עץ חיפוש בינארי

Binary-Search Tree (BST)

מה נלמד?

- **הגדרת עץ חיפוש בינארי**
- **ייצוג של עץ חיפוש בינארי**
- **פעולות על עץ חיפוש בינארי**

Ordered Dictionary

$Search(k)$

$Insert(x)$

$Delete(x)$

$Min()$

$Max()$

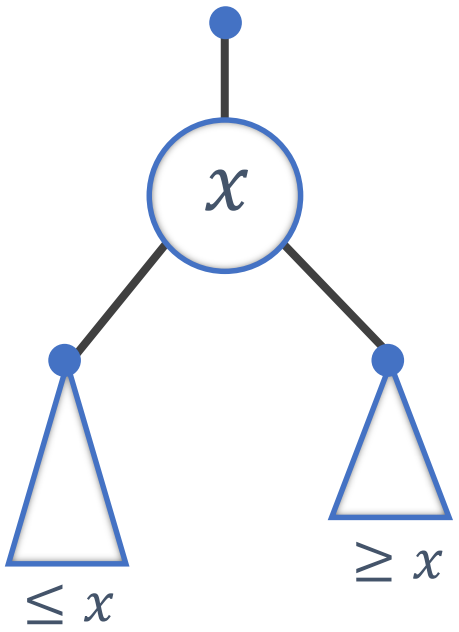
$Successor(x)$

$Predecessor(x)$

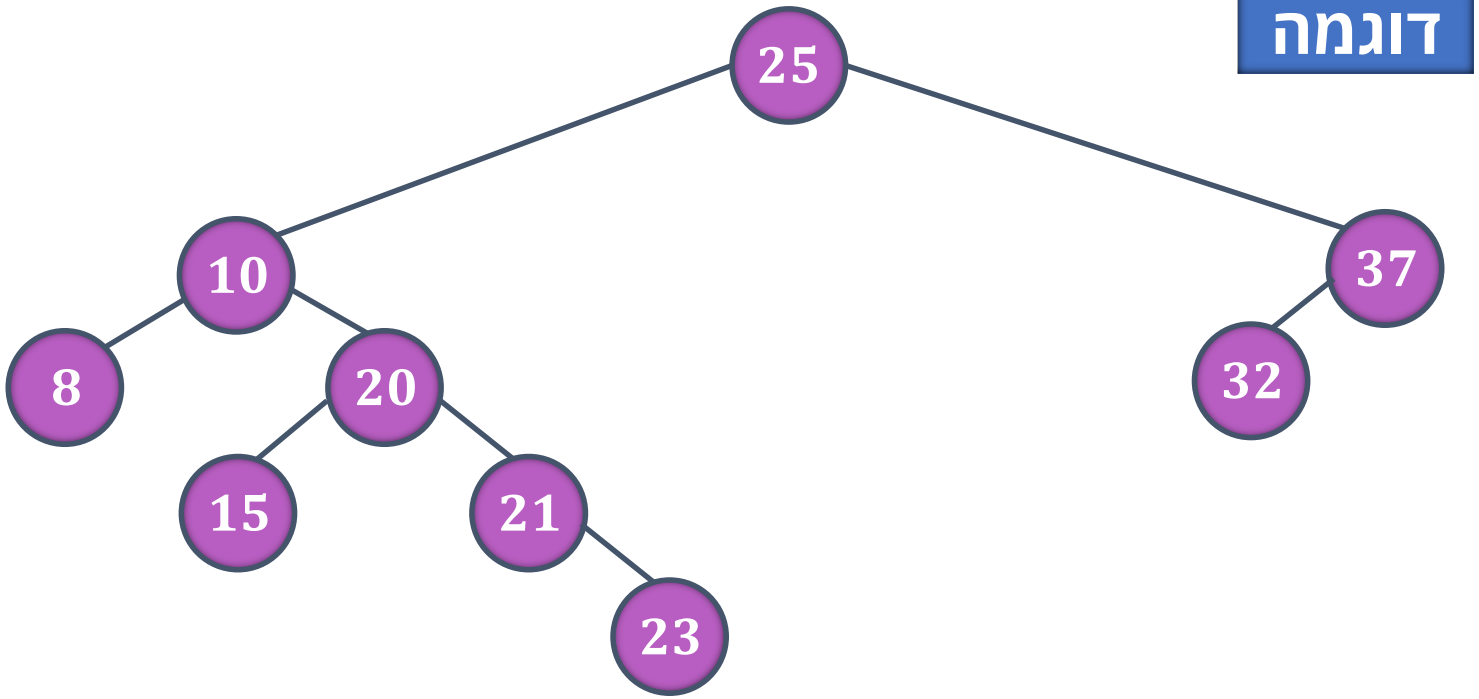
עץ חיפוש בינארי - הגדרה

עץ חיפוש בינארי הוא עץ בינארי המקיים תכונה שנקראת **תכונת-עץ-החיפוש-הבינארי**

יהי x צומת בעץ חיפוש בינארי, אם y הוא צומת בתת-עץ השמאלי של x ו- z צומת בתת-עץ הימני של x , אזי $y.key \leq x.key \leq z.key$

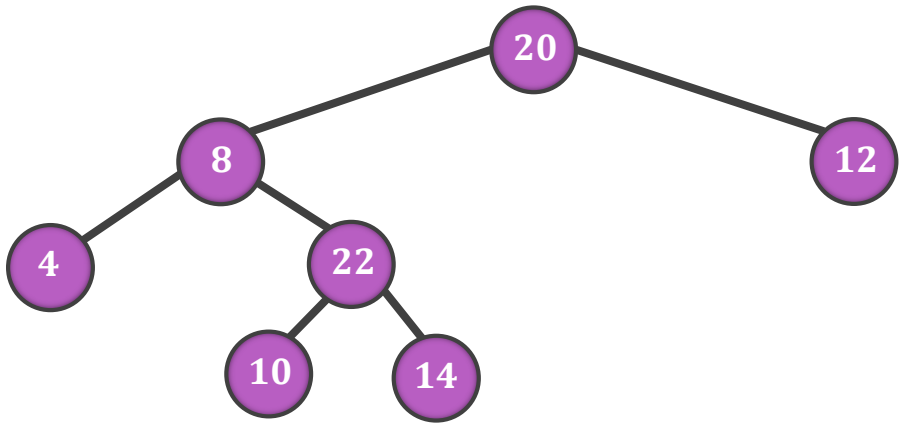


דוגמה

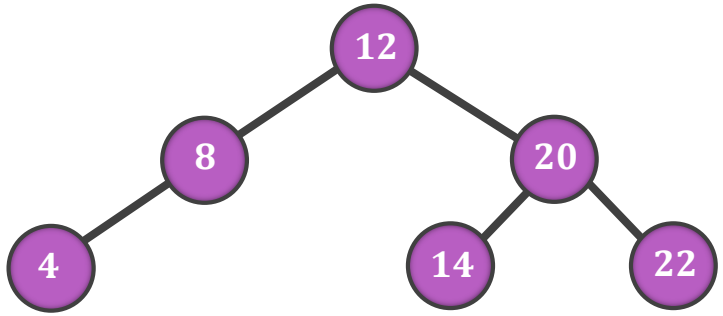


? אילו מהעצים הבאים הם עצי חיפש בינאריים?

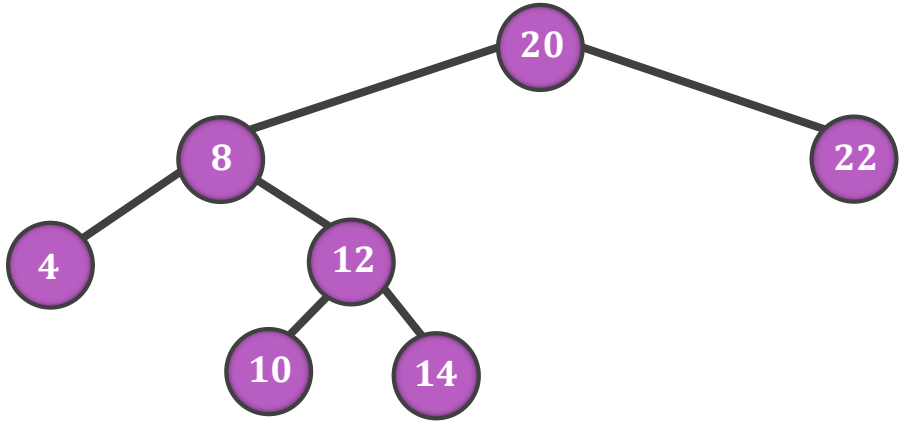
.1



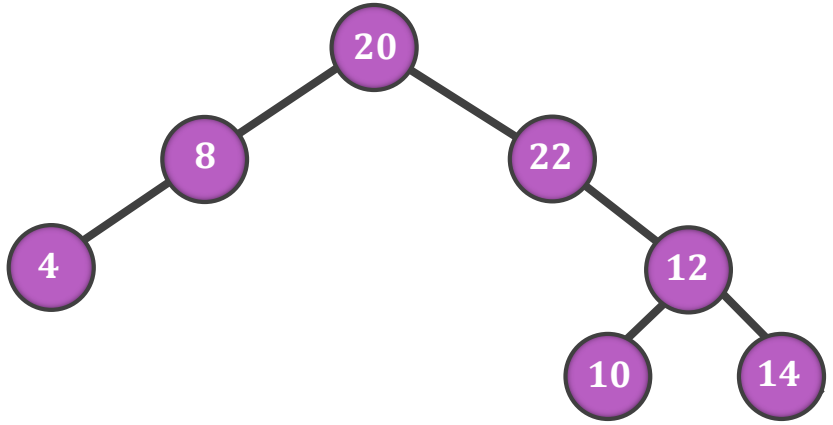
2.



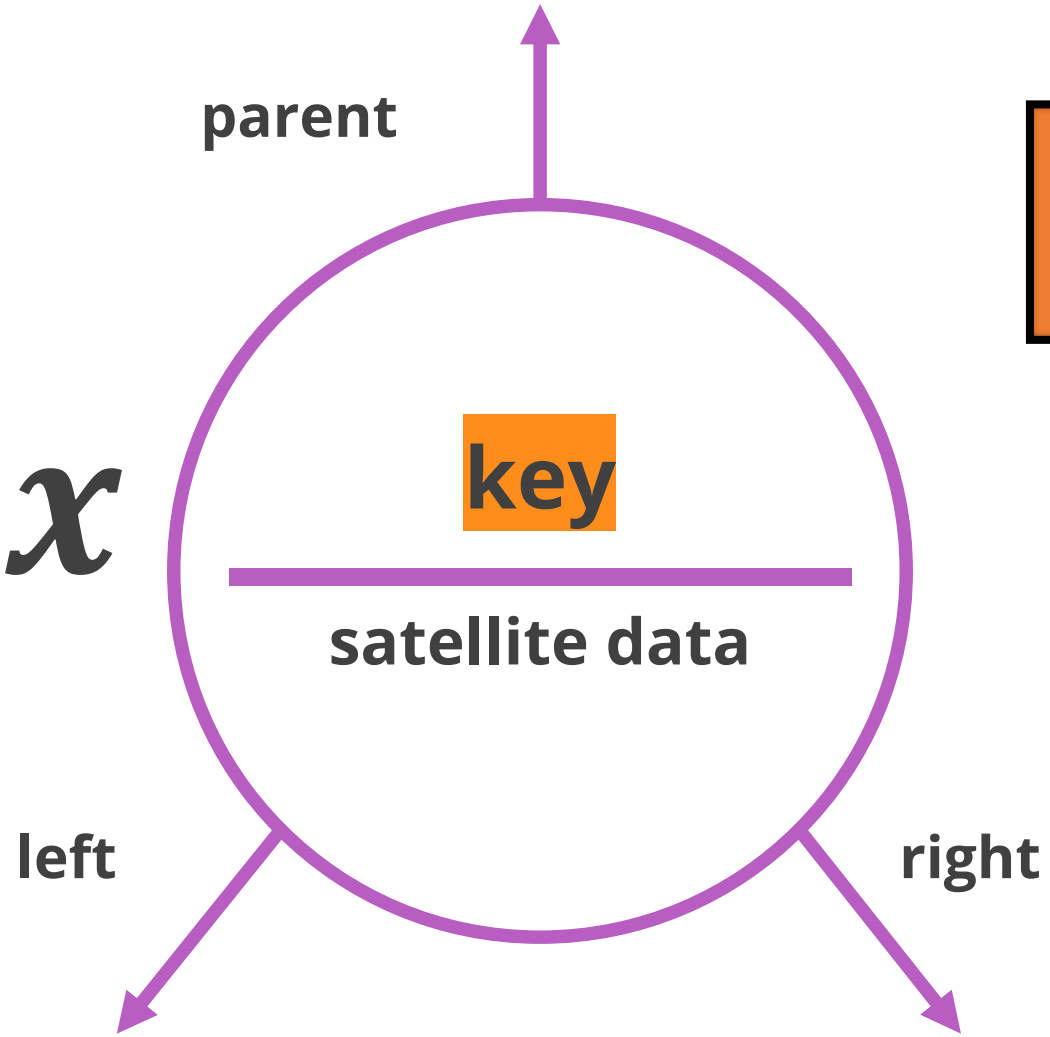
3.



4.



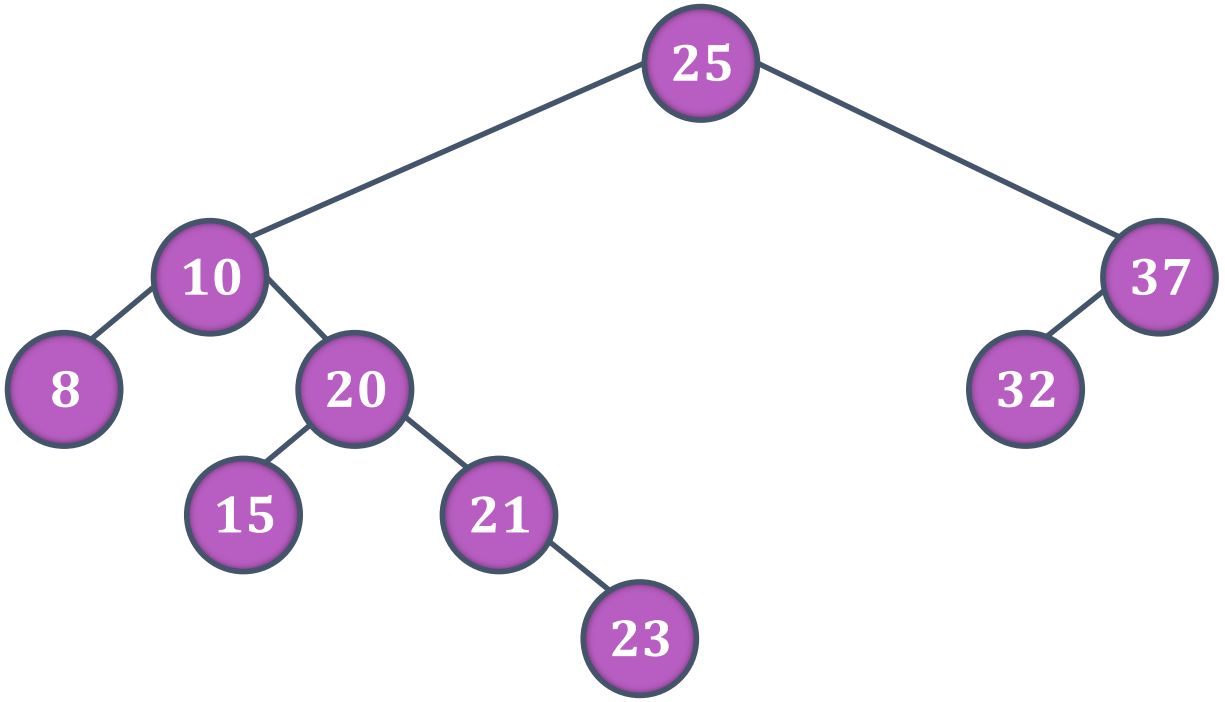
ייצוג של עץ חיפוש בינארי



העץ נתון על ידי מצביע לשורש $T.root$

סריקה תוכית של עץ חיפוש בינארי

```
Inorder(x)
1  if (x ≠ NULL)
2      Inorder (x.left)
3      print (x.key)
4      Inorder(x.right)
```

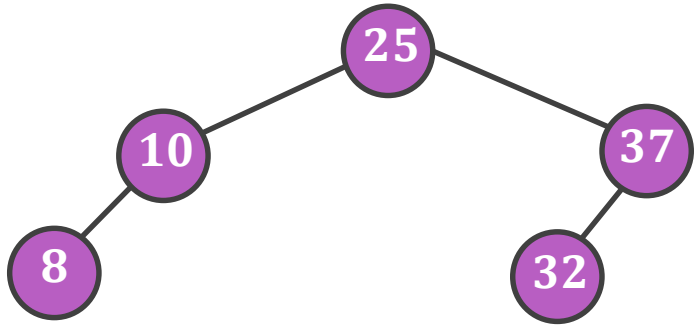


8	10	15	20	21	23	25	32	37

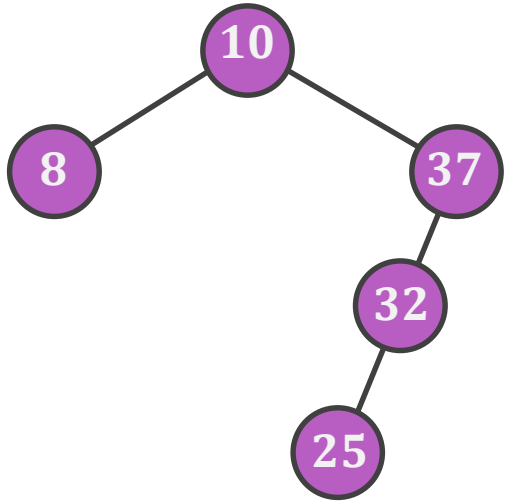
גובה של עץ חיפוש בינארי

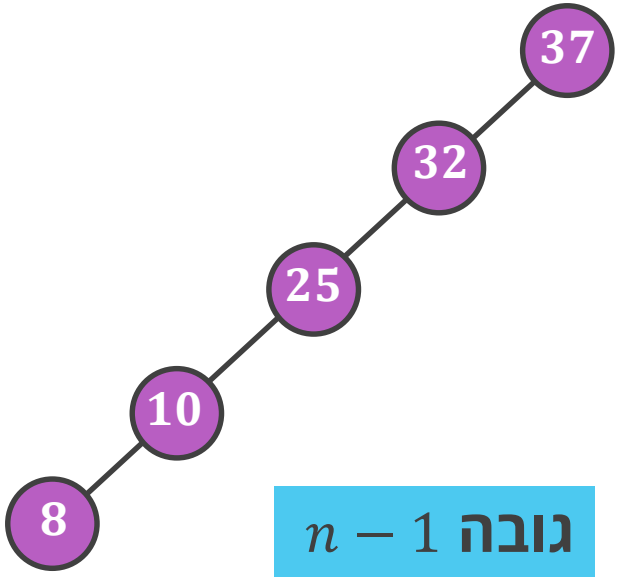
עבור קבוצה נתונה של מפתחות ניתן לבנות הרבה עצי חיפוש בינאריים

דוגמה: עבור קבוצת מפתחות 8,10,25,37,32



גובה $\lfloor \log n \rfloor$





גובה $n-1$

הערה: גובה של עץ חיפוש בינארי יכול לקבל כל ערך בין $\lfloor \log n \rfloor$ **ל-** $n-1$

Ordered Dictionary

$Search(k)$

$Insert(x)$

$Delete(x)$

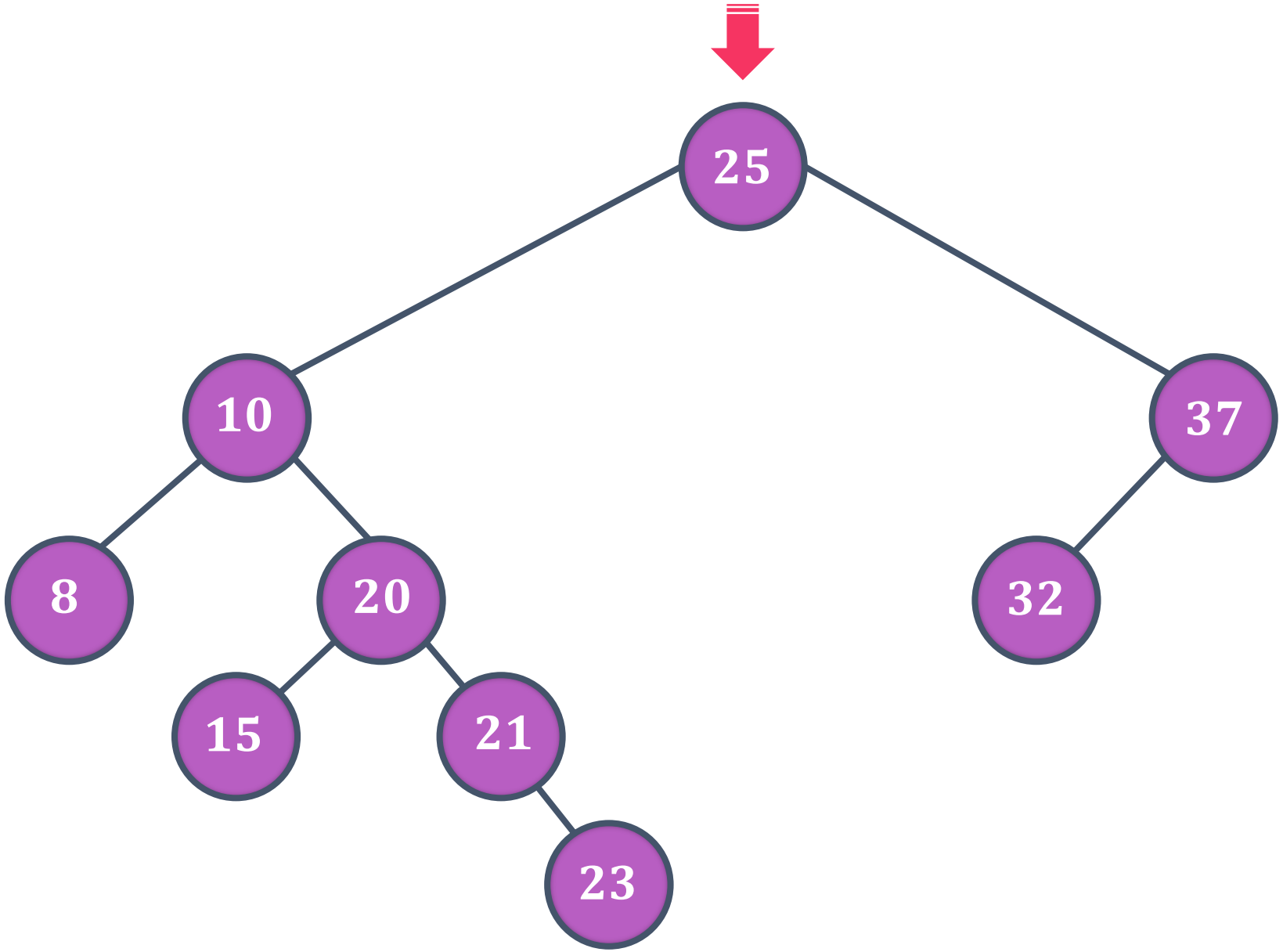
$Min()$

$Max()$

$Successor(x)$

$Predecessor(x)$

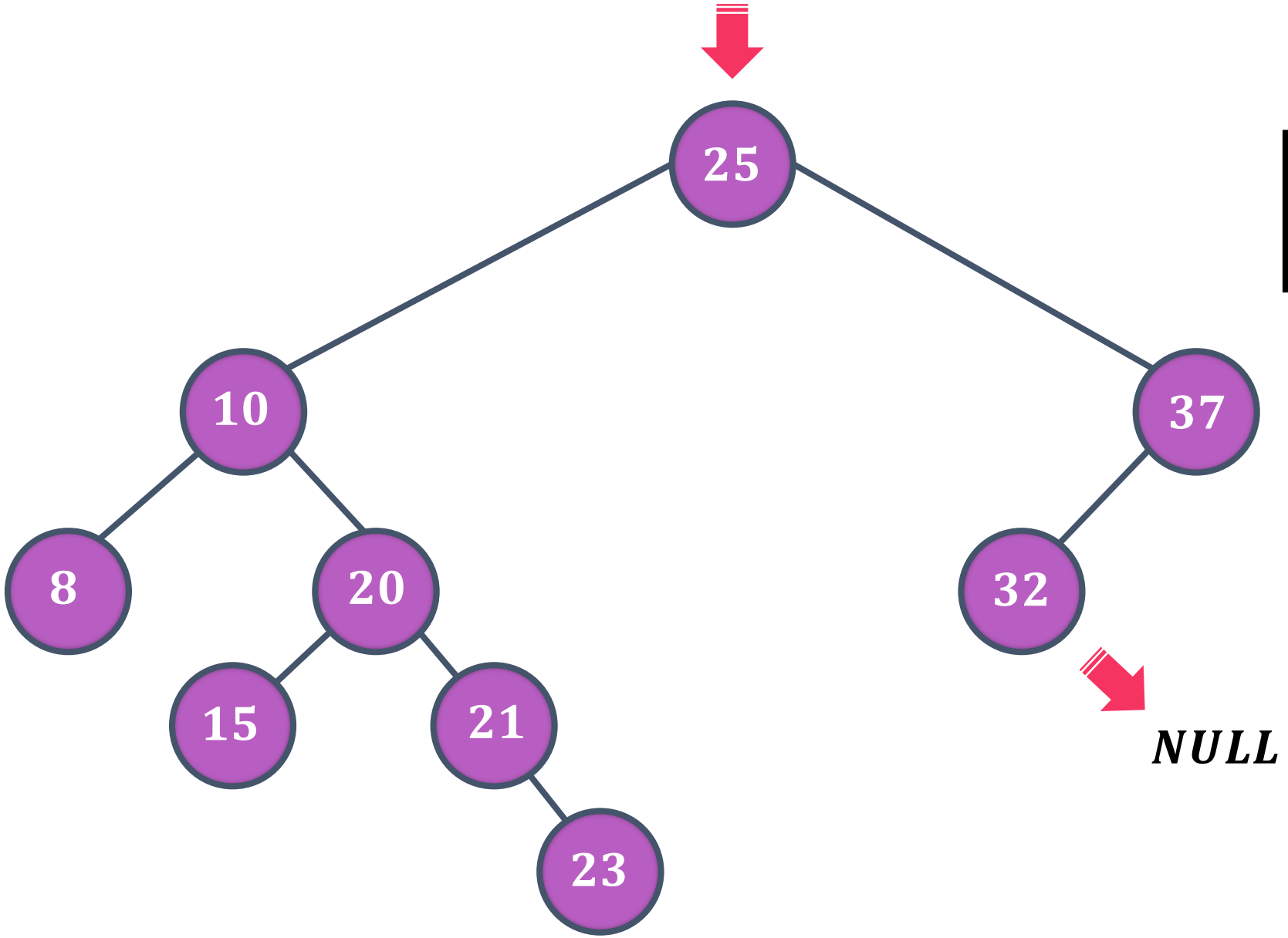
פעולת חיפוש



Search 21

פעולת חיפוש

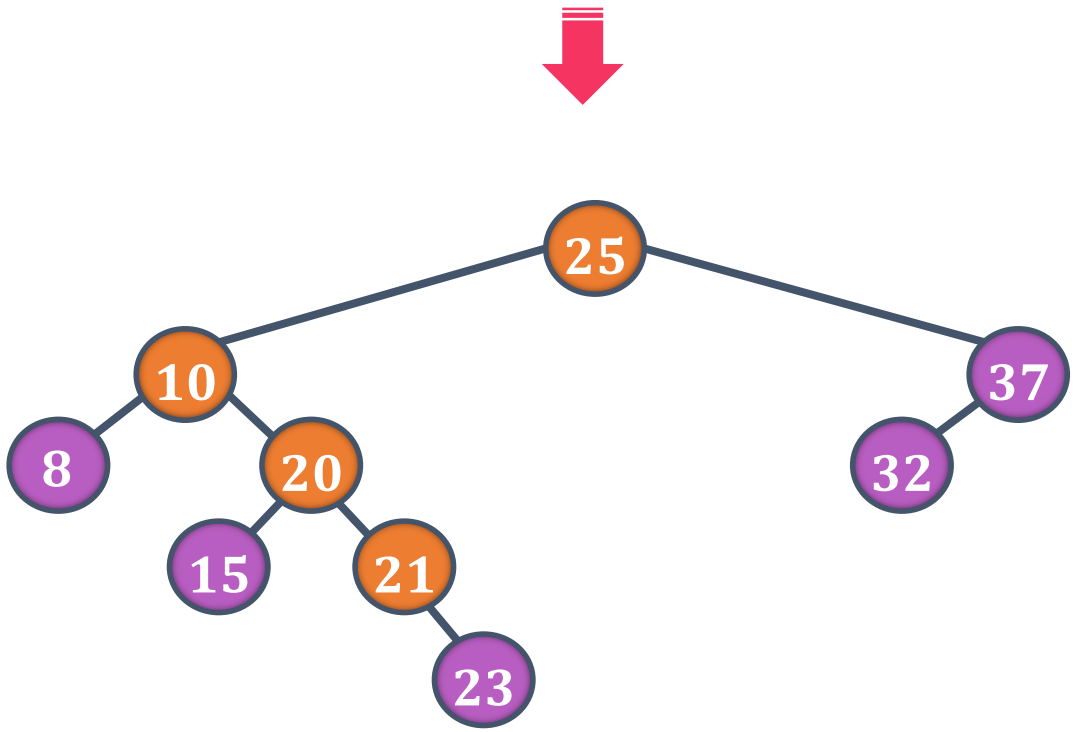
Search 35



חיפוש פסאודו קוד

```
Search (x, k)
if x = NULL or k = x.key
        return x
if k < x.key
        return Search(x.left, k)
else return Search(x.right, k)
```

קריאה חיצונית $Search(T.root,k)$

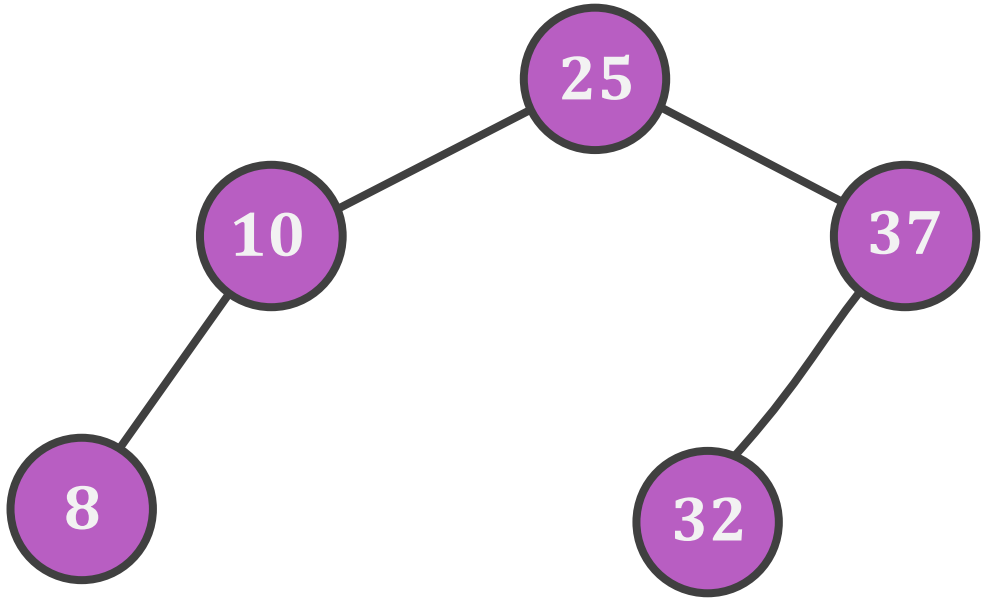


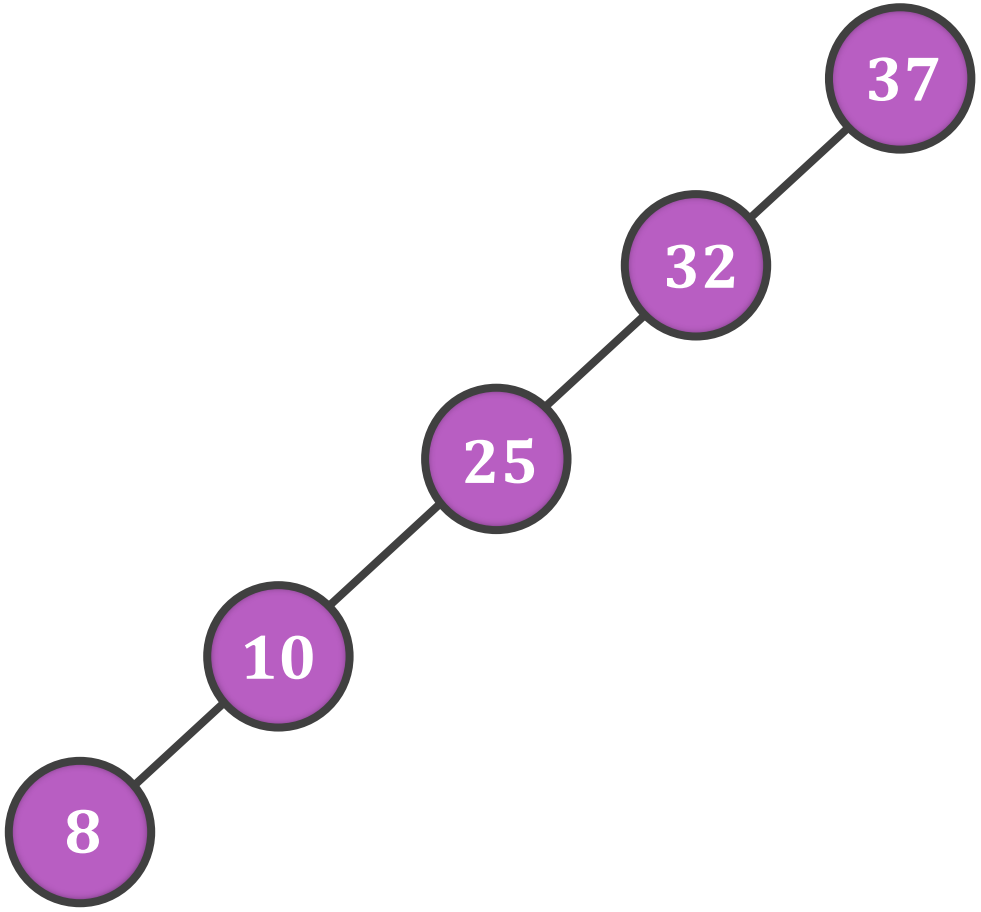
? זמן ריצה במקרה הגרוע של פעולת $Search$ בעץ חיפוש בינארי בעל n צמתים וגובה h הוא:

1. $\Theta(n)$
2. $\Theta(1)$
3. $\Theta(h)$
4. $\Theta(\log n)$

פעולת חיפוש

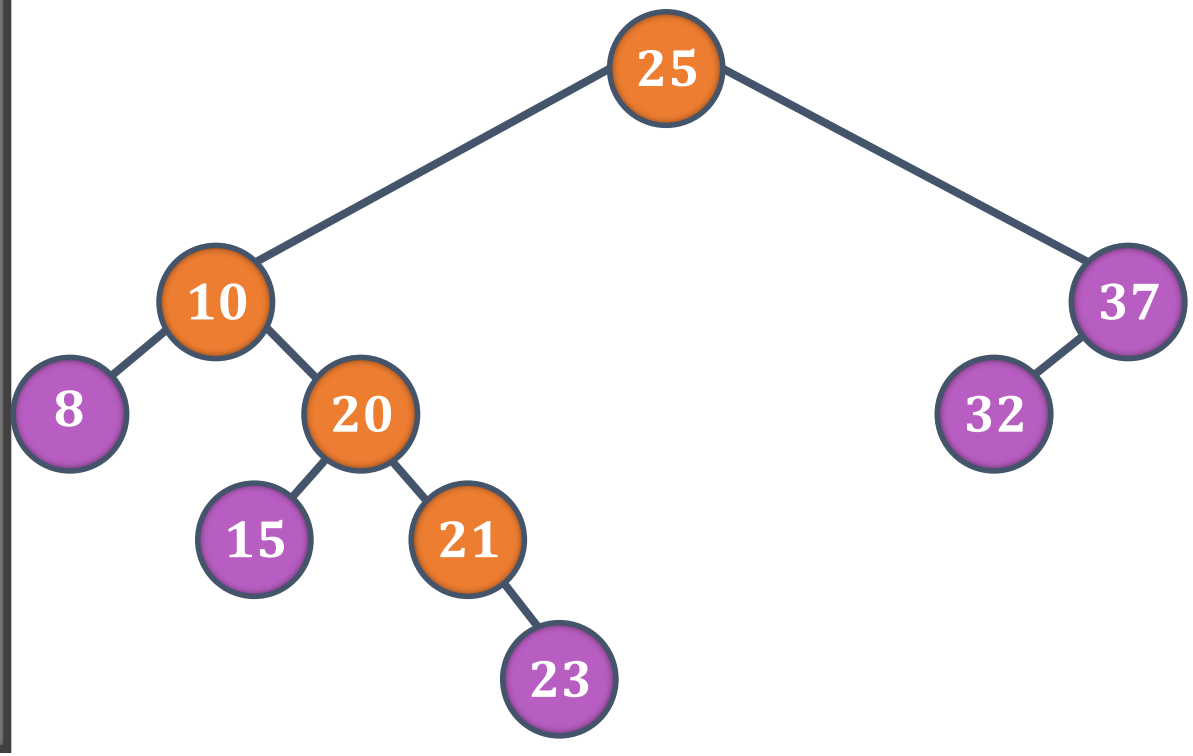
זמן ריצה





חיפוש פסאודו קוד

```
Search (x, k)
if x = NULL or k = x.key
        return x
if k < x.key
        return Search(x.left, k)
else return Search(x.right, k)
```



זמן ריצה הוא $\Theta(h)$, כאשר h הוא גובה העץ

מחפשים את 52 בעץ חיפוש בינארי.
איזה מבין הסדרות הבאות אינה יכולה להיות סדרת מספרים בה נתקל במהלך החיפוש?

1. 9,10,20,34,52
2. 9,100,30,42,52
3. 9,42,32,40,52
4. 9,80,15,70,52
5. כל הסדרות אפשריות

Ordered Dictionary

$Search(k)$

$Insert(x)$

$Delete(x)$

$Min()$

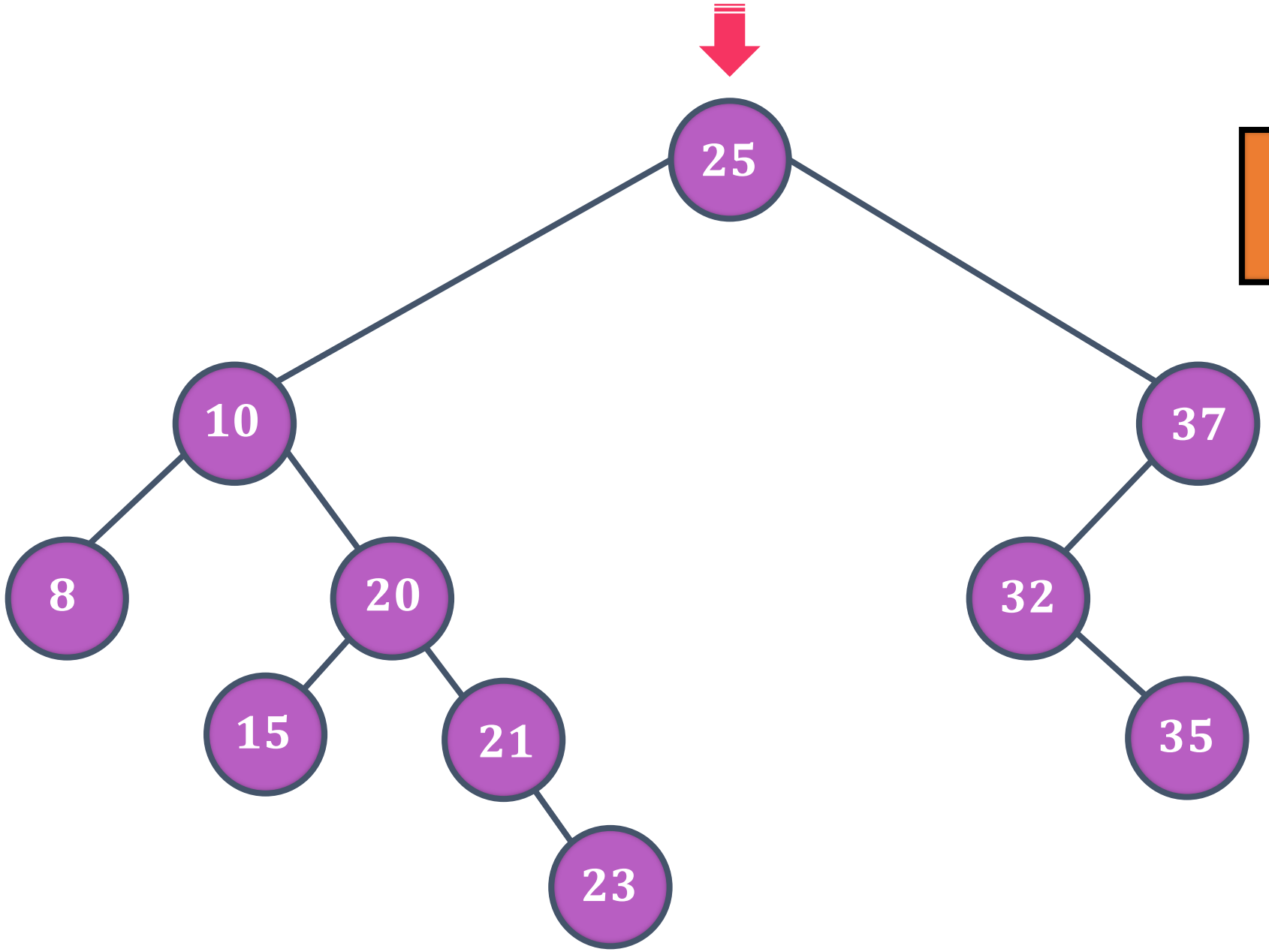
$Max()$

$Successor(x)$

$Predecessor(x)$

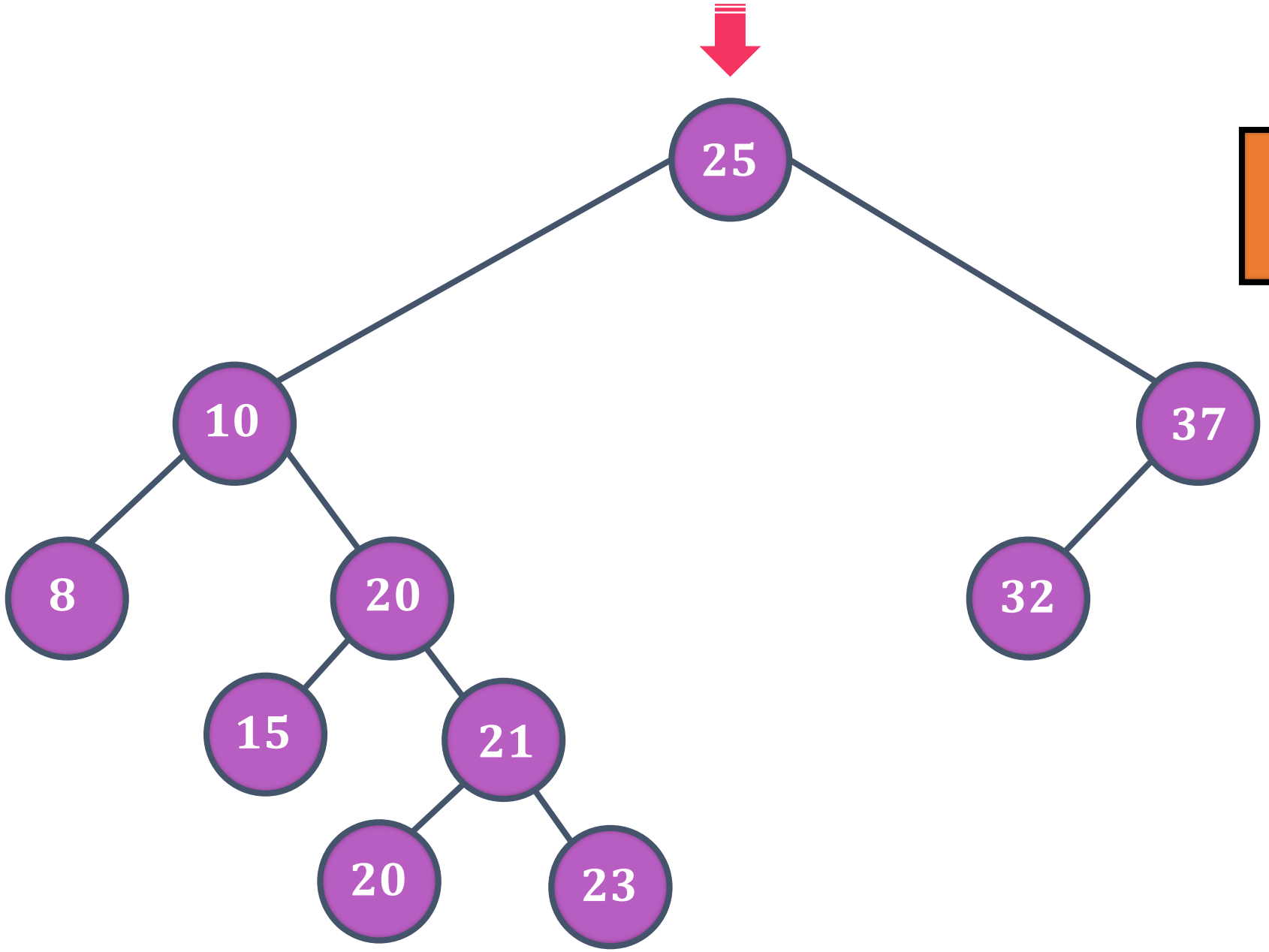
פעולת הכנסה

Insert 35



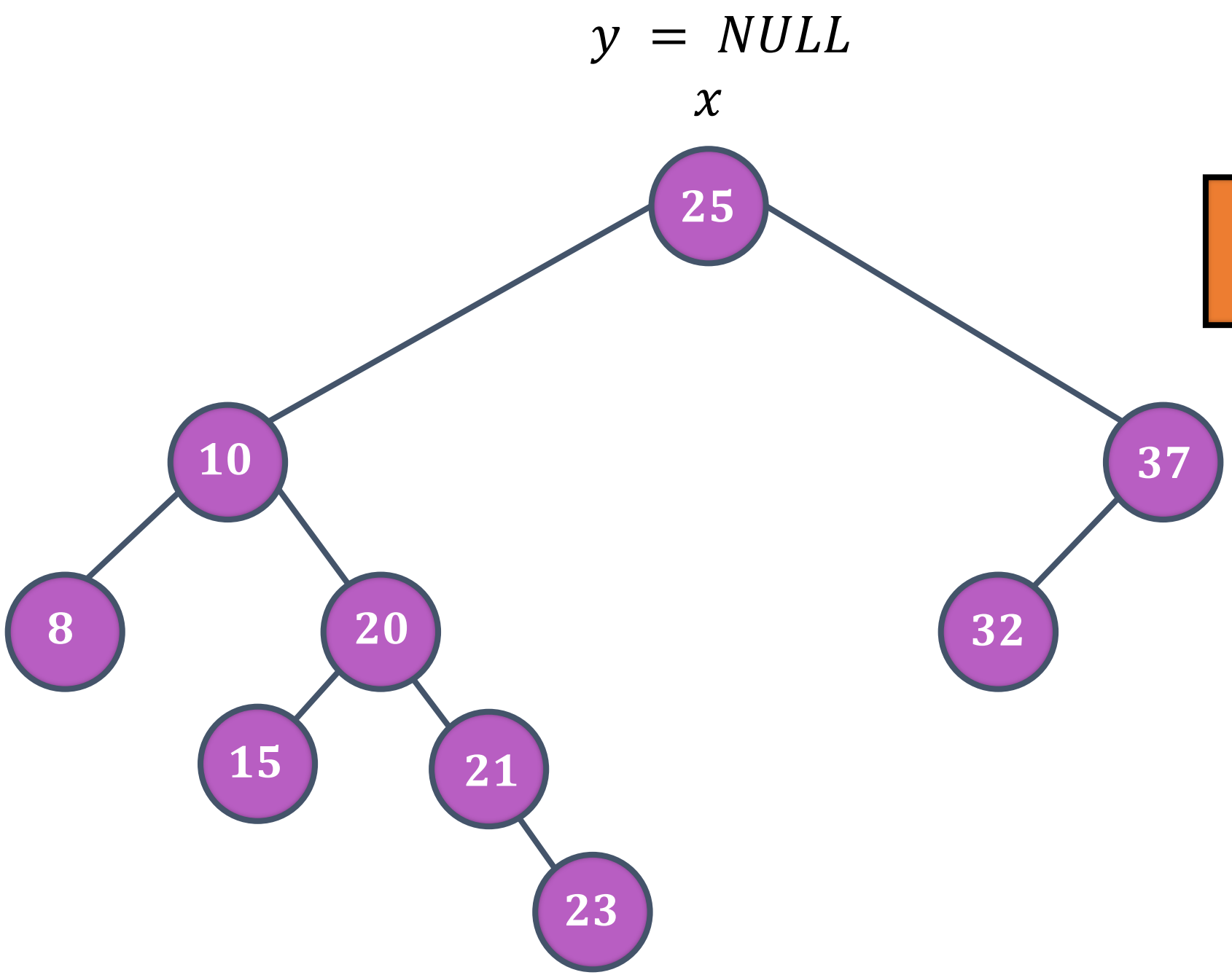
פעולת הכנסה

Insert 20



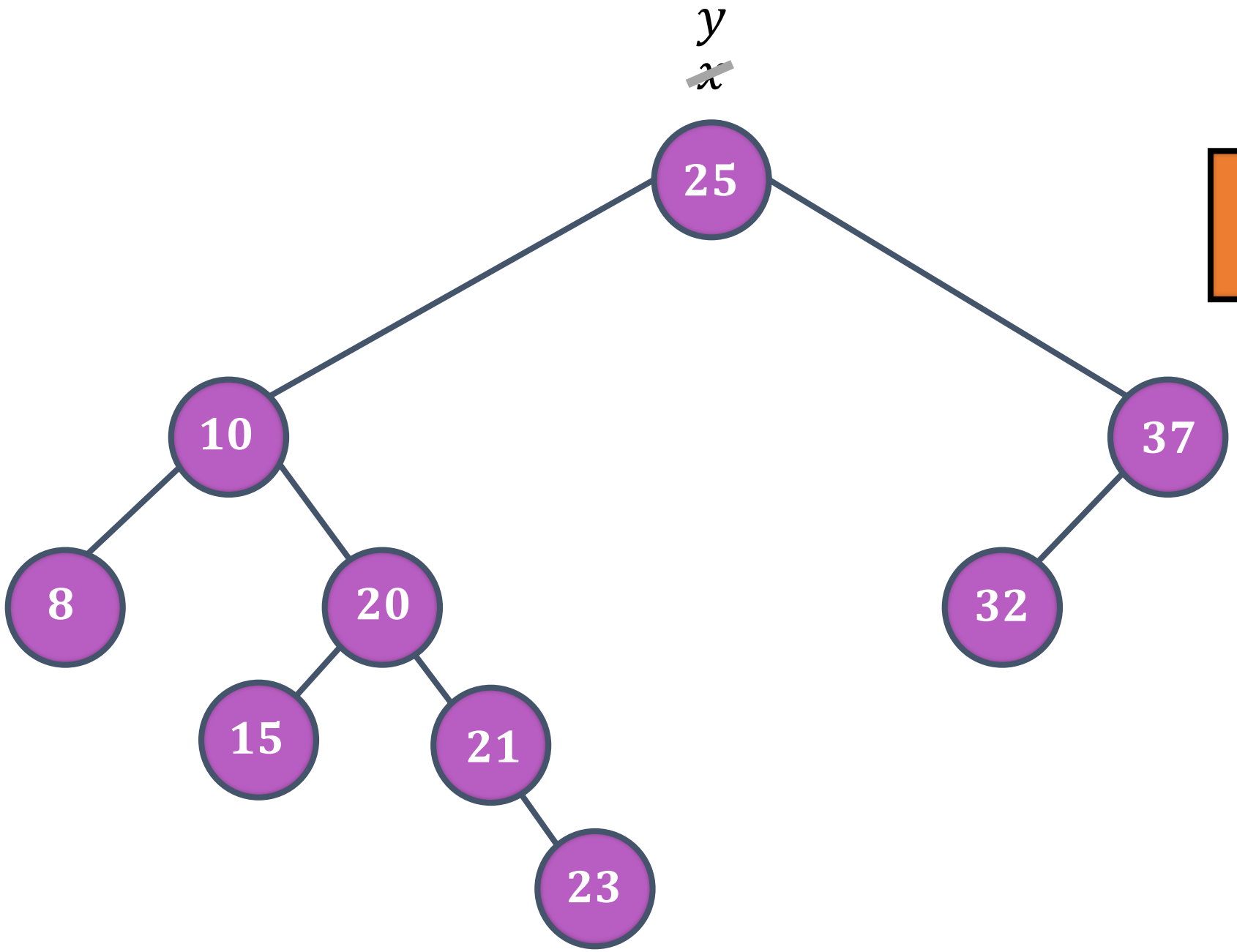
פעולת הכנסה

Insert 20



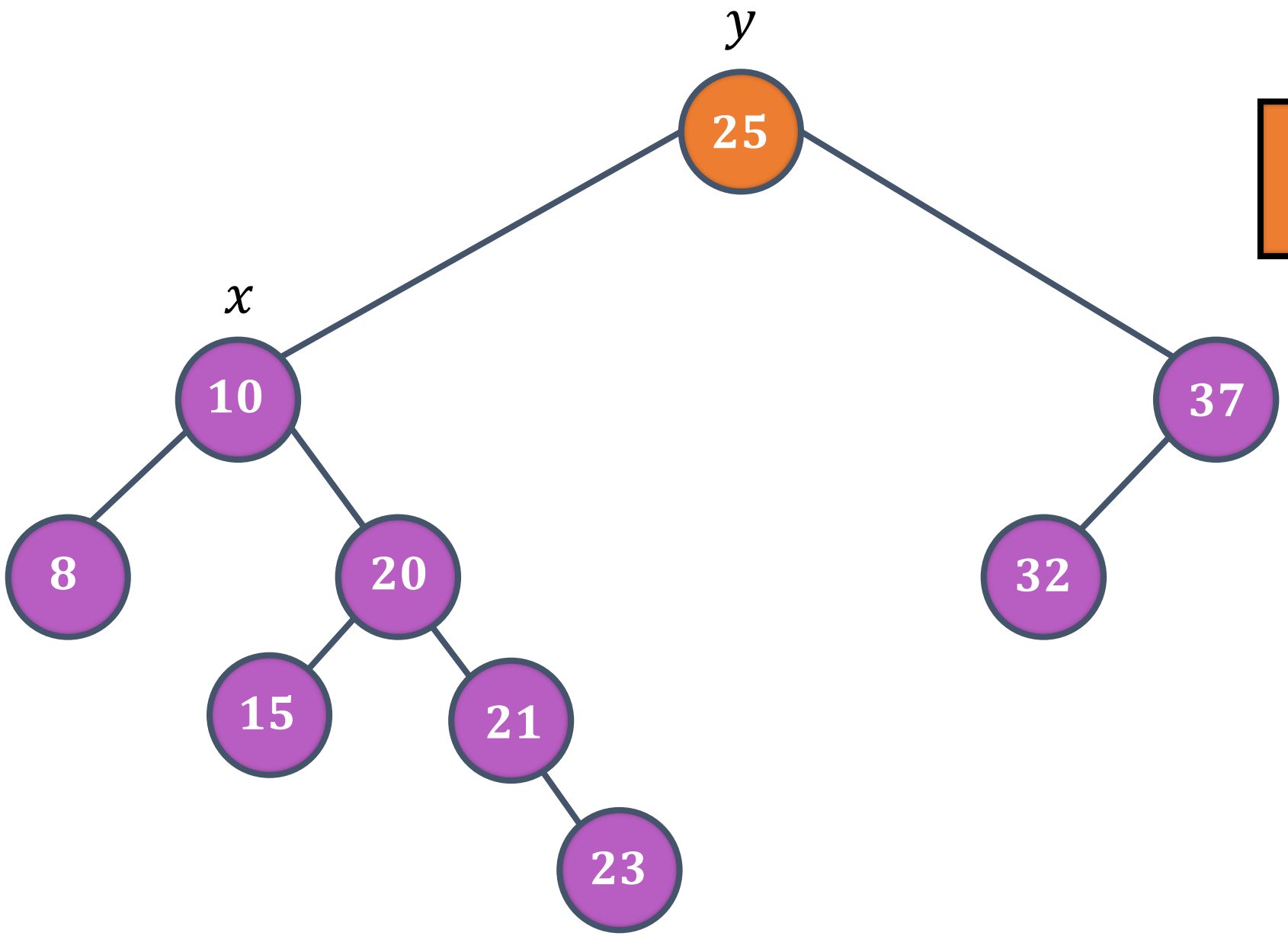
פעולת הכנסה

Insert 20



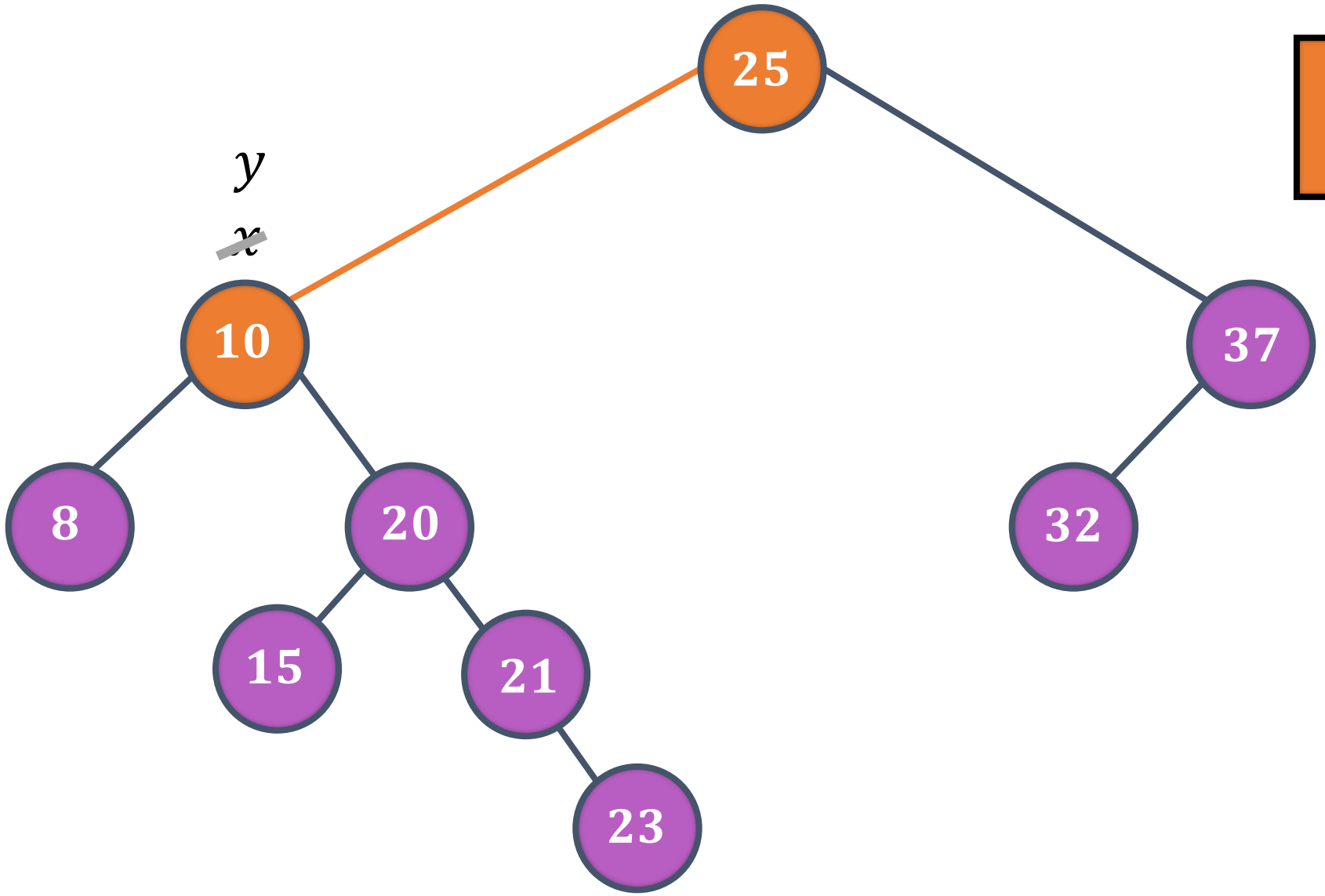
פעולת הכנסה

Insert 20



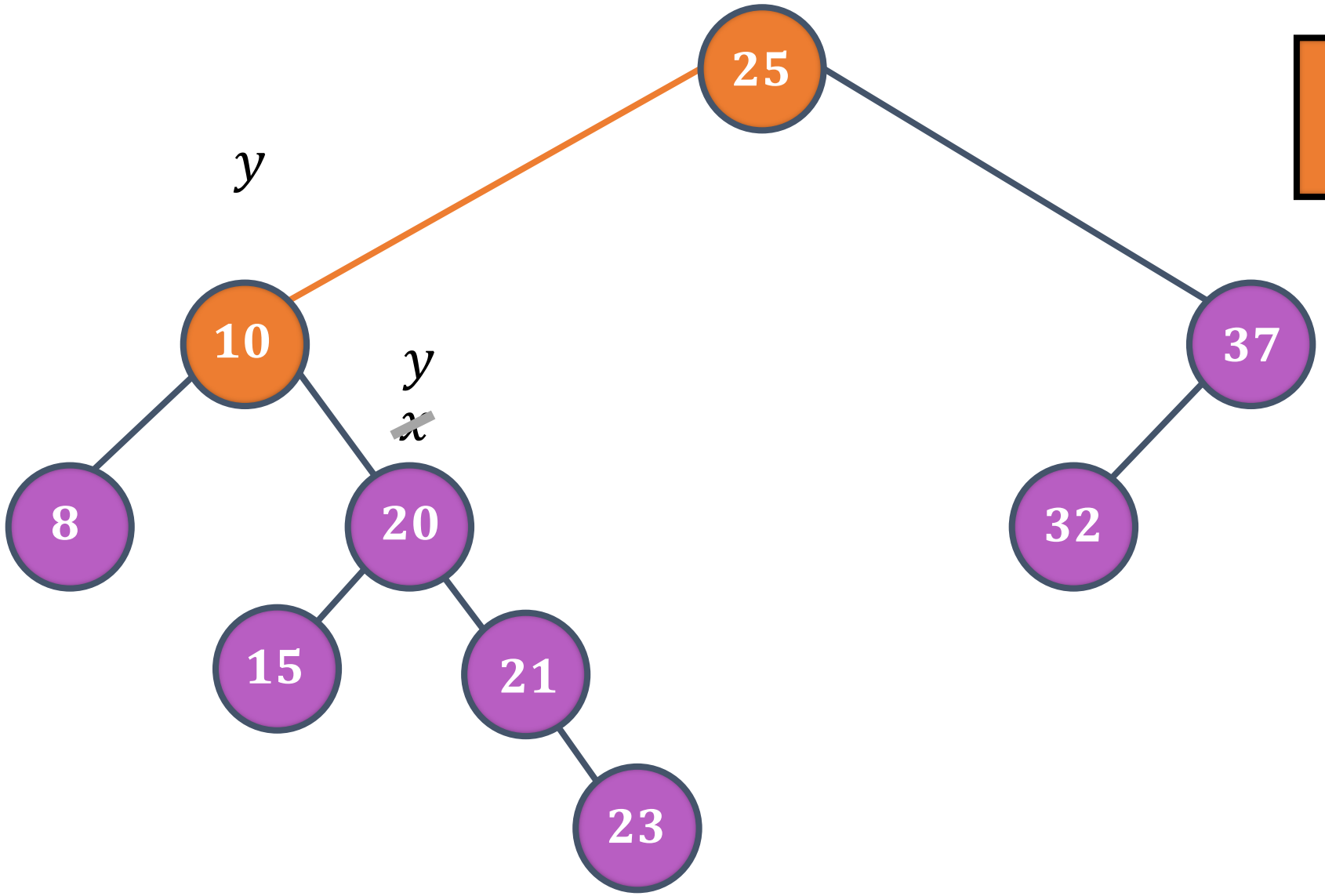
פעולת הכנסה

Insert 20



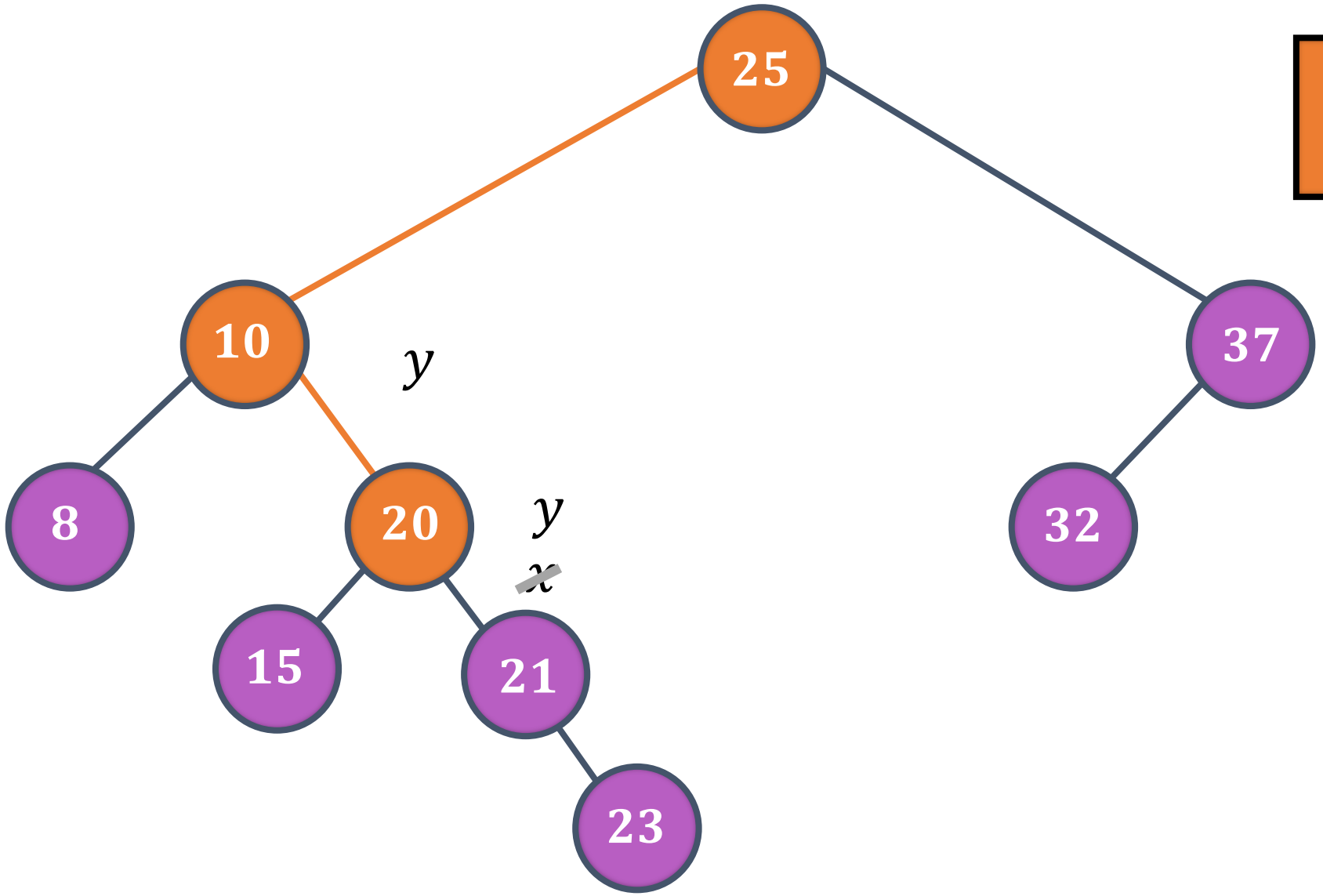
פעולת הכנסה

Insert 20



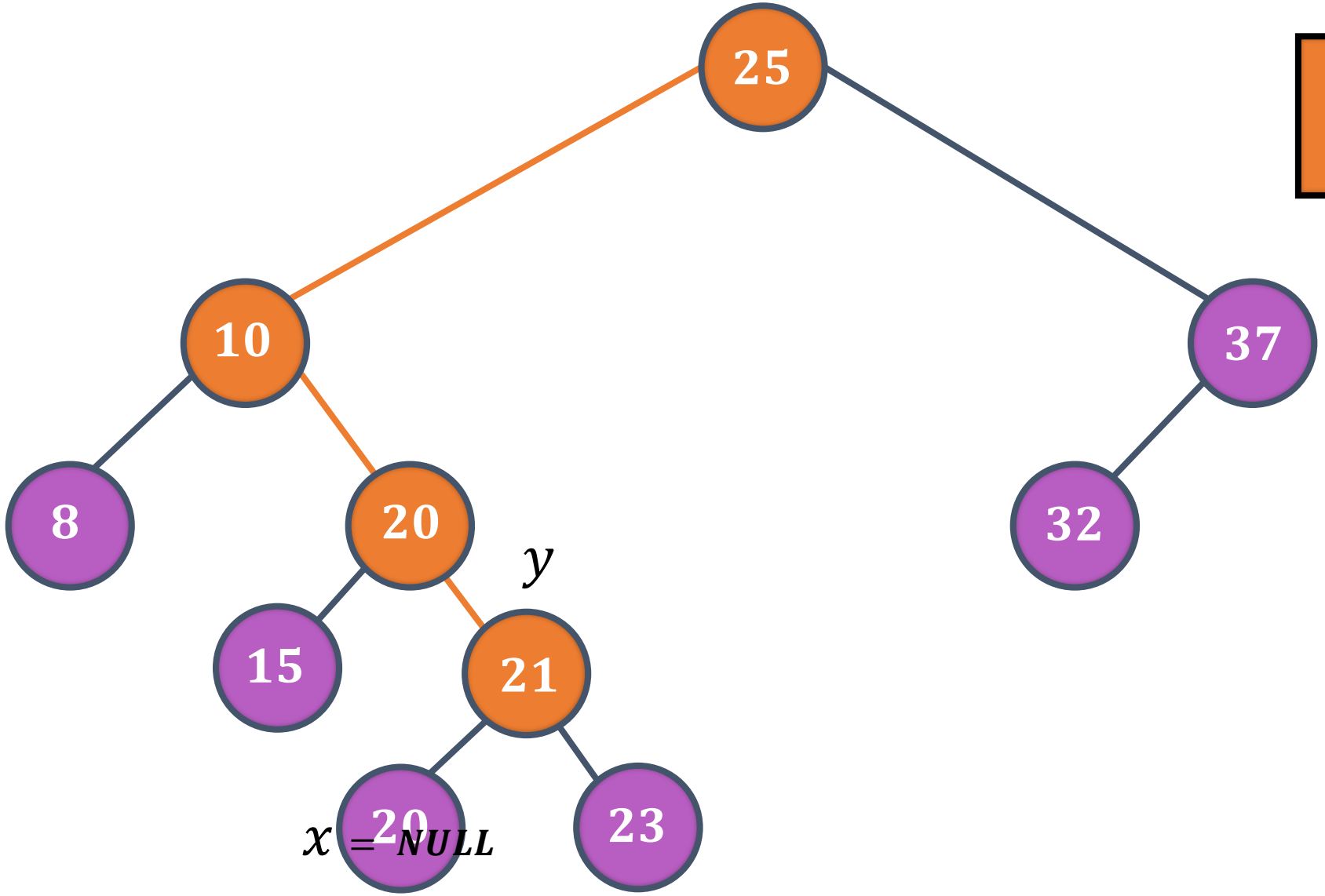
פעולת הכנסה

Insert 20



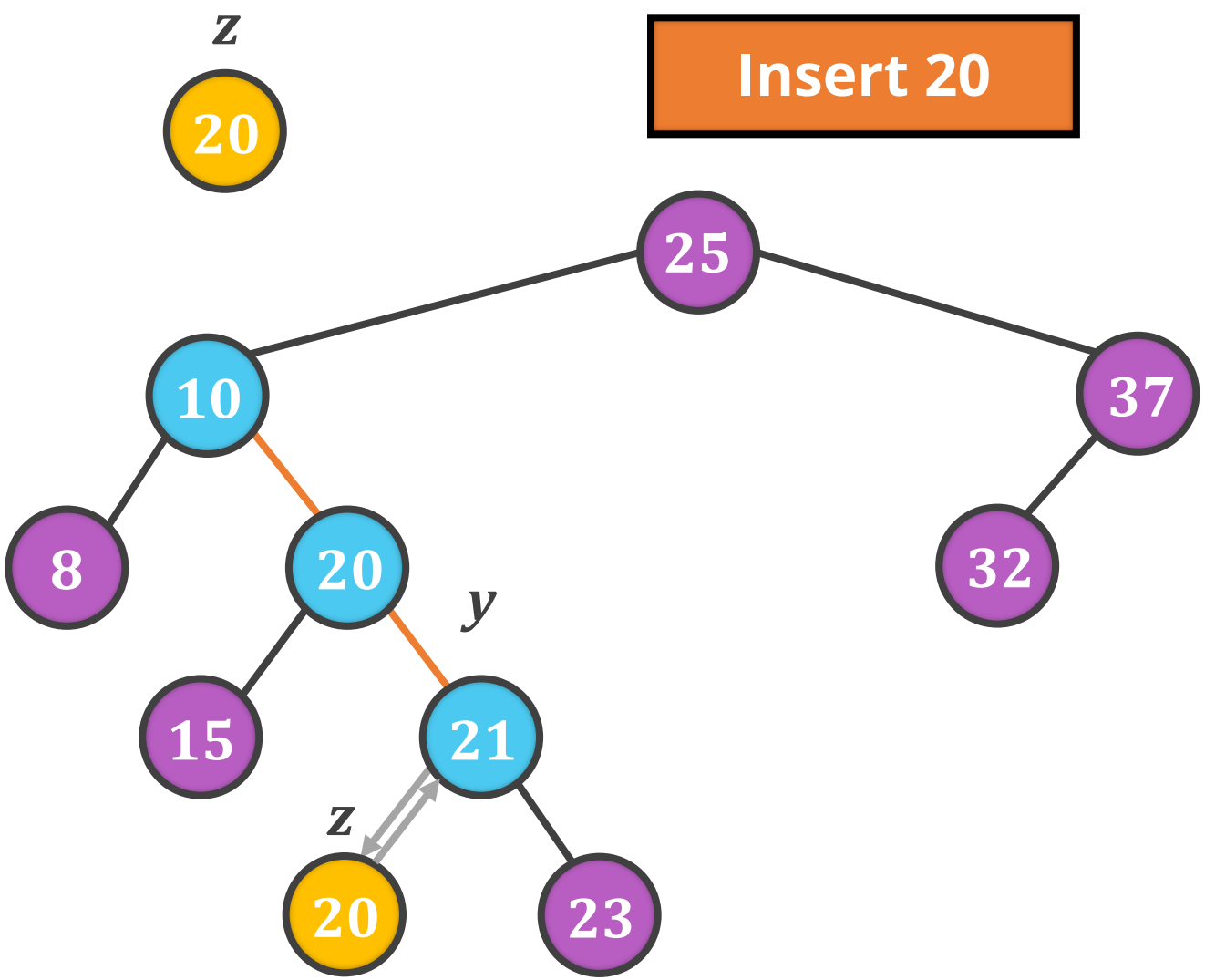
פעולת הכנסה

Insert 20



הכנסה פסאודו קוד

```
Insert (T, z)
y ← NULL
x ← T.root
while (x ≠ NULL)
        y ← x
        if (z.key < x.key)
                x ← x.left
        else x ← x.right
z.p ← y
if (y = NULL) // T was empty
        T.root ← z
elseif (z.key < y.key)
        y.left ← z
else y.right ← z
```



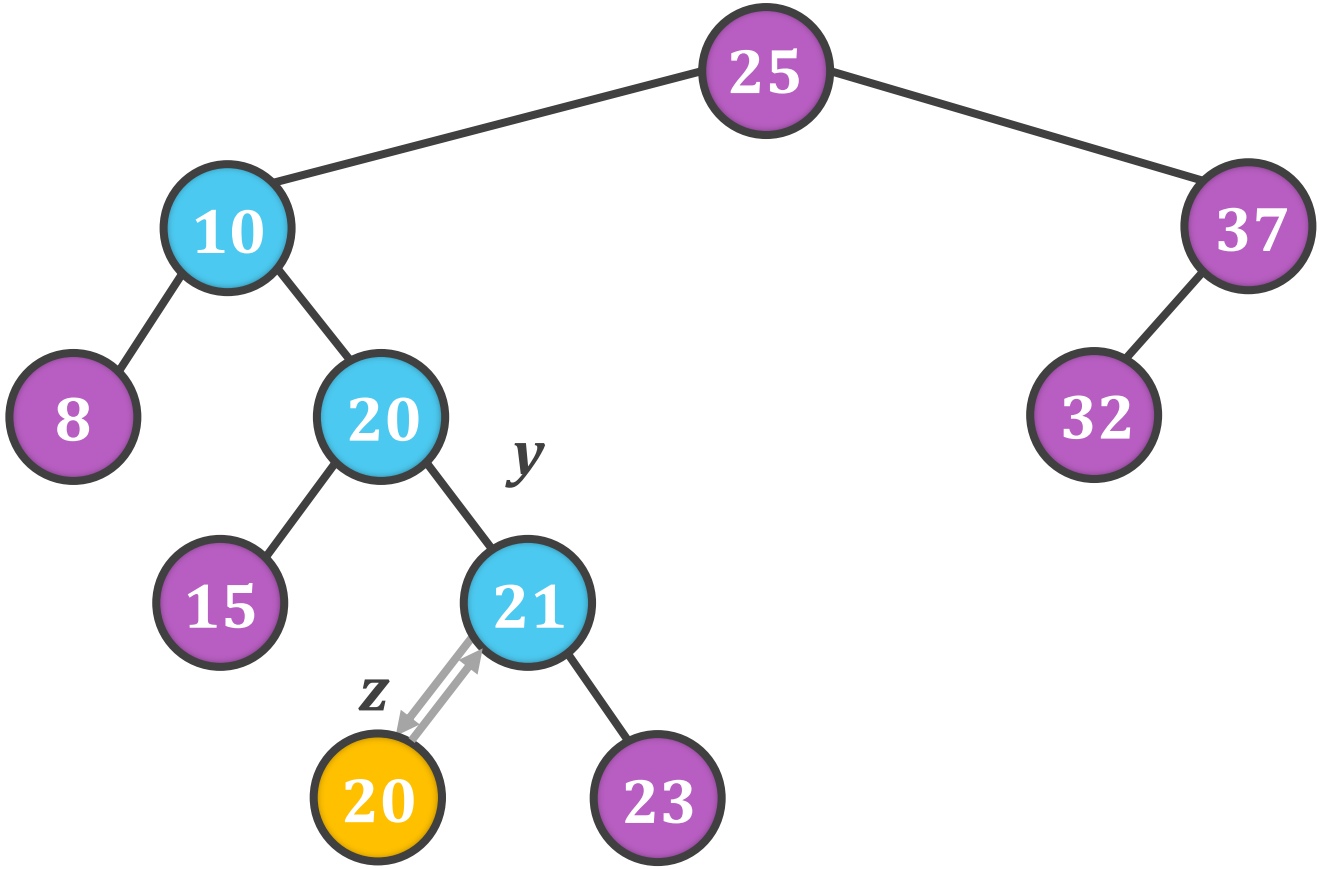
? זמן ריצה במקרה הגרוע של פעולת $Insert$ בעץ חיפוש בינארי בעל n צמתים וגובה h הוא:

1. $\Theta(n)$
2. $\Theta(1)$
3. $\Theta(h)$
4. $\Theta(\log n)$

הכנסה פסאודו קוד

```
Insert(T, z)
y ← NULL
x ← T.root
while (x ≠ NULL)
        y ← x
        if (z.key < x.key)
                x ← x.left
        else x ← x.right
z.p ← y
if (y = NULL) // T was empty
        T.root ← z
elseif (z.key < y.key)
        y.left ← z
else y.right ← z
```

Insert 20

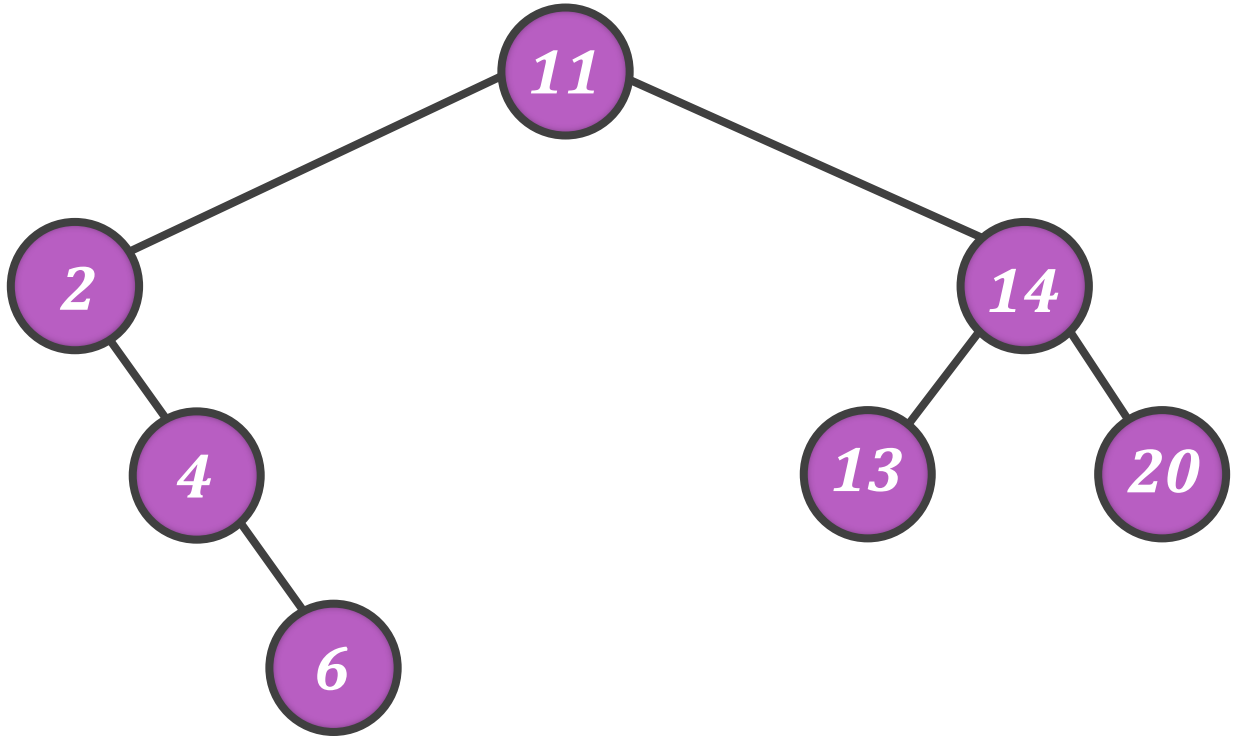


זמן ריצה הוא $\Theta(h)$, כאשר h הוא גובה העץ

מפתחות הבאים הוכנסו לעץ חיפוש בינארי ריק מהתחלה:
20, 13, 14, 6 , 4 ,2 ,11 (משמאל לימין). מה הוא גובה העץ שהתקבל?

1. 2
2. 3
3. 4
4. 6

מפתחות הבאים הוכנסו לעץ חיפוש בינארי ריק מהתחלה:
11, 2, 4 , 6,14, 13, 20 (משמאל לימין). מה הוא גובה העץ שהתקבל?



.1 2

.2 3

.3 4

.4 6

Ordered Dictionary

$Search(k)$

$Insert(x)$

$Delete(x)$

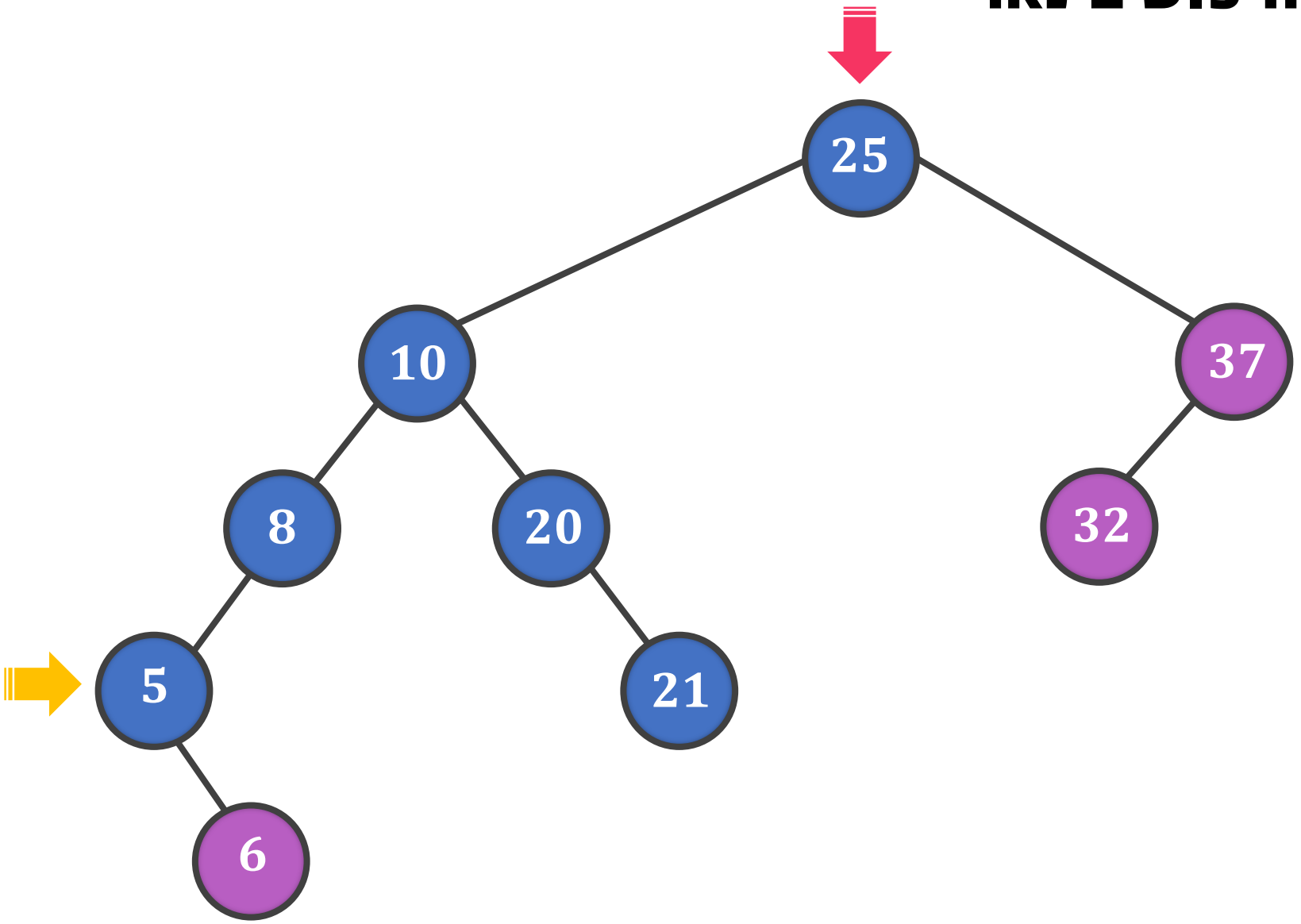
$Min()$

$Max()$

$Successor(x)$

$Predecessor(x)$

מציאת מינימום בעץ חיפוש בינארי



מינימום פסאודו קוד

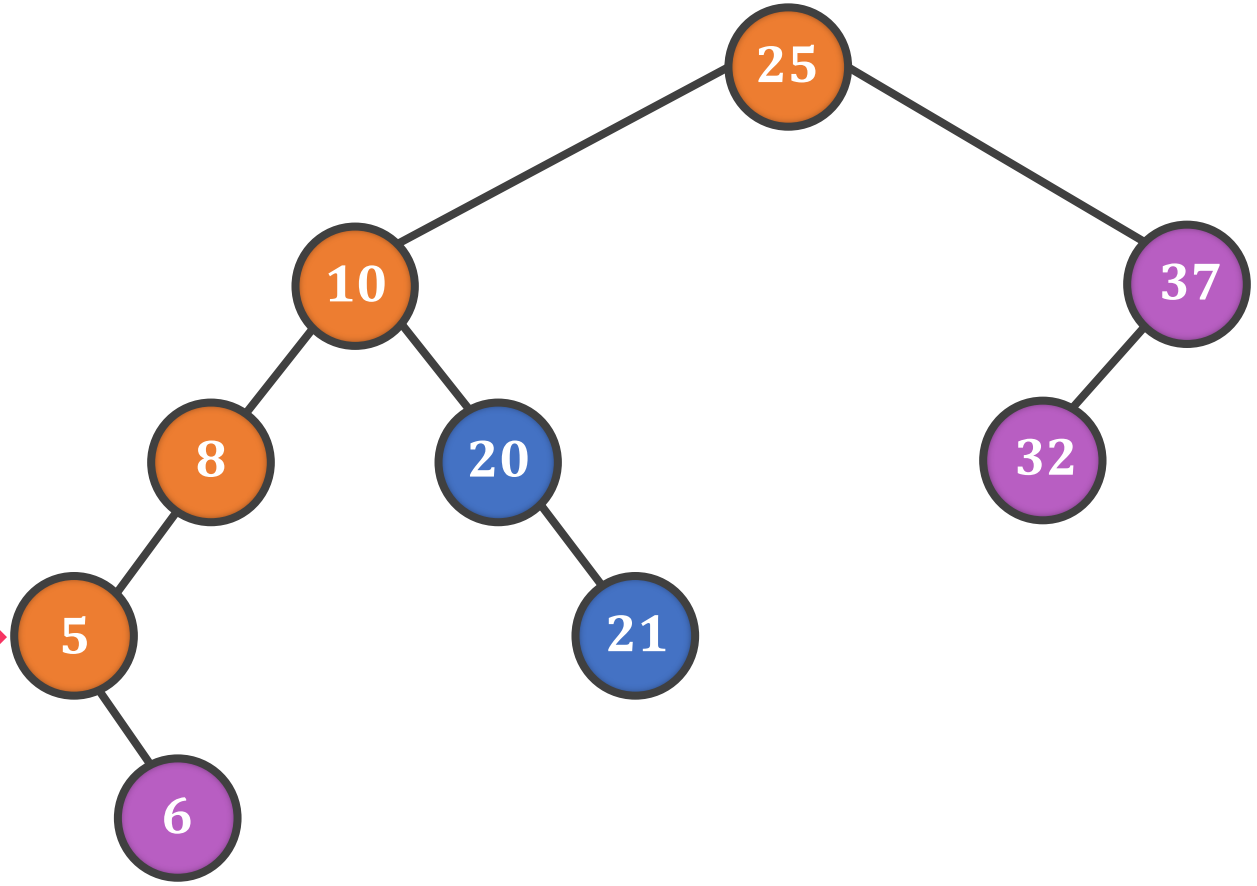
$Min\ (x)$

1 while $x.left \neq NULL$

2 $x \leftarrow x.left$

3 return x

קריאה חיצונית $Min(T.root)$



השלימו את המשפט:
זמן ריצה במקרה הגרוע של פעולת Min בעץ חיפוש בינארי בעל n צמתים וגובה h הוא:

1. $\Theta(n)$
2. $\Theta(1)$
3. $\Theta(h)$
4. $\Theta(\log n)$

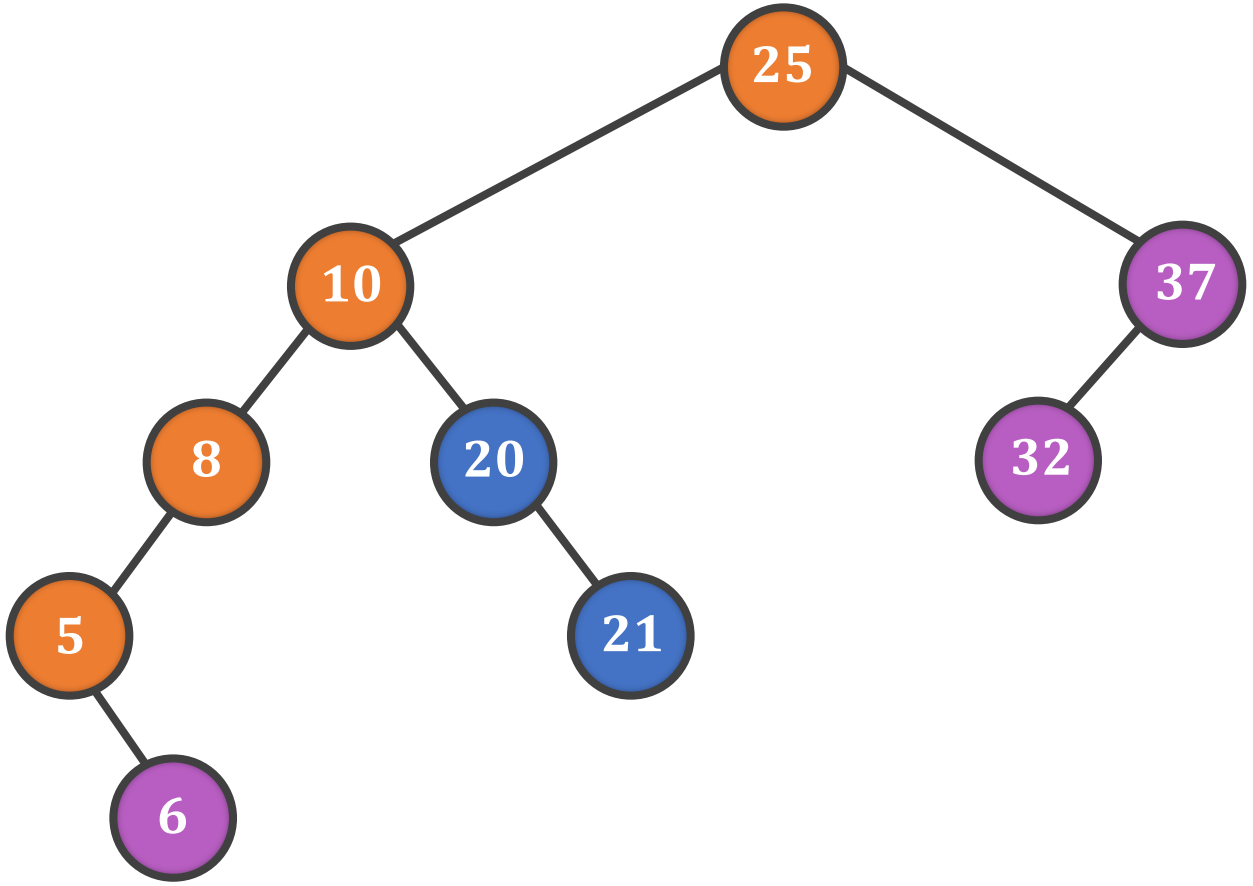
מינימום פסאודו קוד

$Min\ (x)$

1 while $x.left \neq NULL$

2 $x \leftarrow x.left$

3 return x



זמן ריצה הוא $\Theta(h)$, כאשר h הוא גובה העץ

מקסימום פסאודו קוד

Max

$Min\ (x)$

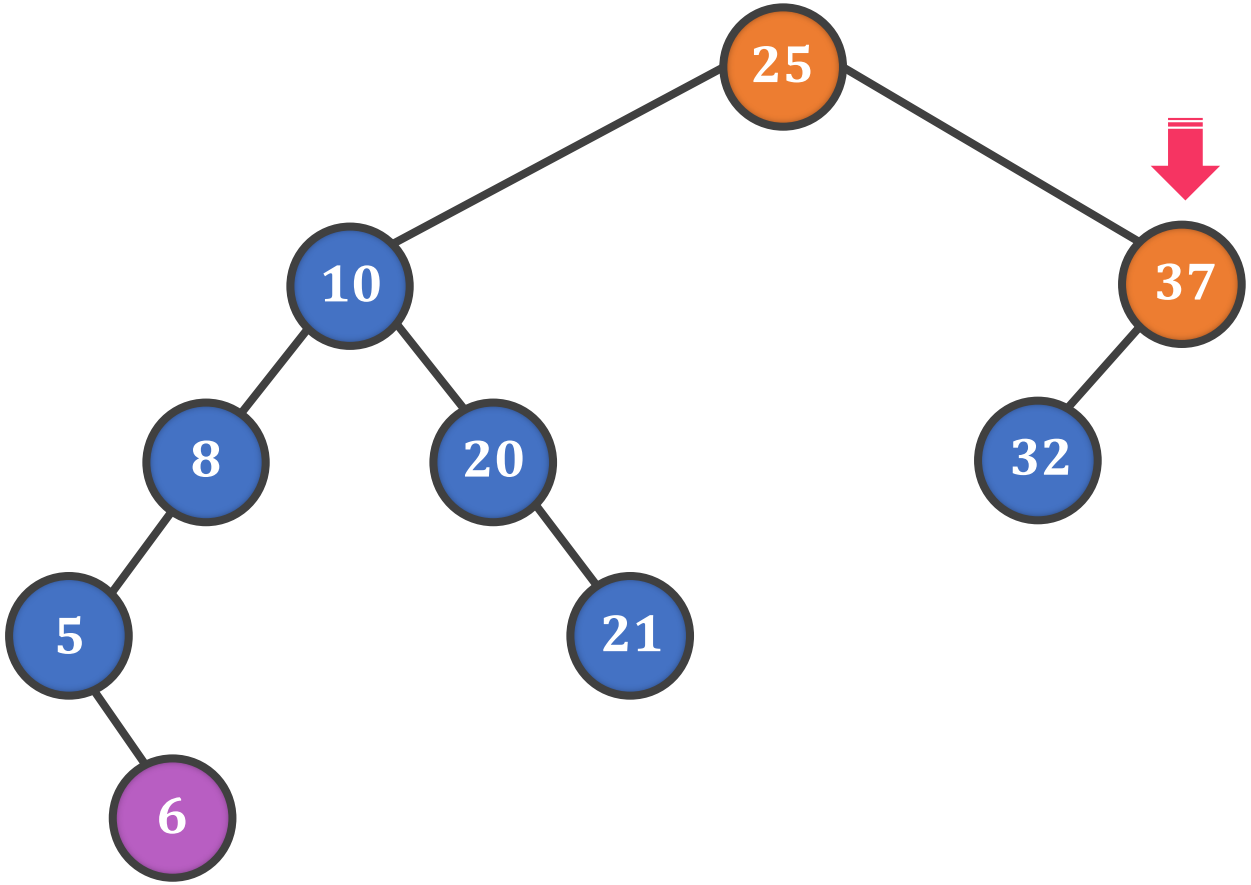
$x.right$

1 while $x.left \neq NULL$

$x.right$

2 $x \leftarrow x.left$

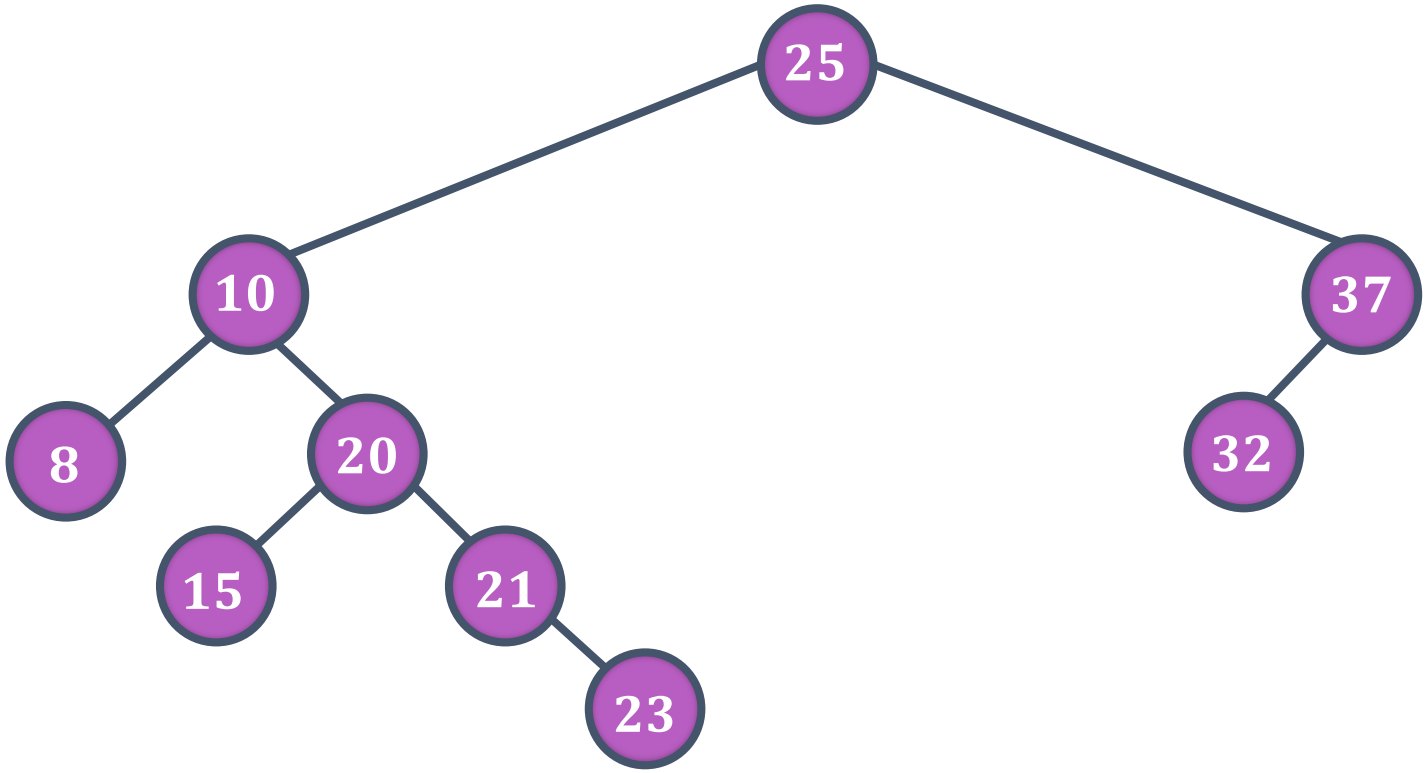
3 return x



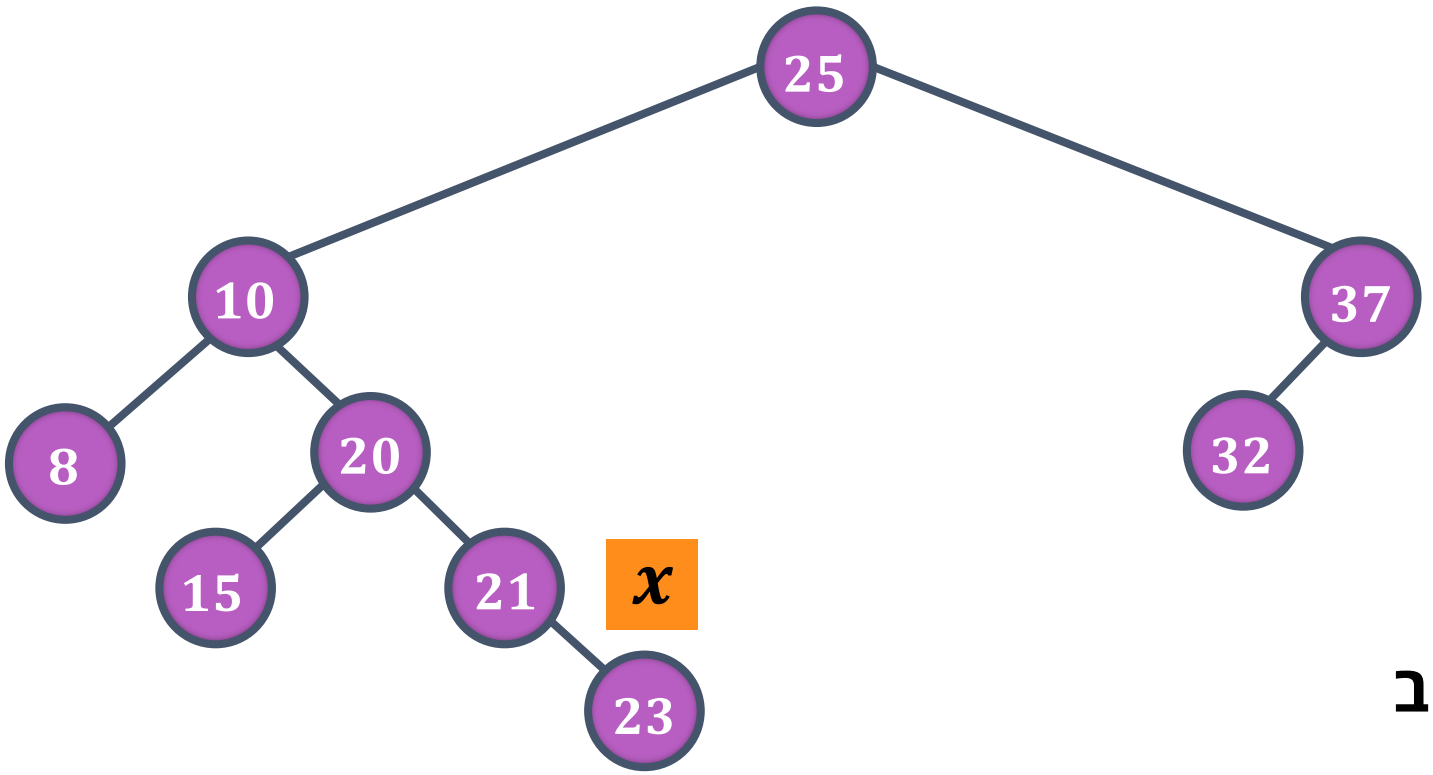
זמן ריצה הוא $\Theta(h)$, כאשר h הוא גובה העץ

מציאות עוקב לצומת הגדרה

אם כל המפתחות שונים זה מזה, העוקב ($Successor$) לצומת x הוא הצומת בעל מפתח הקטן ביותר הגדול מ-$x.key$



מיהו צומת עוקב לצומת x בעל מפתח 23 ?

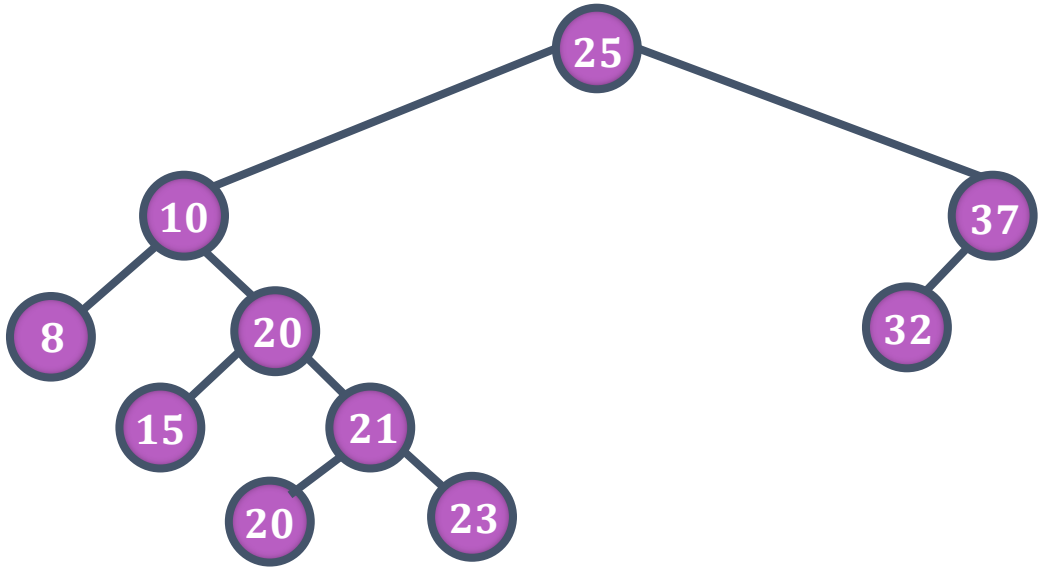


1. 21
2. 32
3. 25
4. אין לצומת x עוקב

מציאות עוקב לצומת הגדרה

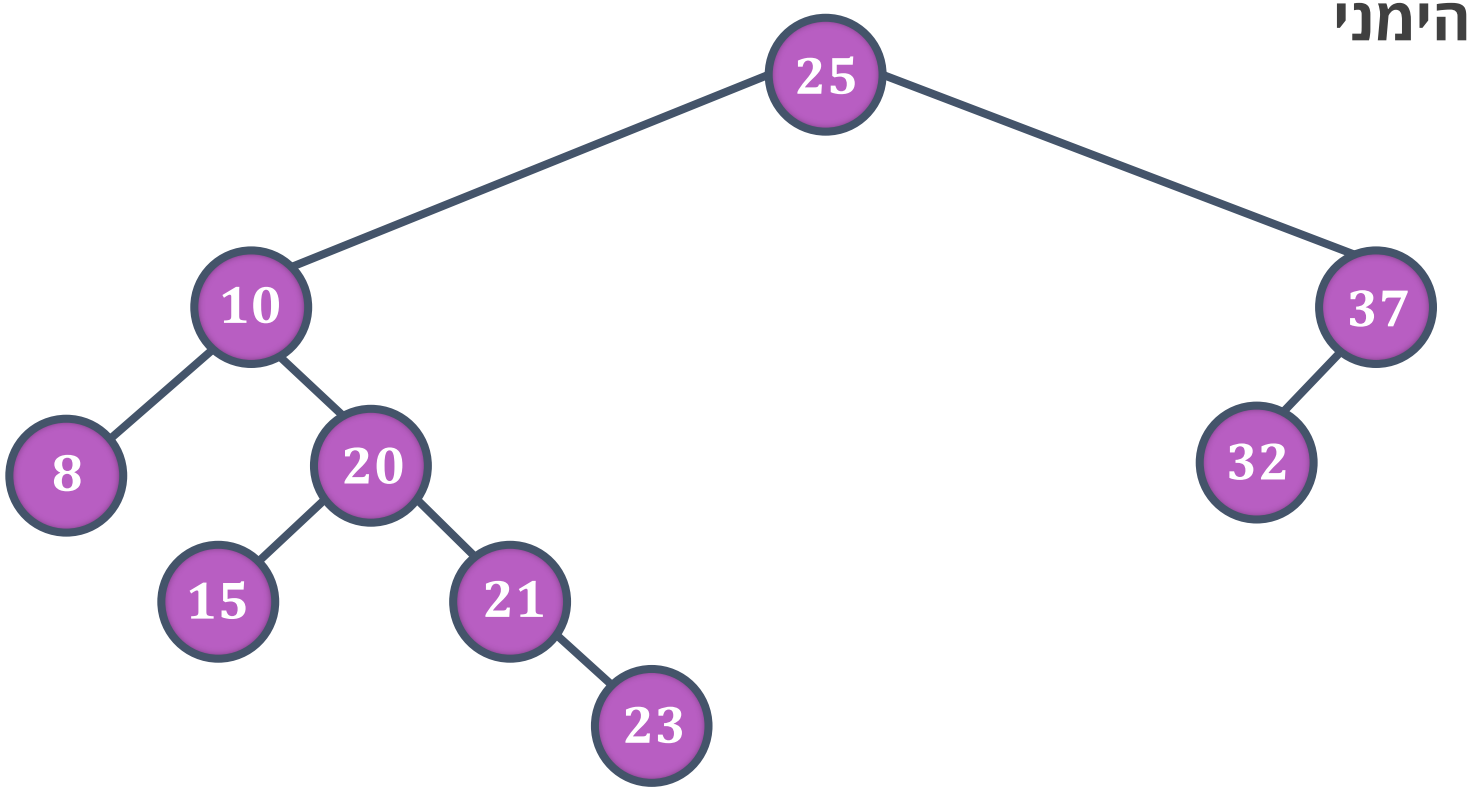
אם כל המפתחות שונים זה מזה, העוקב ($Successor$) לצומת x הוא הצומת בעל מפתח הקטן ביותר הגדול מ-$x.key$

במידה והמפתחות יכולים לחזור על עצמם, העוקב לצומת x הוא צומת הבא בסדר הממוין הנקבע על ידי סריקת $inorder$ של העץ



מציאות עוקב לצומת

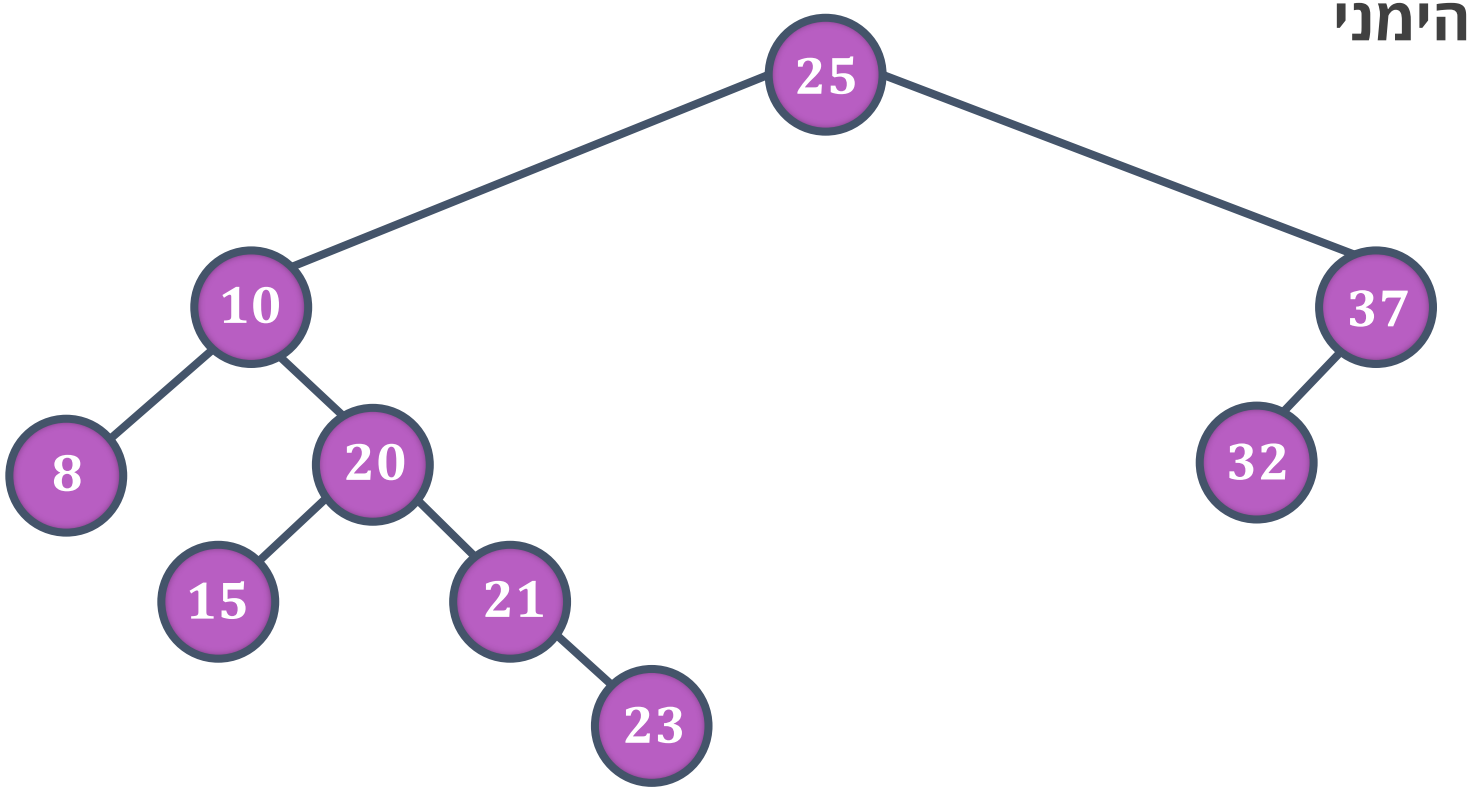
- מקרה פשוט
  - לצומת x יש תת עץ ימני
  - החזר מינימאלי בתת העץ הימני



מציאות עוקב לצומת

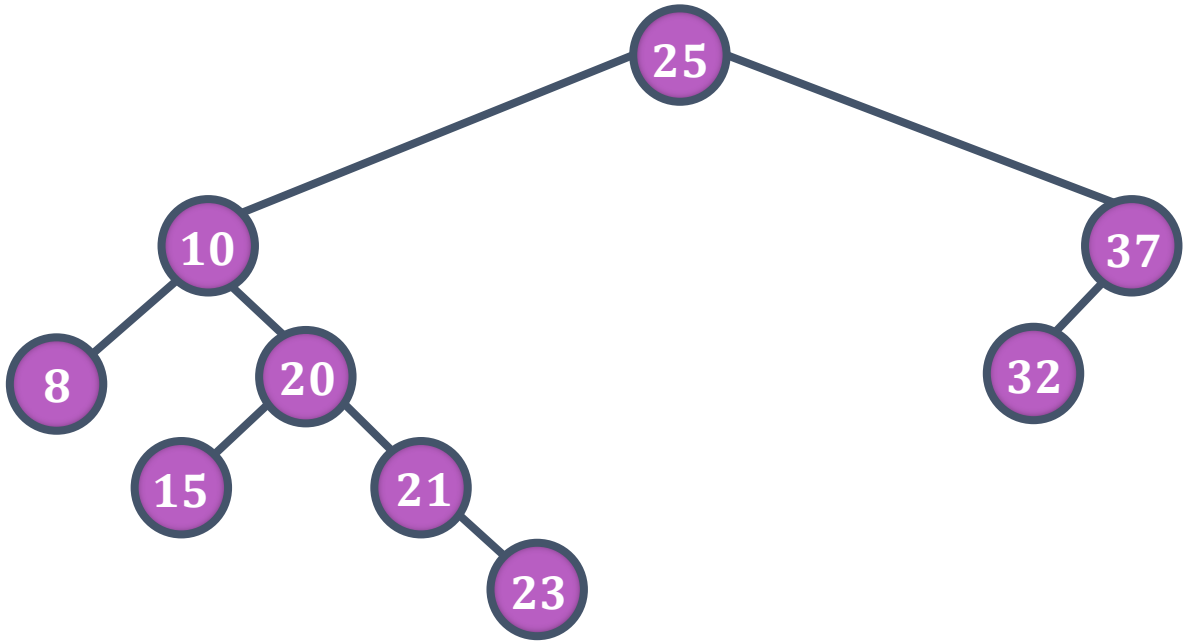
- מקרה פשוט
  - לצומת x יש תת עץ ימני
    - החזר מינימאלי בתת העץ הימני

- אחרת:
  - יש לעלות במסלול לשורש עד "לפנייה ראשונה ימינה"



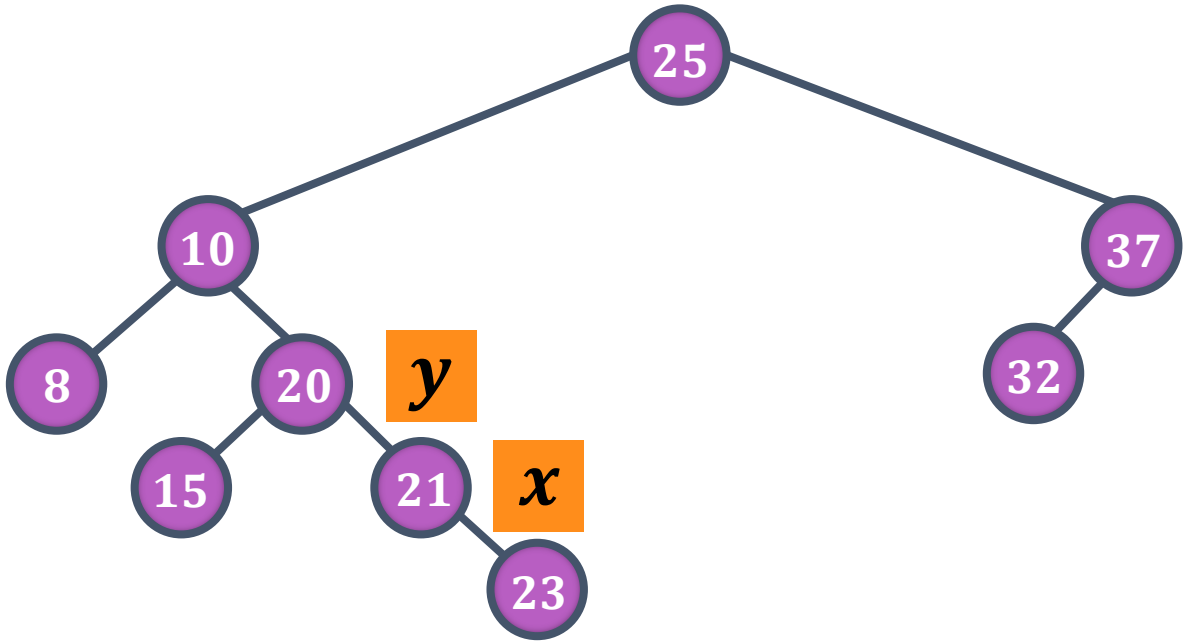
מציאת עוקב לצומת פסאודו קוד

```
Successor (x)
if x.right ≠ NULL
        return Min(x.right)
y ← x.parent
while y ≠ NULL and x = y.right
        x ← y
        y ← y.parent
return y
```



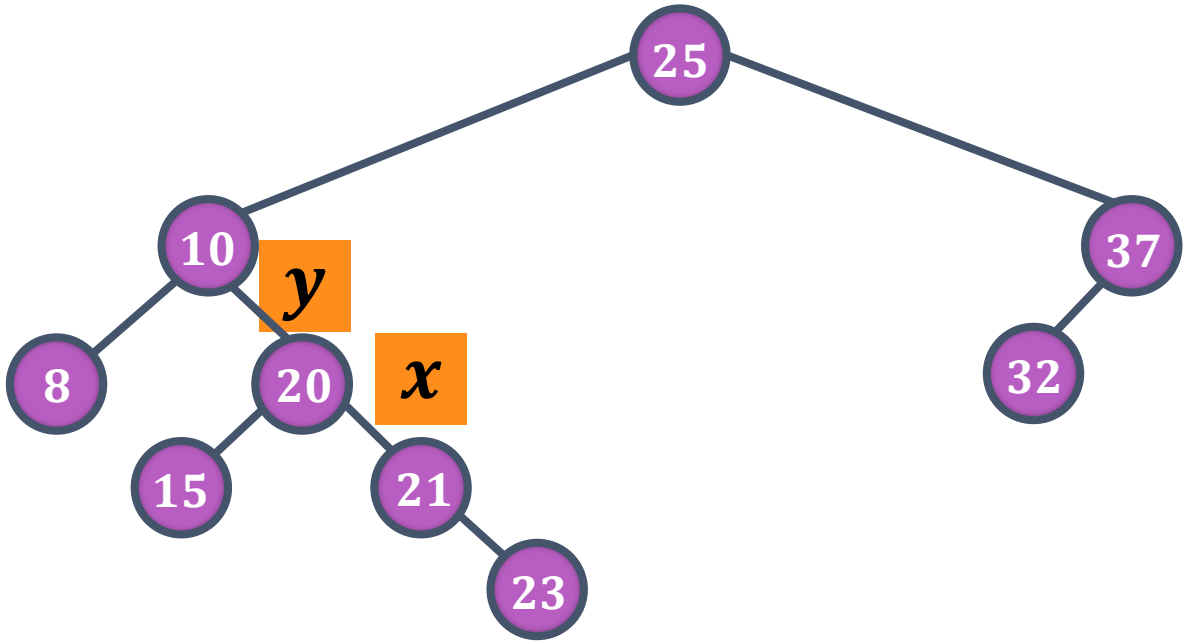
מציאת עוקב לצומת פסאודו קוד

```
Successor (x)
if x.right ≠ NULL
        return Min(x.right)
y ← x.parent
while y ≠ NULL and x = y.right
        x ← y
        y ← y.parent
return y
```



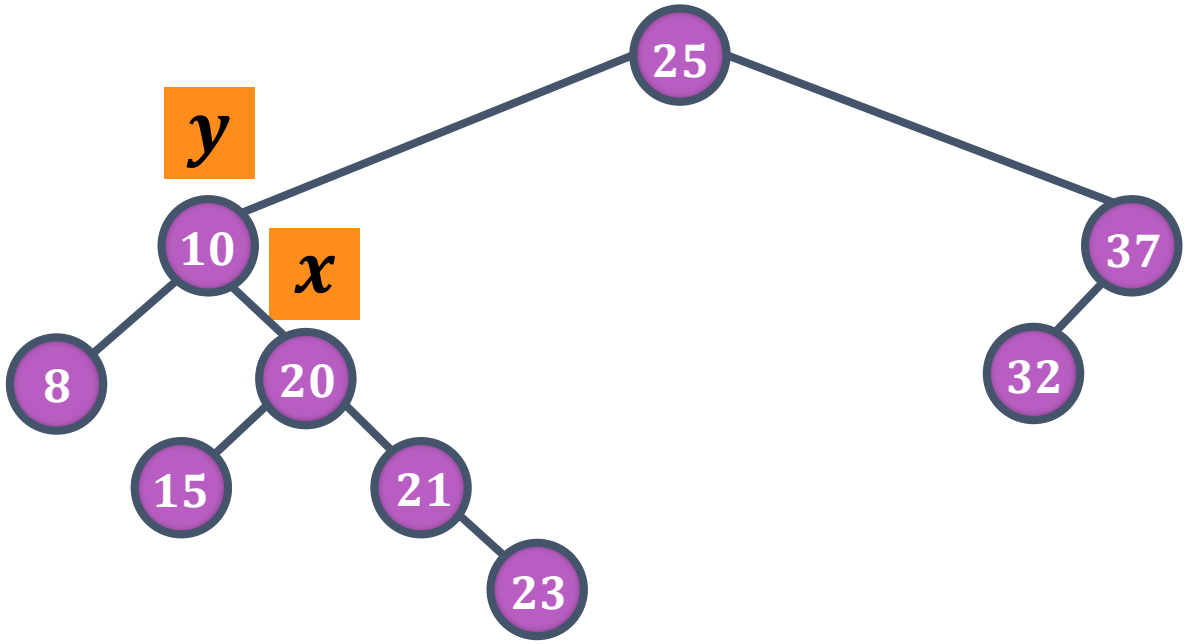
מציאת עוקב לצומת פסאודו קוד

```
Successor (x)
if x.right ≠ NULL
        return Min(x.right)
y ← x.parent
while y ≠ NULL and x = y.right
        x ← y
        y ← y.parent
return y
```



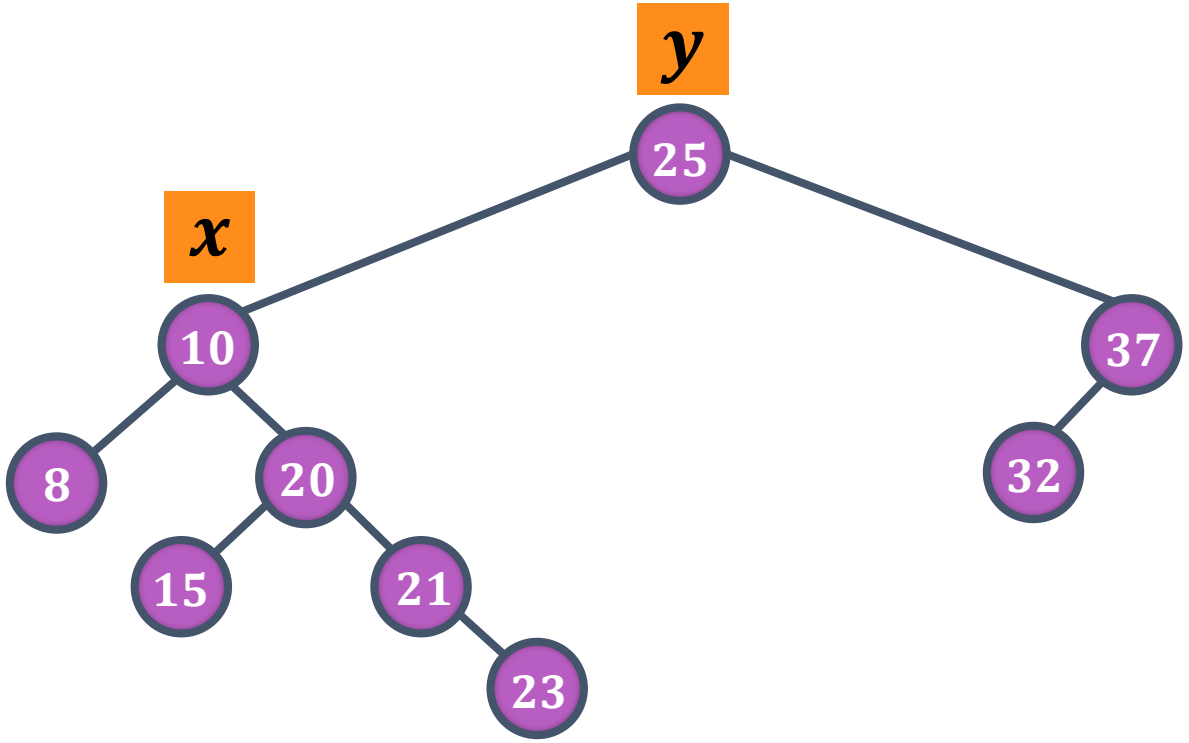
מציאת עוקב לצומת פסאודו קוד

```
Successor (x)
if x.right ≠ NULL
        return Min(x.right)
y ← x.parent
while y ≠ NULL and x = y.right
        x ← y
        y ← y.parent
return y
```

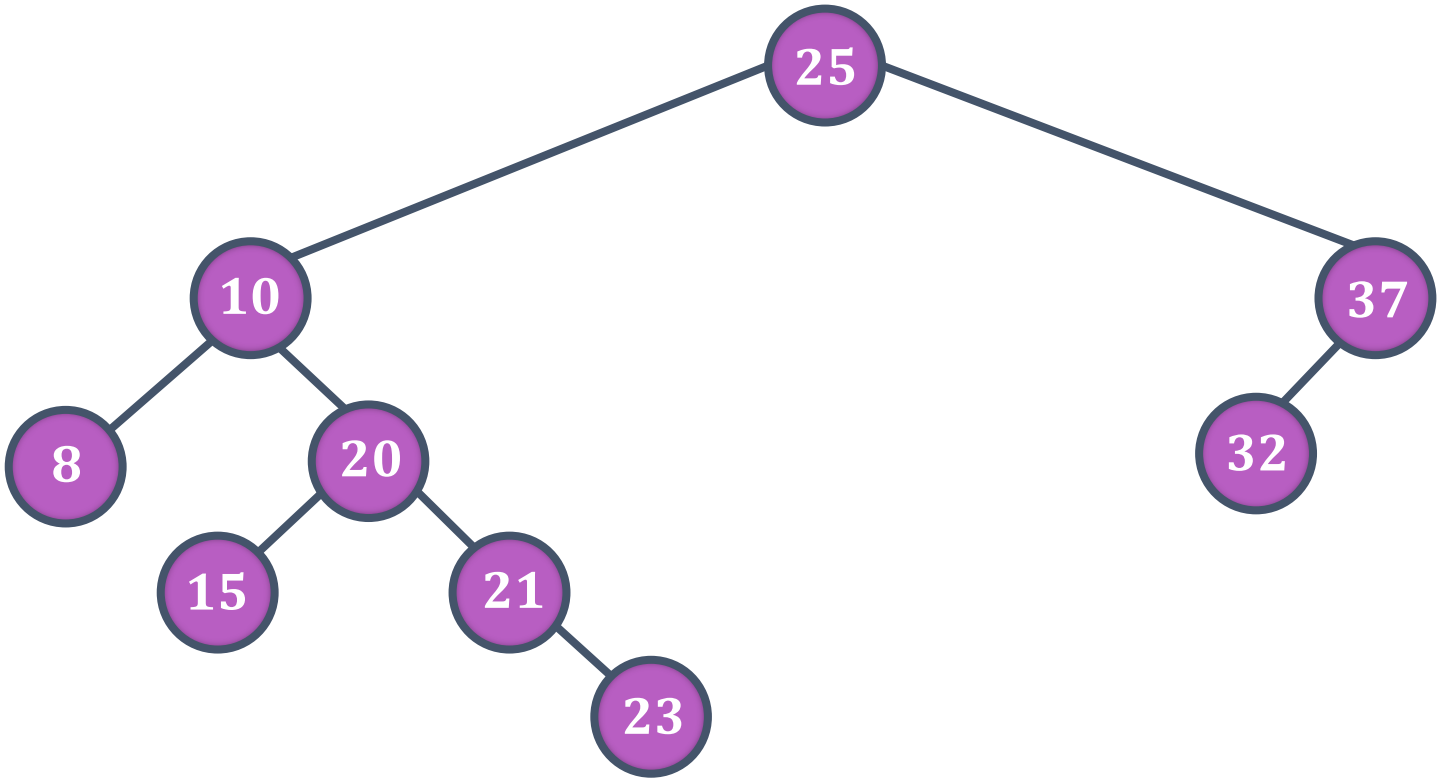


מציאת עוקב לצומת פסאודו קוד

```
Successor (x)
if x.right ≠ NULL
        return Min(x.right)
y ← x.parent
while y ≠ NULL and x = y.right
        x ← y
        y ← y.parent
return y
```



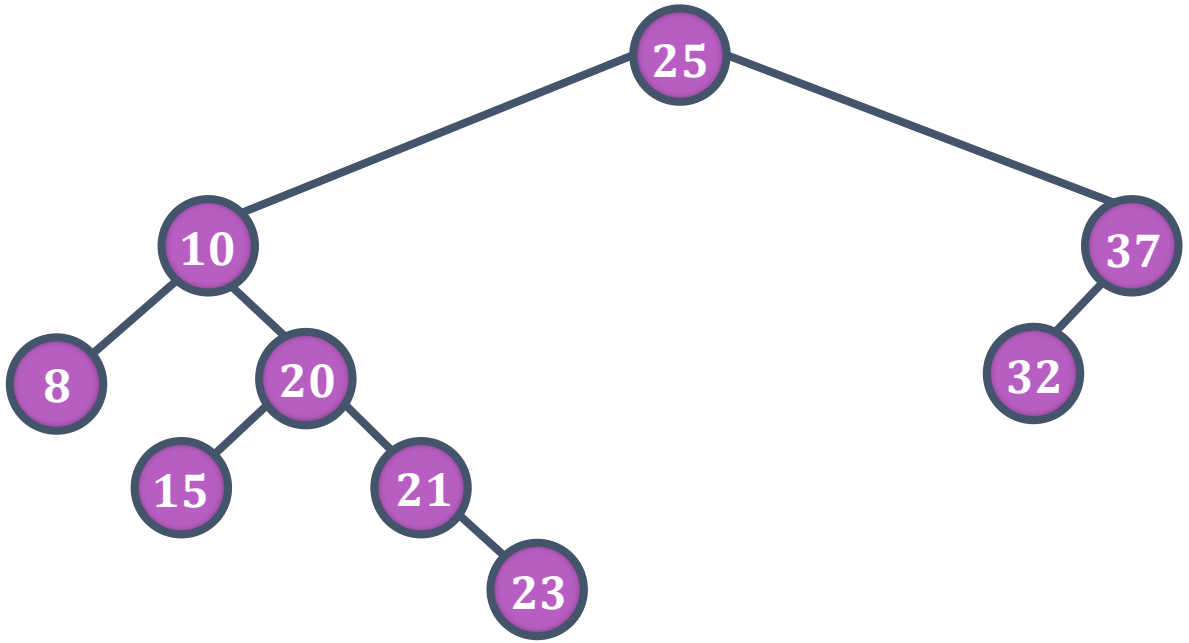
השלימו את המשפט:
זמן ריצה במקרה הגרוע של פעולת $Successor$ בעץ חיפוש בינארי בעל n צמתים וגובה h הוא:



1. $\Theta(n)$
2. $\Theta(1)$
3. $\Theta(h)$
4. $\Theta(\log n)$

מציאת עוקב לצומת פסאודו קוד

```
Successor (x)
if x.right ≠ NULL
        return Min(x.right)
y ← x.parent
while y ≠ NULL and x = y.right
        x ← y
        y ← y.parent
return y
```

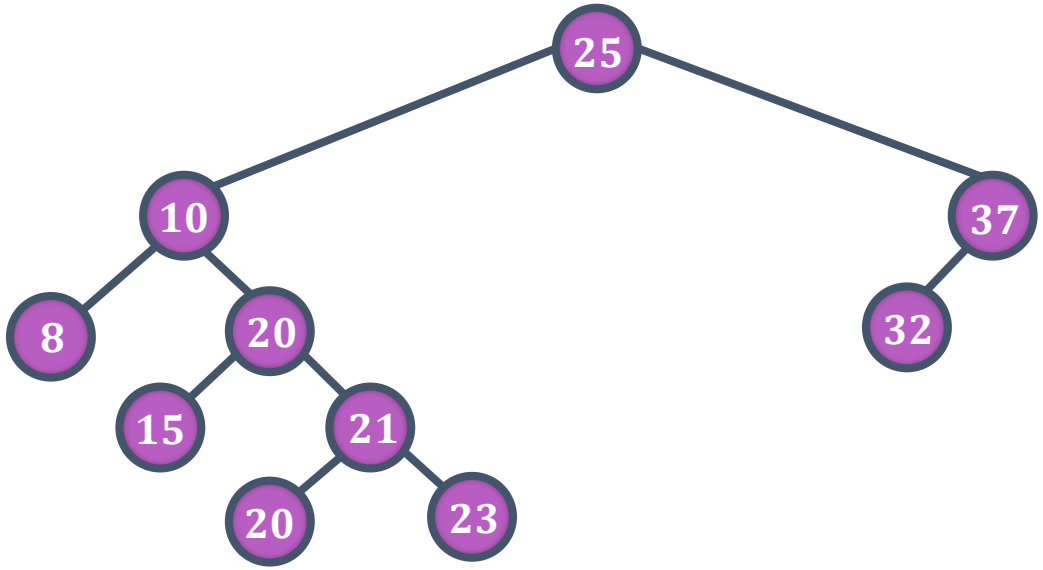


זמן ריצה הוא $\Theta(h)$, כאשר h הוא גובה העץ

קודם לצומת הגדרה

אם כל המפתחות שונים זה מזה, הקודם ($Predecessor$) לצומת x הוא הצומת בעל מפתח הגדול ביותר הקטן מ-$x.key$

במידה והמפתחות יכולים לחזור על עצמם, הקודם לצומת x הוא צומת הקודם בסדר הממוין הנקבע על ידי סריקת $inorder$ של העץ



סיכום

פעולות המילון $Predecessor, Successor, Min, Max$

ניתנות למימוש על עץ חיפוש בינארי בגובה h בזמן $O(h)$

Ordered Dictionary

$Search(k)$

$Insert(x)$

$Delete(x)$

$Min()$

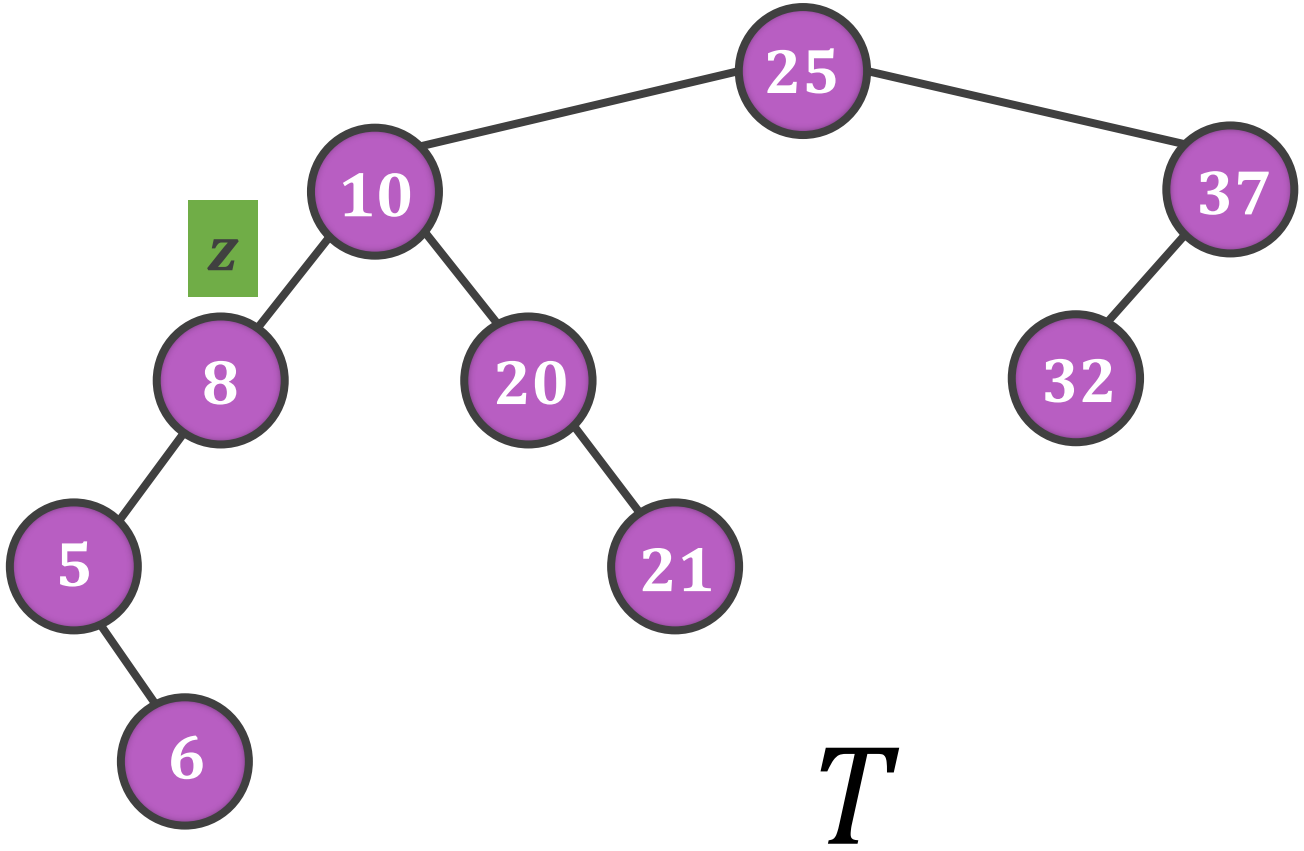
$Max()$

$Successor(x)$

$Predecessor(x)$

מחיקה בעץ חיפוש בינארי

נתון: עץ חיפוש בינארי T
מצביע לצומת z בעץ שיש למחוק

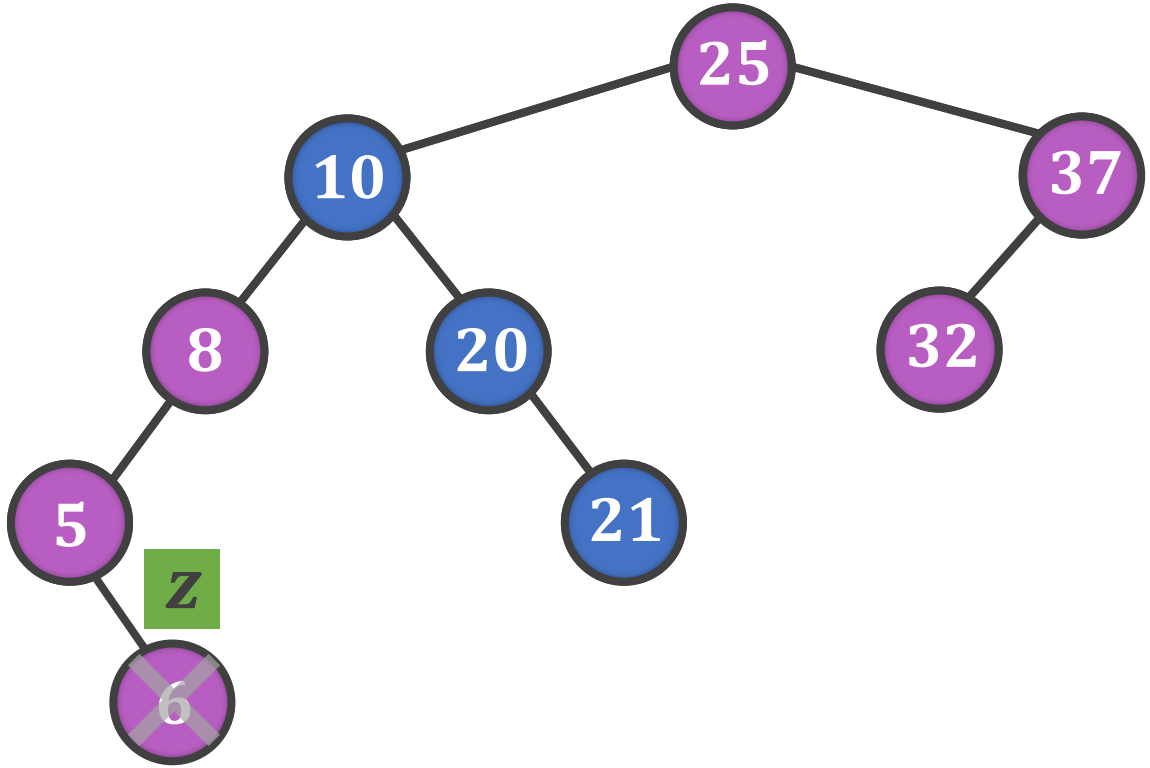


מחיקה בעץ חיפוש בינארי

ישנם שלושה מקרים:

1

ל-z אין בנים:
מעדכנים את אביו $z.parent$ כך שבנו יהיה $NULL$ ולא z

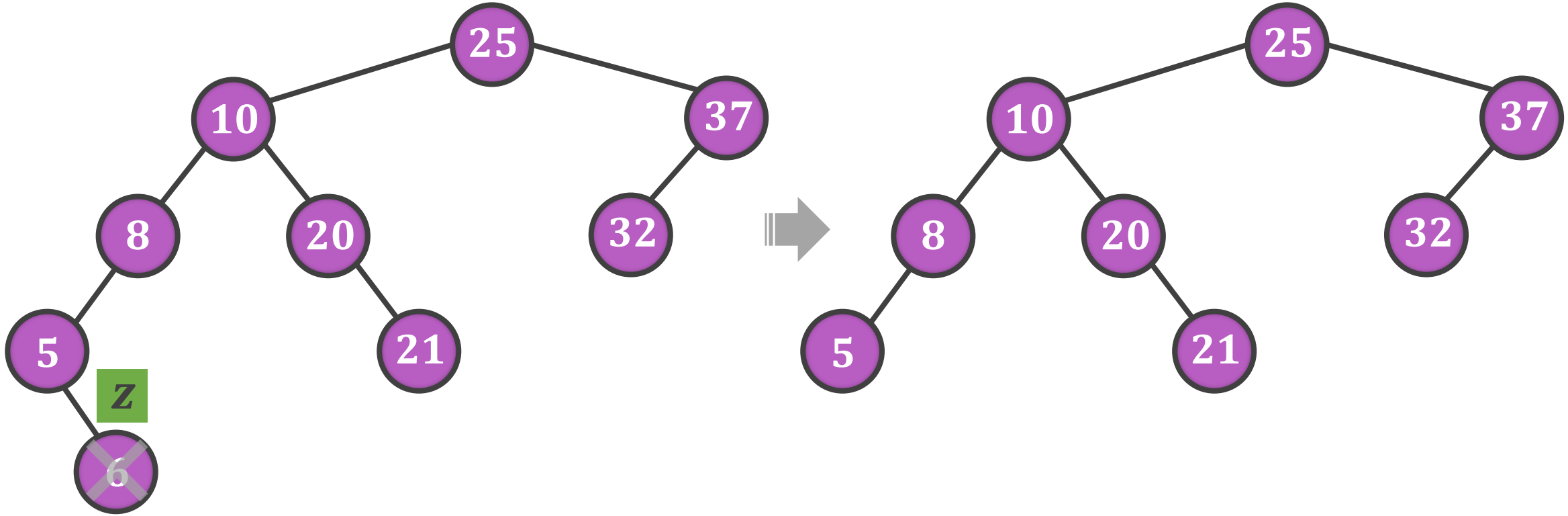


מחיקה בעץ חיפוש בינארי

ישנם שלושה מקרים:

1

ל-z אין בנים:
מעדכנים את אביו $z.parent$ כך שבנו יהיה $NULL$ ולא z

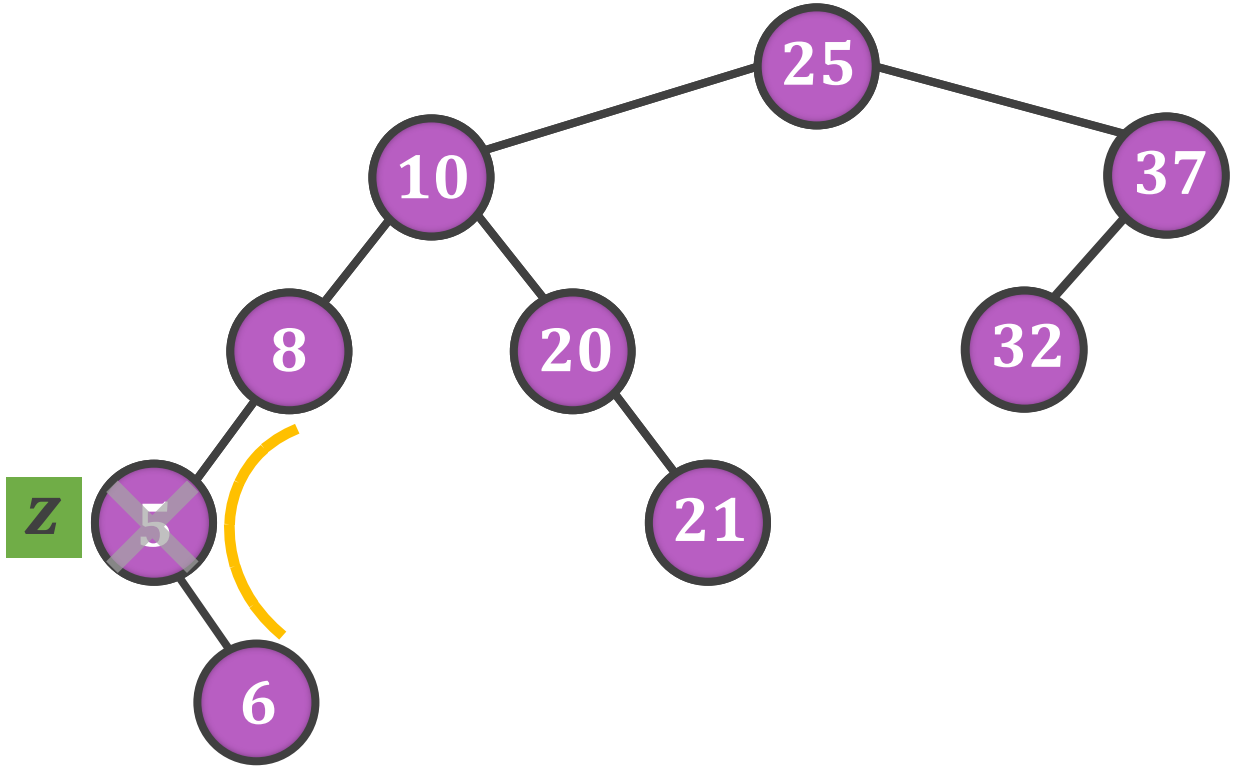


מחיקה בעץ חיפוש בינארי

ישנם שלושה מקרים:

2

ל-z יש רק בן אחד:
הבן של z הופך להיות בן של $z.parent$

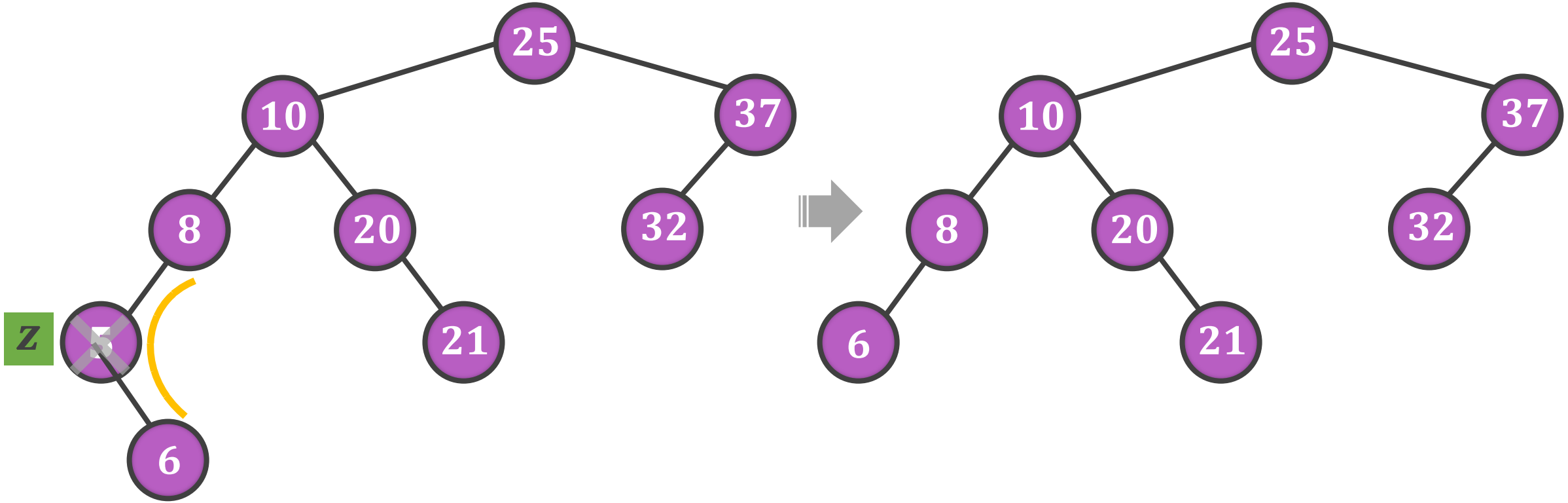


מחיקה בעץ חיפוש בינארי

ישנם שלושה מקרים:

2

ל-z יש רק בן אחד:
הבן של z הופך להיות בן של $z.parent$

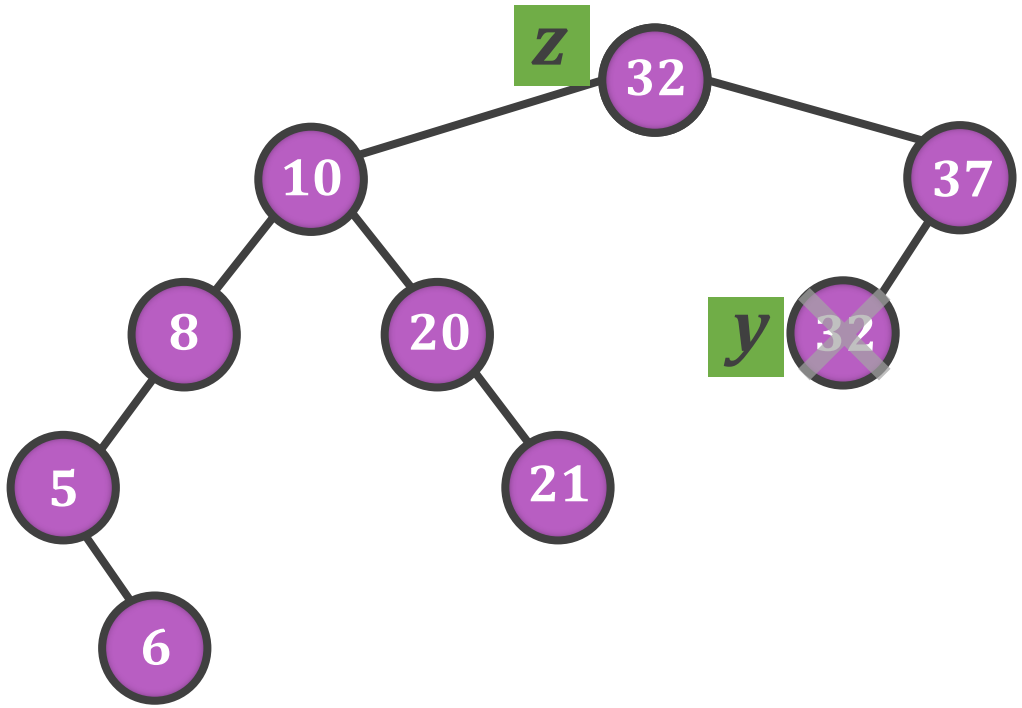


מחיקה בעץ חיפוש בינארי

ישנם שלושה מקרים:

3

ל-z יש שני בנים:
לעוקב של z , נסמן ב-y, אין בן שמאלי
נסיר את y (מקרה 1 או 2)
המפתח והנתונים נלווים של z מוחלפים באלו של y

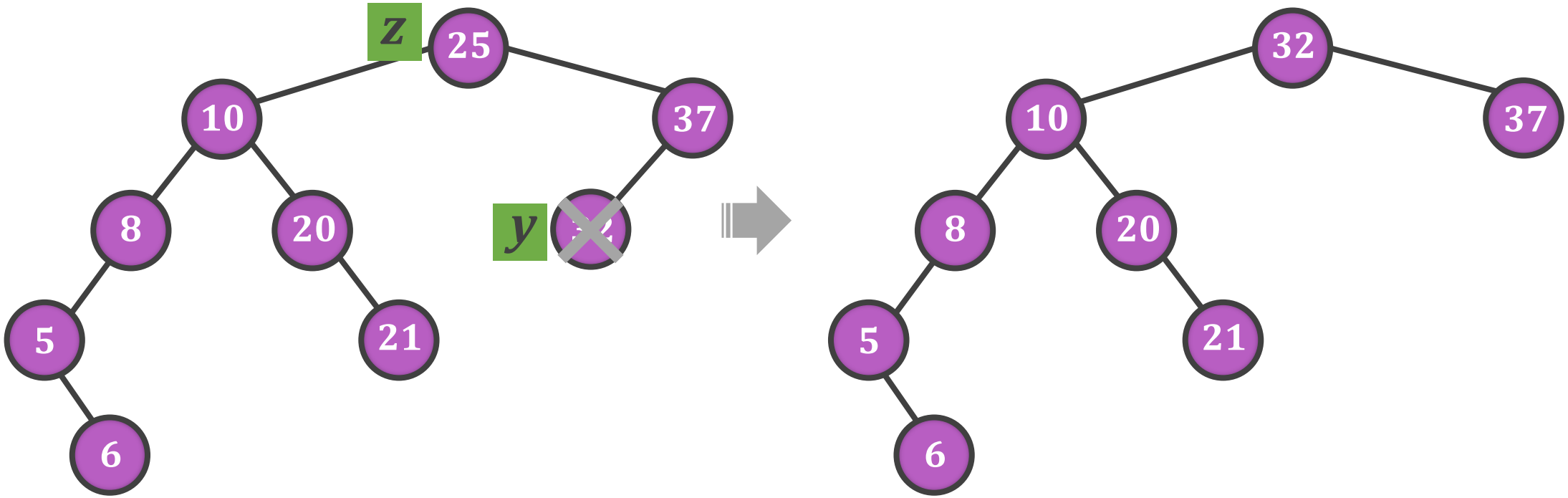


מחיקה בעץ חיפוש בינארי

ישנם שלושה מקרים:

3

ל-z יש שני בנים:
לעוקב של z , נסמנו ב-y, אין בן שמאלי
נסיר את y (מקרה 1 או 2)
המפתח והנתונים נלווים של z מוחלפים באלו של y



מחיקה בעץ חיפוש בינארי

ישנם שלושה מקרים

1. ל-z אין בנים:
משנים את אביו $z.parent$ כך שבנו יהיה $NULL$ ולא z

2. ל-z יש רק בן אחד:
הבן של z הופך להיות בן של $z.parent$

3. ל-z יש שני בנים:
לעוקב של z , נסמן ב-y, אין בן שמאלי
נסיר את y (מקרה 1 או 2)
המפתח והנתונים נלווים של z מוחלפים באלו של y

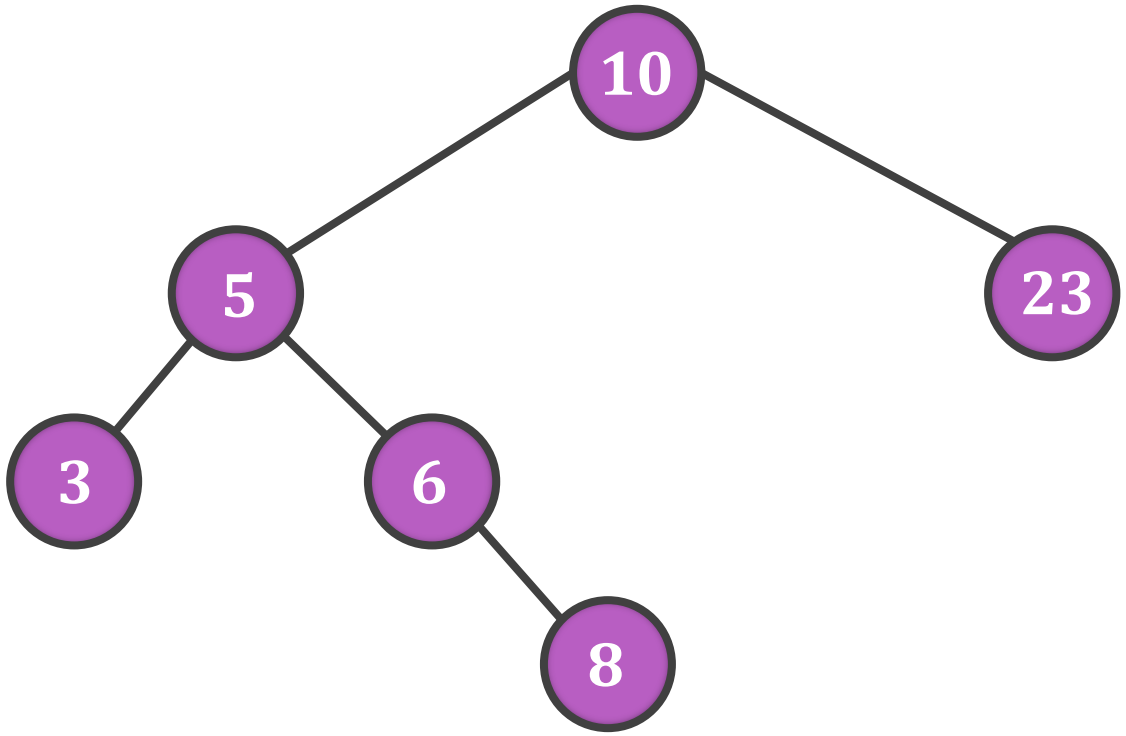
השלימו את המשפט:

זמן ריצה במקרה הגרוע של פעולת $Delete$ בעץ חיפוש בינארי בעל n צמתים וגובה h הוא ___.

1. $\Theta(n)$
2. $\Theta(1)$
3. $\Theta(h)$
4. $\Theta(\log n)$

נתון עץ חיפש בינארי מייד אחרי שבוצעה עליו פעולה אחת –
הכנסה או הוצאה. מה יכולה להיות הפעולה שבוצעה על העץ?

סמנו את כל התשובות האפשריות



1. $Insert(5)$
2. $Insert(8)$
3. $Insert(10)$
4. $Delete(20)$
5. $Delete(9)$

סיכום

מה למדנו?

1

$Search(k)$

$Insert(x)$

$Delete(x)$

$Min()$

$Max()$

$Successor(x)$

$Predecessor(x)$

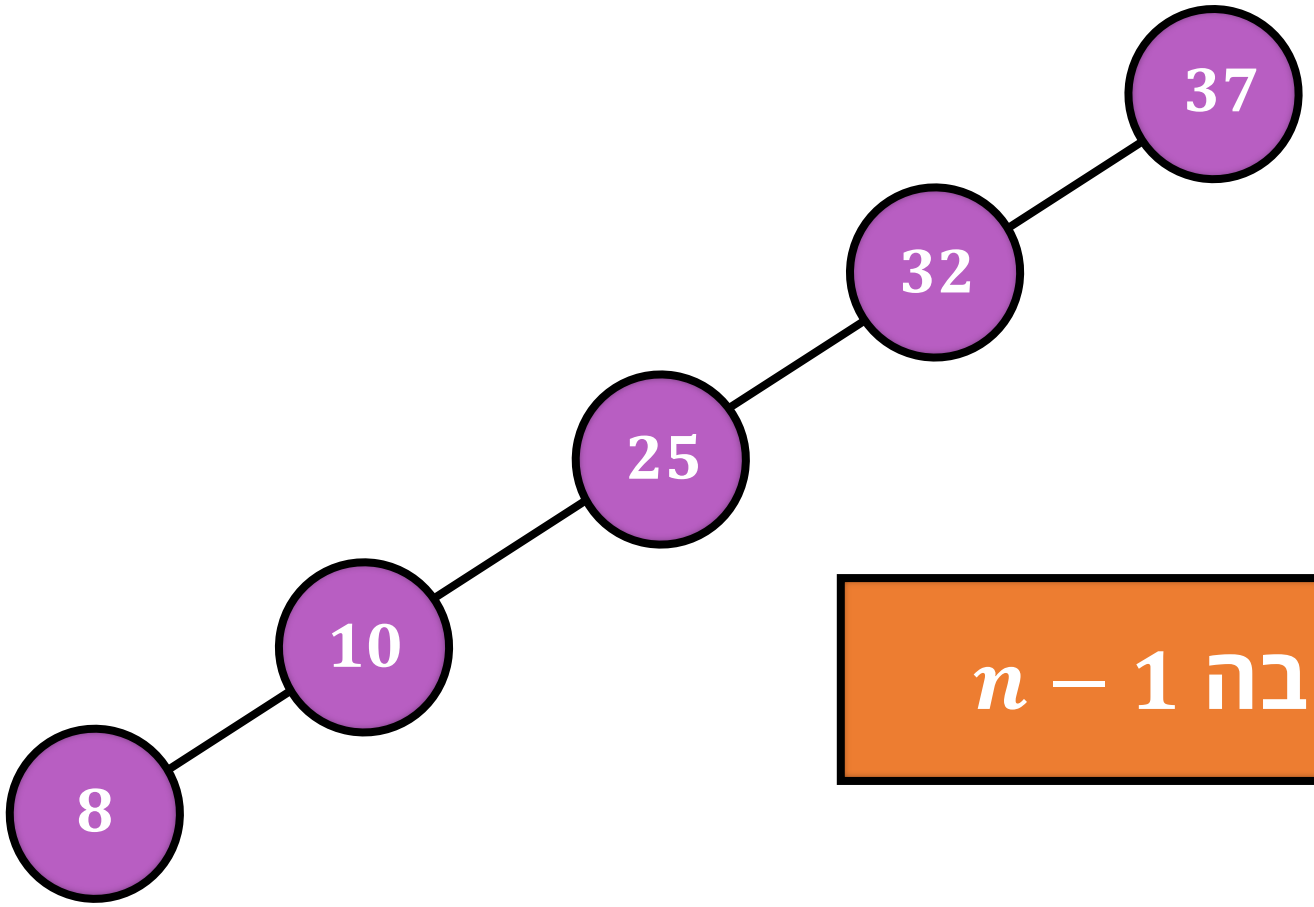
מה למדנו?

2

זמן ריצה של הפעולות על עץ חיפוש בינארי הוא $O(h)$

מה למדנו?

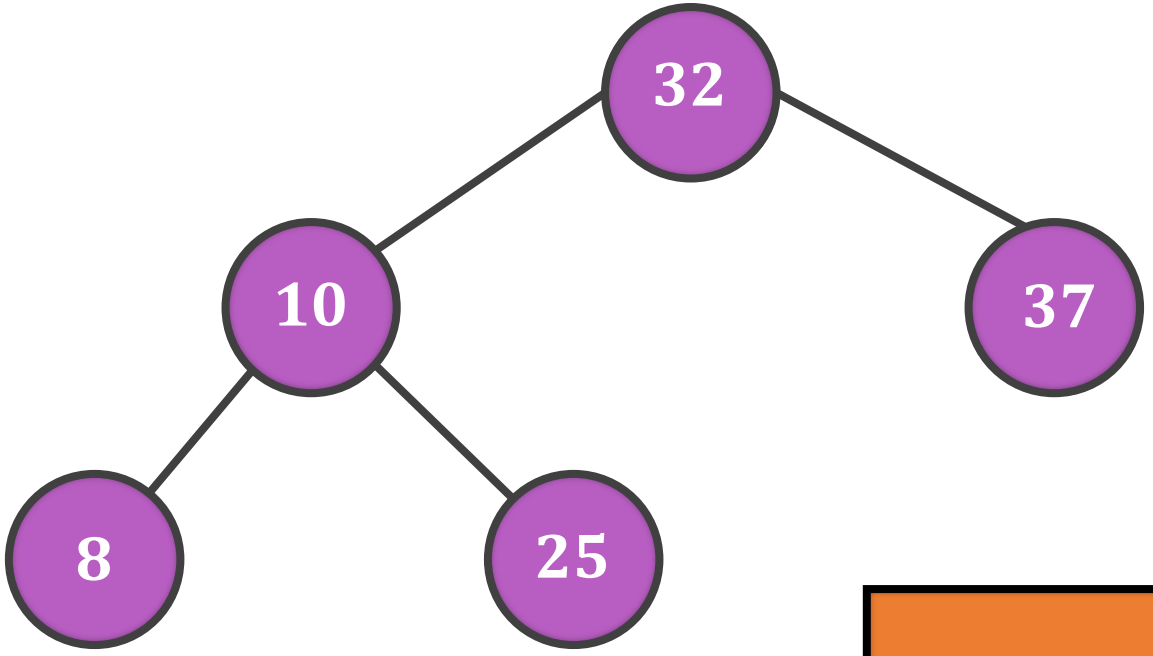
3



גובה $n-1$

מה למדנו?

4



גובה $\lfloor \log n \rfloor$