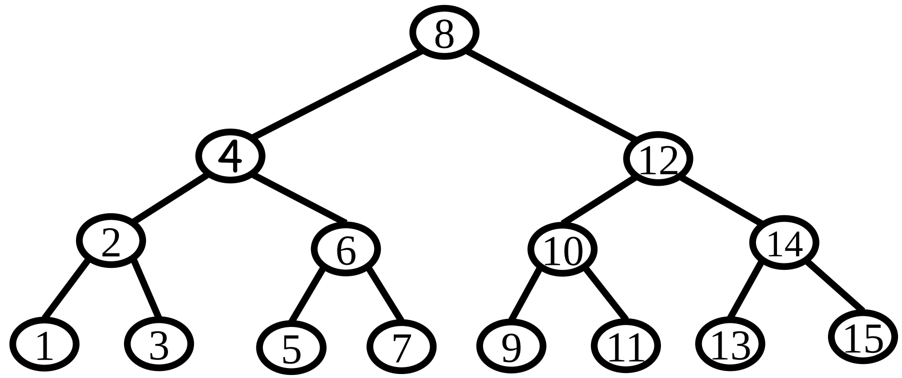
בעץ בינרי כלשהו מתקיים: $\boldsymbol{h=O(n)}$

$\boldsymbol{h=\Omega(\log n)}$

גובה עץ חיפוש בינרי

צורת עץ חיפוש בינרי תלויה בסדר הכנסת האיברים לעץ

במקרה הטוב: העץ הוא עץ שלם ולכן גובה העץ $\theta(\log n)$



במקרה הגרוע: העץ הוא "מסלול" ולכן גובה העץ הוא n-1