



מבני נתונים ואלגוריתמים



# ערימות פיבונאצ'י

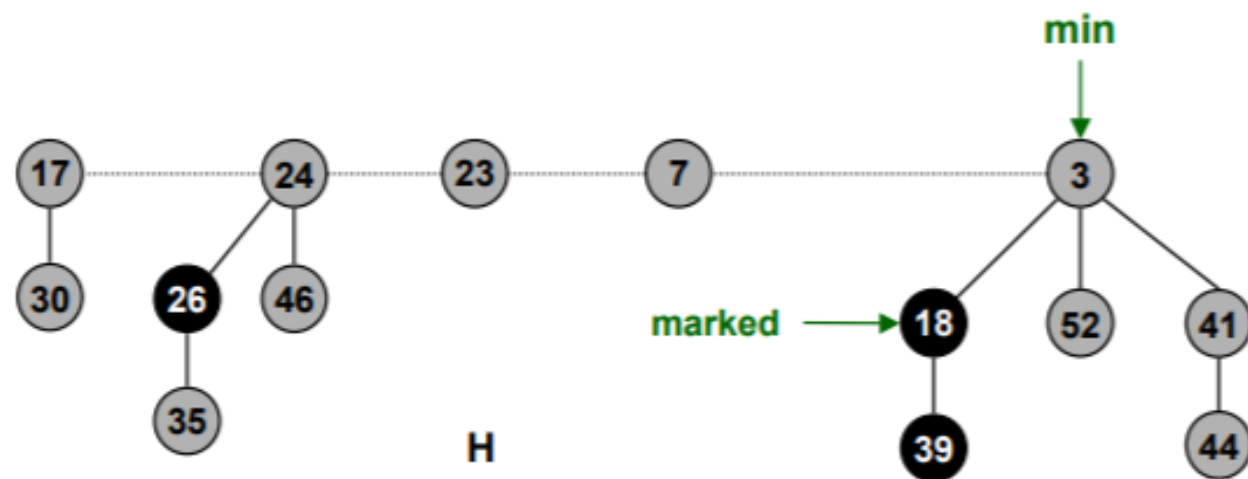
מבנה נתונים שתומך בפעולות union-find

כל הפעולות  $O(1)$  מלבד למעט מחיקת מינימום  $O(\log n)$

# מבנה

- אוסף של עצים שמקיים את תכונת הערימה
- כל שורשי העץ מחבורים ברשימה דו מקושרת מעגלית.  
fast union •
- כל הבנים של קודקוד מסוים מחוברים ברשימה דו מקושרת מעגלית.  
• פיצול מהיר של תתי עצים
- יש פוינטר לערך המינימלי
- ניתן לדעת בקלות מה הערך המינימלי

# מבנה





# הוספת ערך

יוצרים עץ חדש ששורשו  $x$

מוסיפים משמאל לערך  $\min$

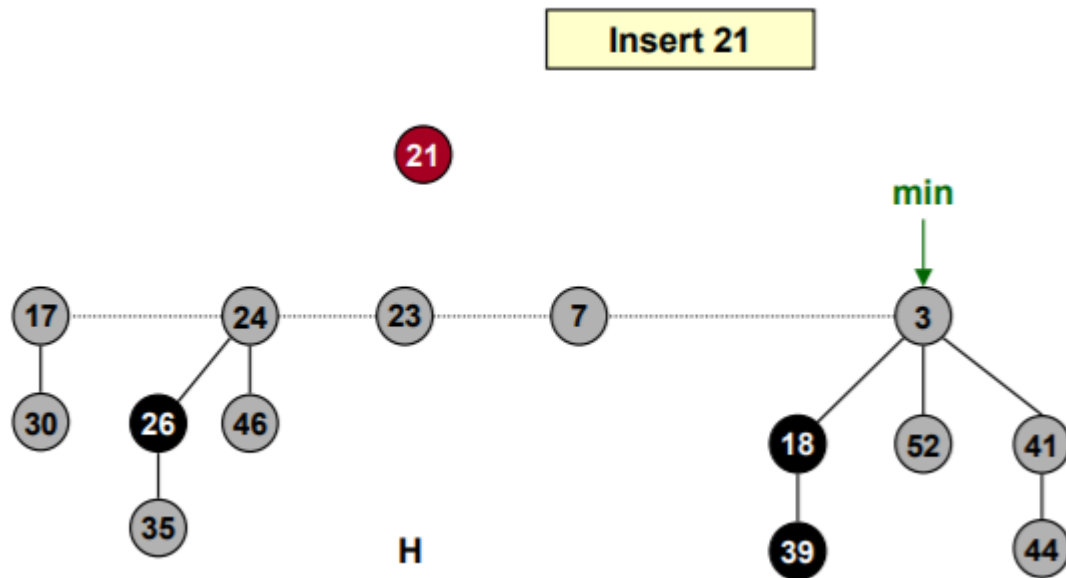
אם יש צורך - מעדכנים מינימום

# הוספת ערך

יוצרים עץ חדש ששורשו x

מוסיפים משמאל לערך min

אם יש צורך - מעדכנים מינימום

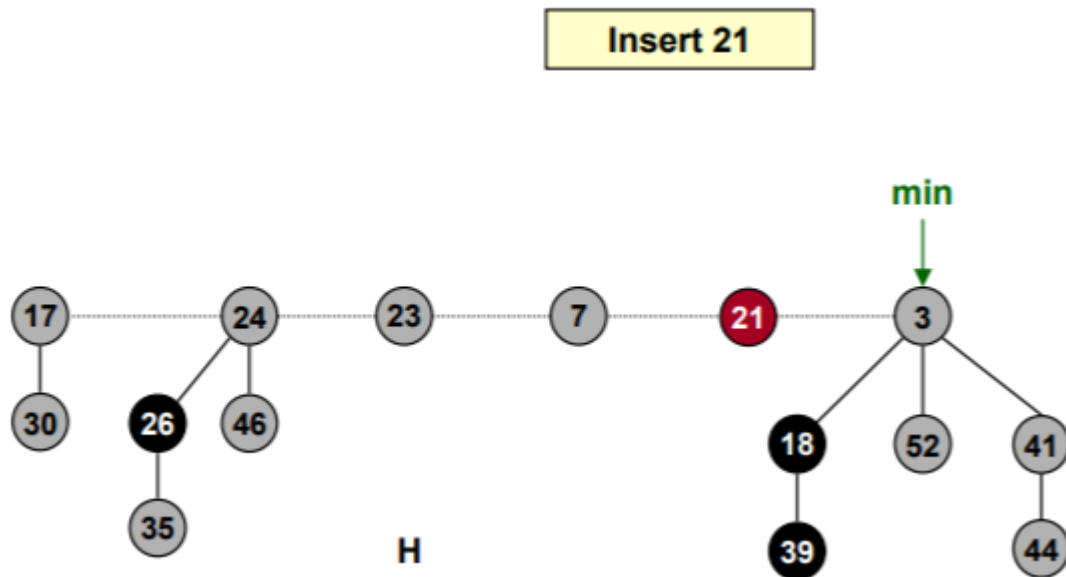


# הוספת ערך

יוצרים עץ חדש ששורשו x

מוסיפים משמאל לערך min

אם יש צורך - מעדכנים מינימום

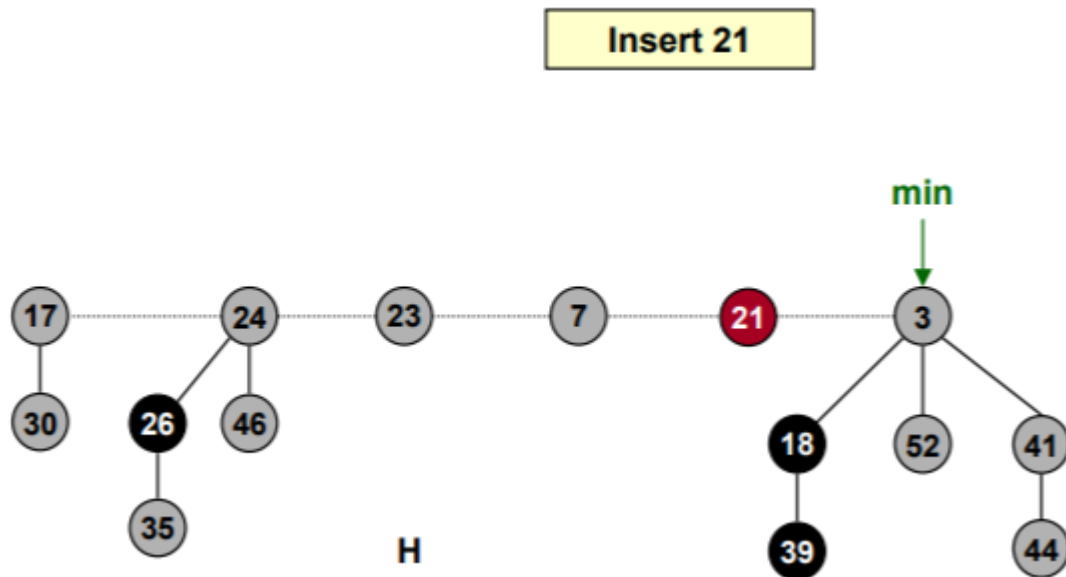


# הוספת ערך

יוצרים עץ חדש ששורשו x

מוסיפים משמאל לערך min

אם יש צורך - מעדכנים מינימום





# איחוד שתי ערימות

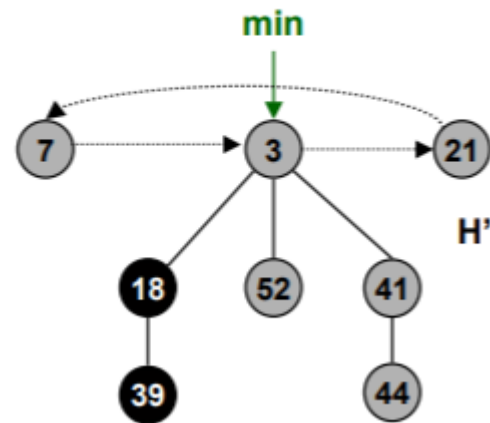
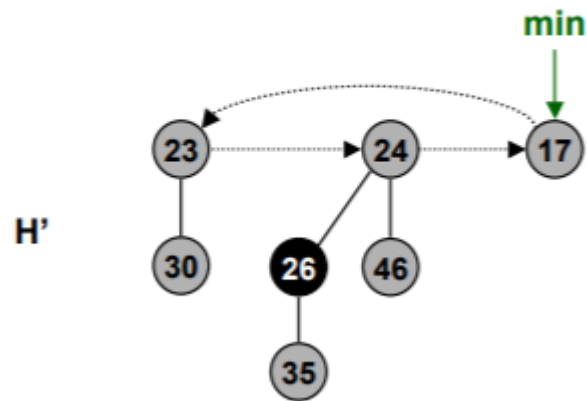
איחוד רשימת השורשים

עידכון min

# איחוד שתי ערימות

איחוד רשימת השורשים

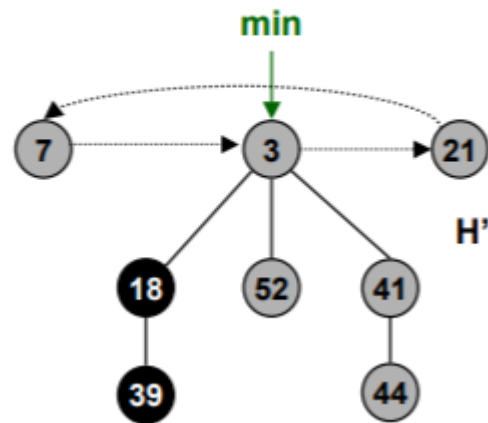
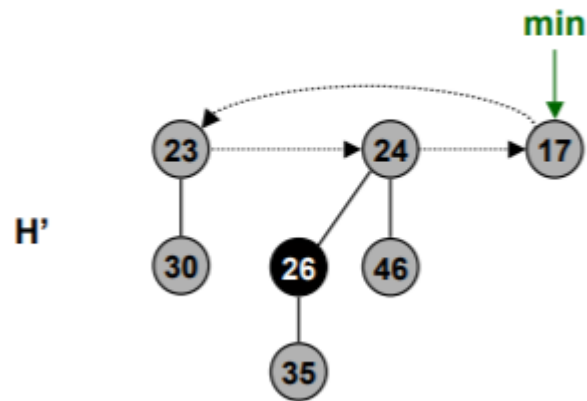
עידכון min



# איחוד שתי ערימות

איחוד רשימת השורשים

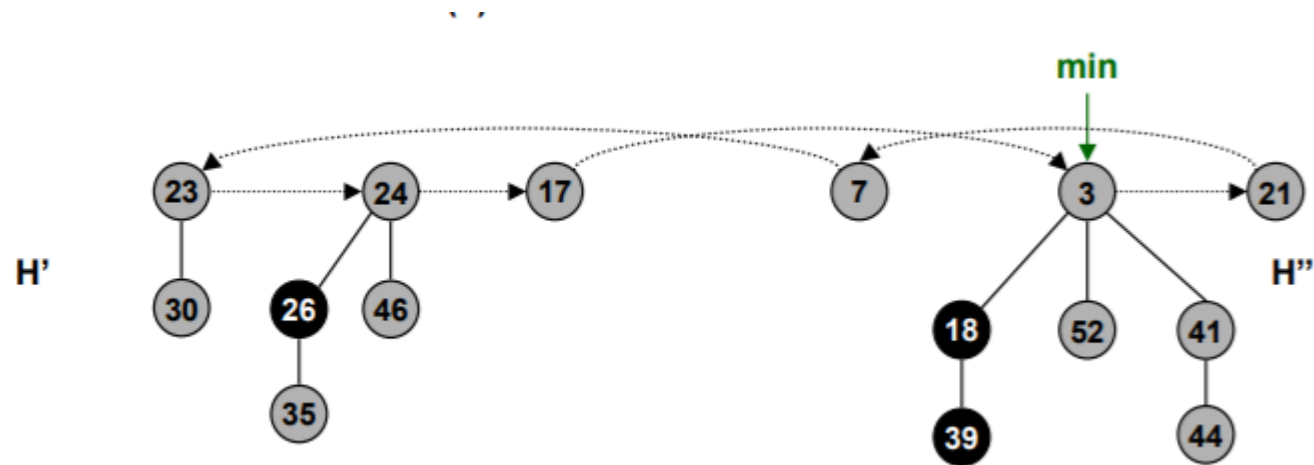
עידכון min



# איחוד שתי ערימות

איחוד רשימת השורשים

עידכון min



# מחיקת מינימום

מחיקת  $\min$  והפיכת בניו לעצים עצמאיים.

נאחד עצים בעלי דרגה זהה – אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה)

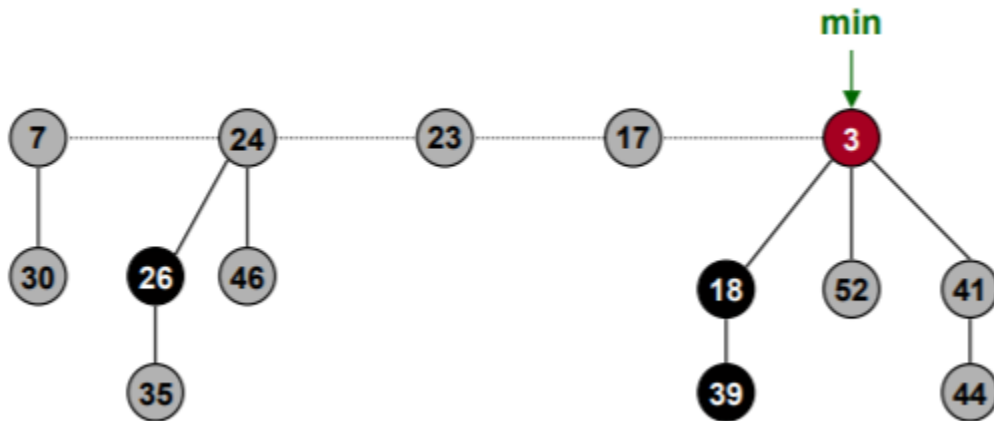
- עוברים על כל השורשים (נעדכן  $\min$  בסוף)
- יוצרים מערך מצביעים  $A$  כך ש  $A[i]$  מצביע על עץ שדרגת שורשו  $i$ .
- אם נתקלים בשני עצים בעלי דרגה זהה – נאחד כך שהנמוך יהיה השורש.

# מחיקת מינימום

מחיקת min והפיכת בניו לעצים עצמאיים.

נאחד עצים בעלי דרגה זהה- אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה)

- עוברים על כל השורשים (נעדכן min בסוף)
- יוצרים מערך מצביעים A כך ש  $A[i]$  מצביע על עץ שדרגת שורשו i.
- אם נתקלים בשני עצים בעלי דרגה זהה- נאחד כך שהנמוך יהיה השורש.

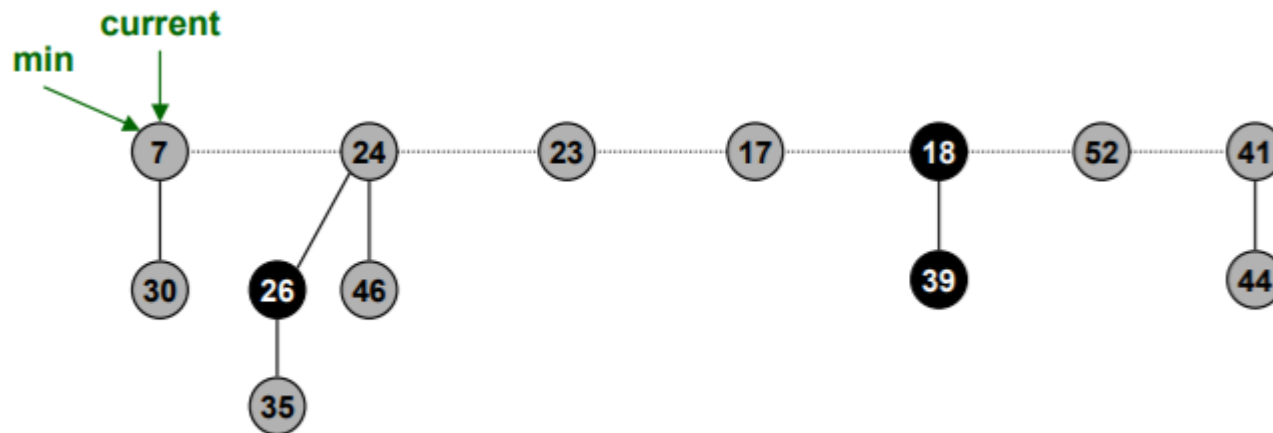


# מחיקת מינימום

מחיקת min והפיכת בניו לעצים עצמאיים.

נאחד עצים בעלי דרגה זהה- אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה)

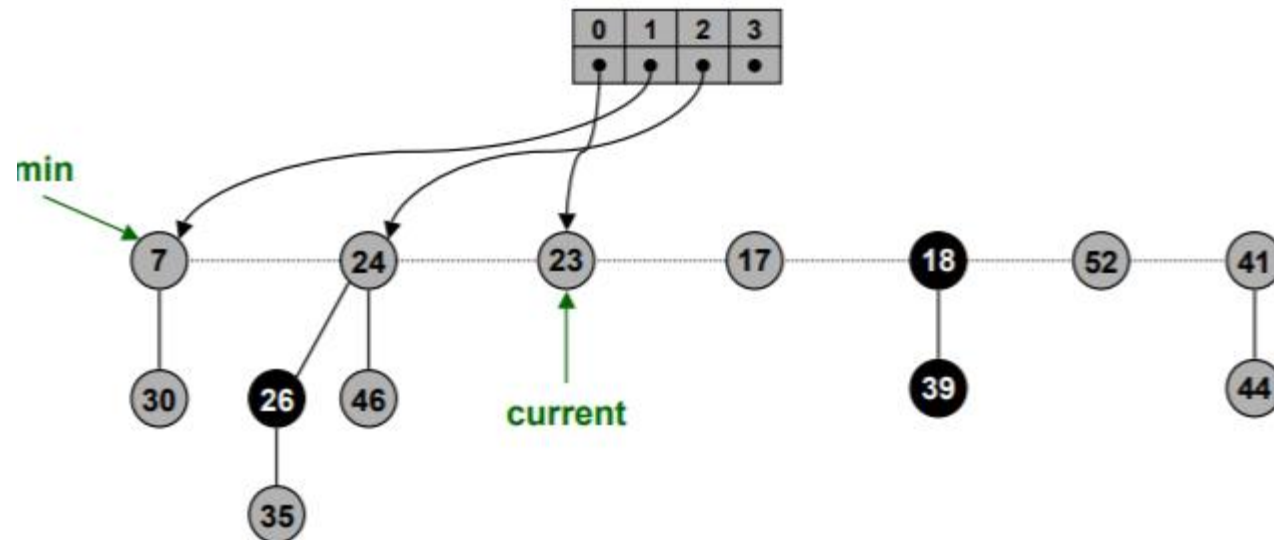
- עוברים על כל השורשים (נעדכן min בסוף)
- יוצרים מערך מצביעים A כך ש  $A[i]$  מצביע על עץ שדרגת שורשו i.
- אם נתקלים בשני עצים בעלי דרגה זהה- נאחד כך שהנמוך יהיה השורש.



# מחיקת מינימום

מחיקת min והפיכת בניו לעצים עצמאיים.

- נאחד עצים בעלי דרגה זהה- אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה) עוברים על כל השורשים (נעדכן min בסוף)
- יוצרים מערך מצביעים A כך ש  $A[i]$  מצביע על עץ שדרגת שורשו i.
- אם נתקלים בשני עצים בעלי דרגה זהה- נאחד כך שהנמוך יהיה השורש.

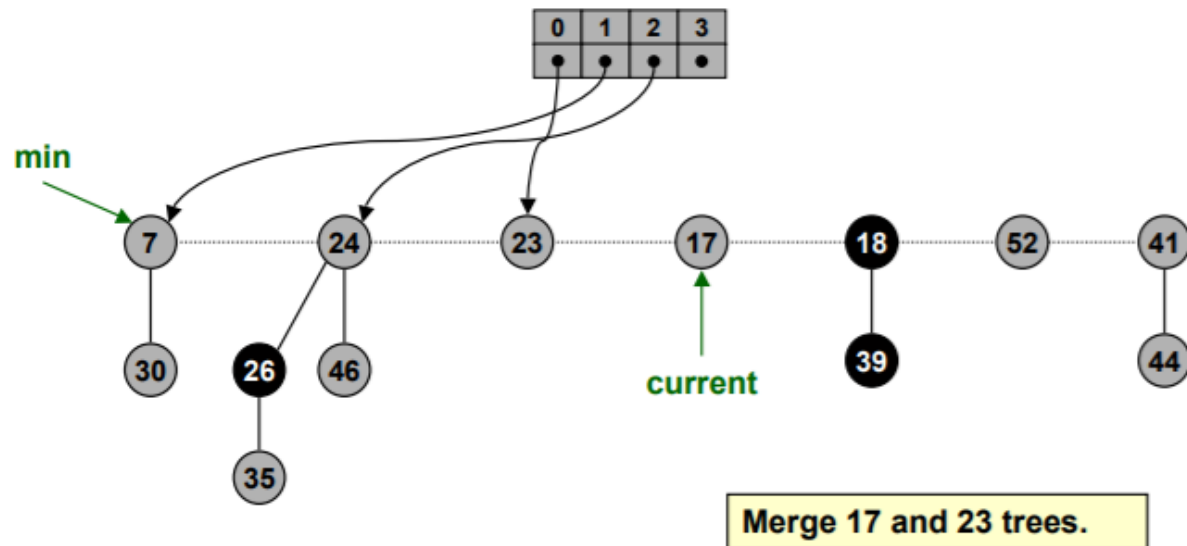




# מחיקת מינימום

מחיקת min והפיכת בניו לעצים עצמאיים.

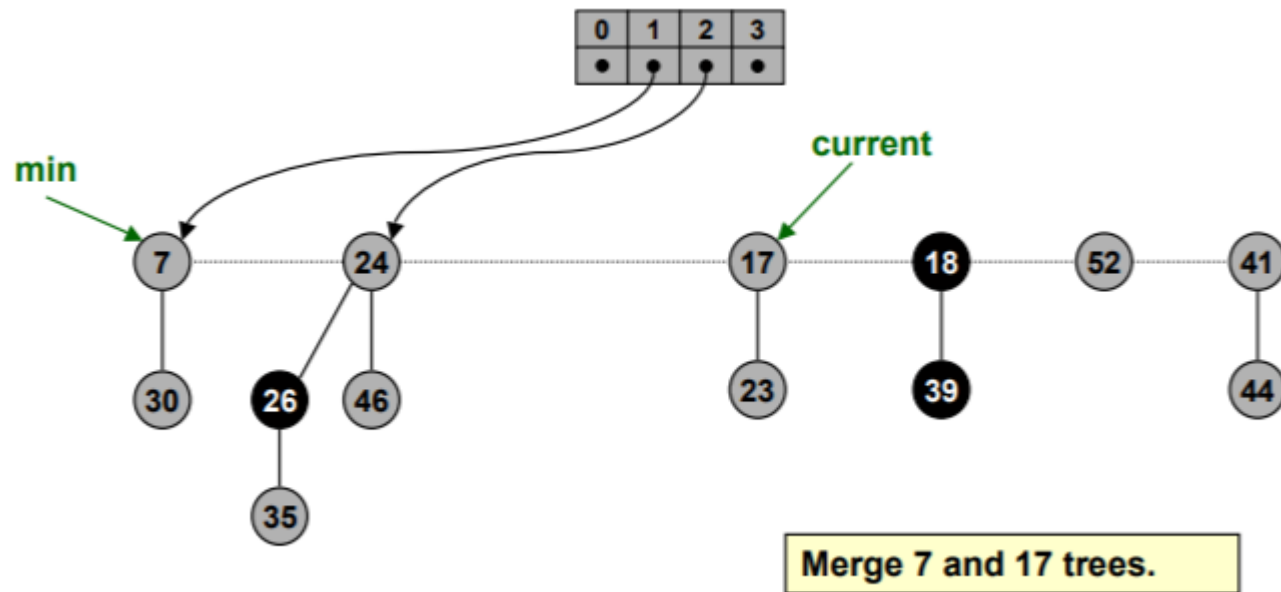
- נאחד עצים בעלי דרגה זהה- אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה (עוברים על כל השורשים (נעדכן min בסוף))
- יוצרים מערך מצביעים A כך ש  $A[i]$  מצביע על עץ שדרגת שורשו i.
- אם נתקלים בשני עצים בעלי דרגה זהה- נאחד כך שהנמוך יהיה השורש.



# מחיקת מינימום

מחיקת min והפיכת בניו לעצים עצמאיים.

- נאחד עצים בעלי דרגה זהה- אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה (עוברים על כל השורשים (נעדכן min בסוף))
- יוצרים מערך מצביעים A כך ש  $A[i]$  מצביע על עץ שדרגת שורשו i.
- אם נתקלים בשני עצים בעלי דרגה זהה- נאחד כך שהנמוך יהיה השורש.

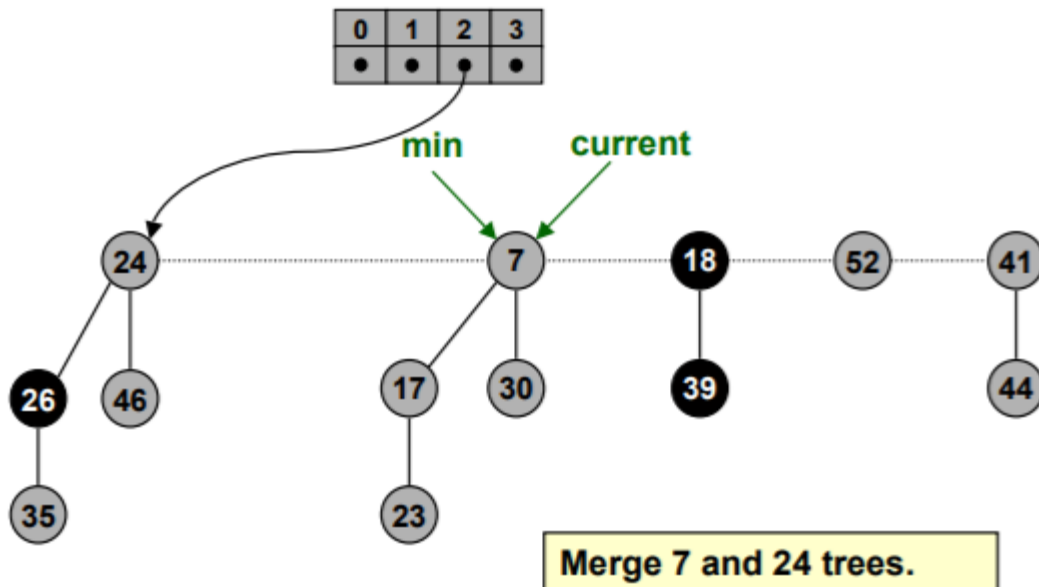


# מחיקת מינימום

מחיקת min והפיכת בניו לעצים עצמאיים.

נאחד עצים בעלי דרגה זהה- אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה)

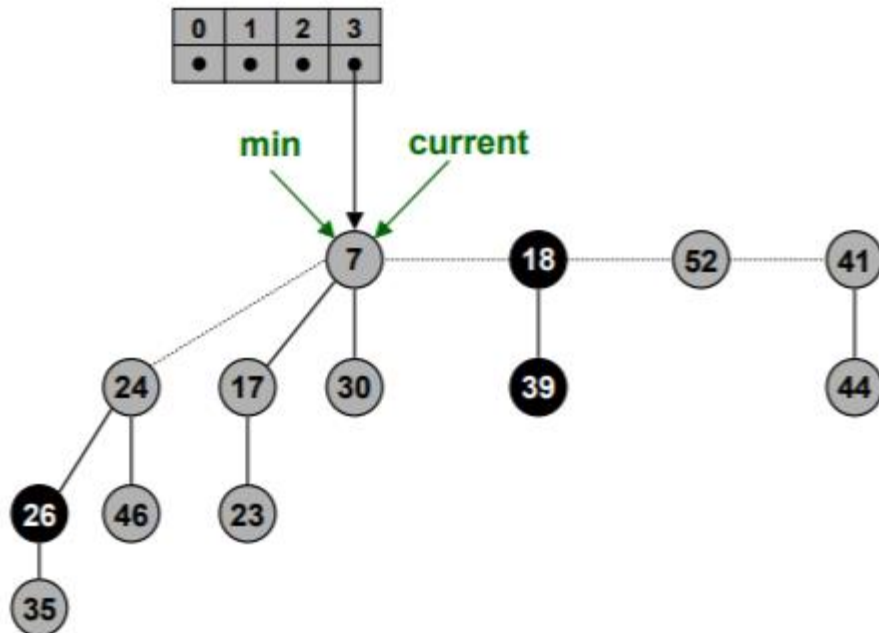
- עוברים על כל השורשים (נעדכן min בסוף)
- יוצרים מערך מצביעים A כך ש  $A[i]$  מצביע על עץ שדרגת שורשו i.
- אם נתקלים בשני עצים בעלי דרגה זהה- נאחד רד ושהמוד יהיה הנוורו



# מחיקת מינימום

מחיקת min והפיכת בניו לעצים עצמאיים.

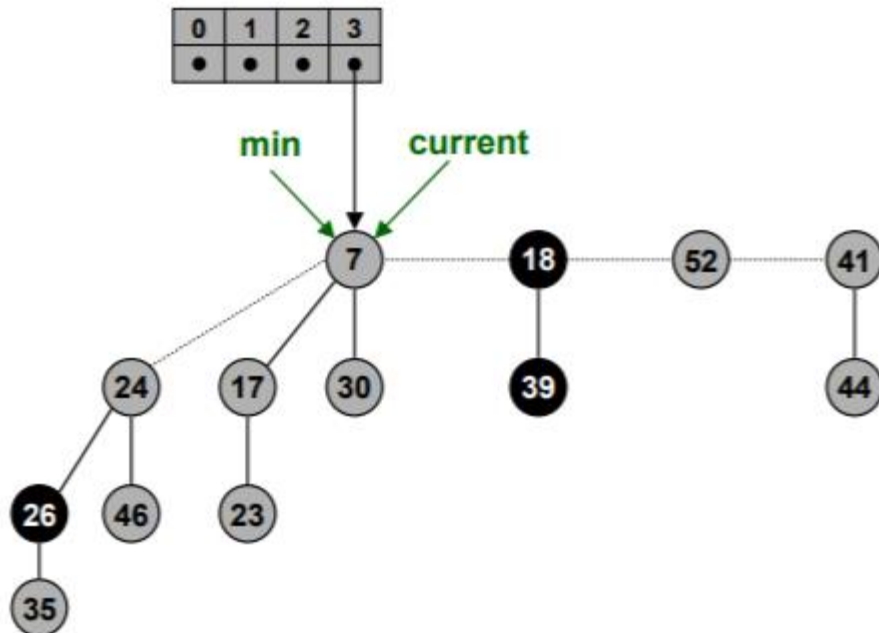
- נאחד עצים בעלי דרגה זהה- אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה) עוברים על כל השורשים (נעדכן min בסוף)
- יוצרים מערך מצביעים A כך ש  $A[i]$  מצביע על עץ שדרגת שורשו  $i$ .
- אם נתקלים בשני עצים בעלי דרגה זהה- נאחד כך שהנמוך יה



# מחיקת מינימום

מחיקת min והפיכת בניו לעצים עצמאיים.

- נאחד עצים בעלי דרגה זהה- אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה) עוברים על כל השורשים (נעדכן min בסוף)
- יוצרים מערך מצביעים A כך ש  $A[i]$  מצביע על עץ שדרגת שורשו i.
- אם נתקלים בשני עצים בעלי דרגה זהה- נאחד כך שהנמוך יה

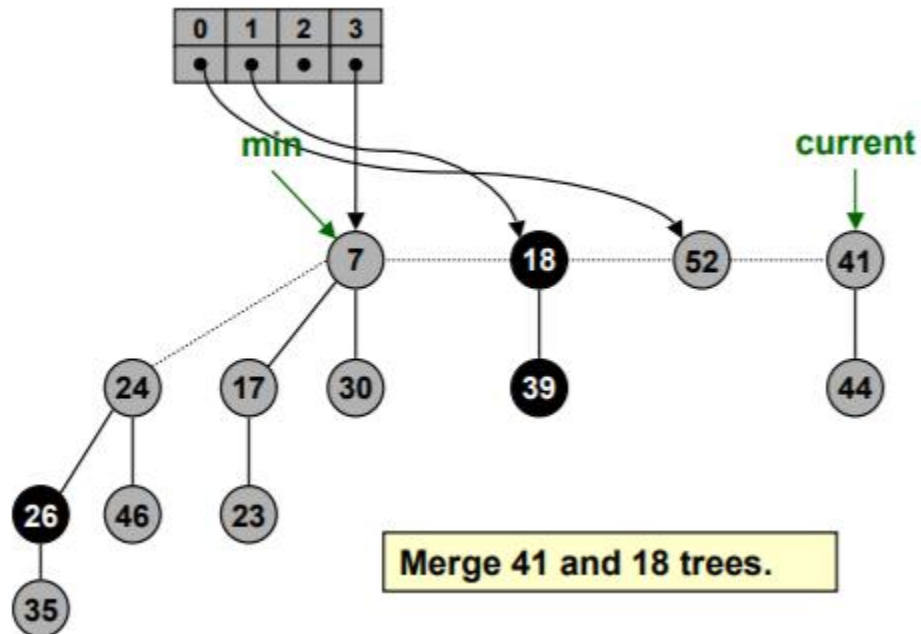


# מחיקת מינימום

מחיקת min והפיכת בניו לעצים עצמאיים.

נאחד עצים בעלי דרגה זהה- אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה)

- עוברים על כל השורשים (נעדכן min בסוף)
- יוצרים מערך מצביעים A כך ש  $A[i]$  מצביע על עץ שדרגתו  $i$
- אם נתקלים בשני עצים בעלי דרגה זהה- נאחד כך שהנמוך

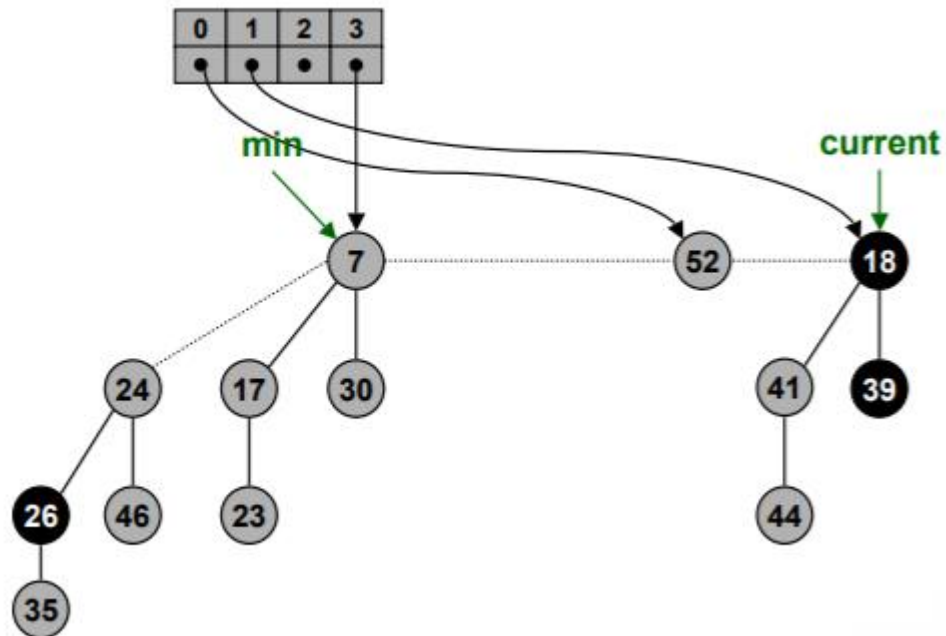


18 52 (לא צירפתי תמונה במיוחד עבור העצים עמכם הסליחה.)

# מחיקת מינימום

מחיקת min והפיכת בניו לעצים עצמאיים.

- נאחד עצים בעלי דרגה זהה- אסור שיהיו שני עצים ששורשיהם בעלי דרגה זהה)
  - עוברים על כל השורשים (נעדכן min בסוף)
  - יוצרים מערך מצביעים A כך ש  $A[n]$  מצביע על עץ שדרגה וזוויתו ;
  - אם נתקלים בשני עצים בעלי דרגה זהה- נאחד כך שהנמ



כרגע העץ עומד בכל הקרטיונים.

# DECREASE KEY(X)

ישנם שני מקרים:

א. הסדר לא מופר: אם  $x$  שורש או שכלל הערימה עדיין נשמר.

ב. הסדר מופר וצריך לתקן.

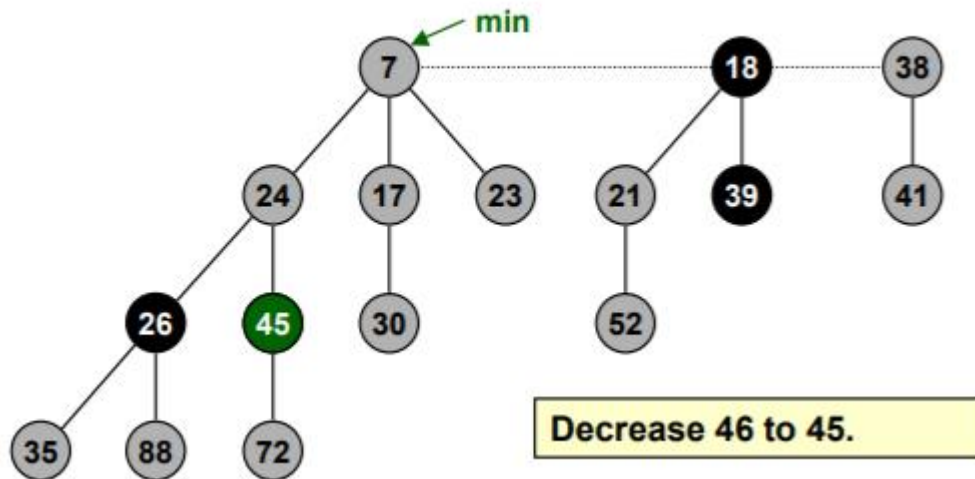


# DECREASE KEY(X)

הסדר לא מופר: אם  $x$  שורש או שכלל הערימה עדיין נשמר.

א. שנה את הערך בהתאם.

ב. עדכן מינימום אם יש צורך.



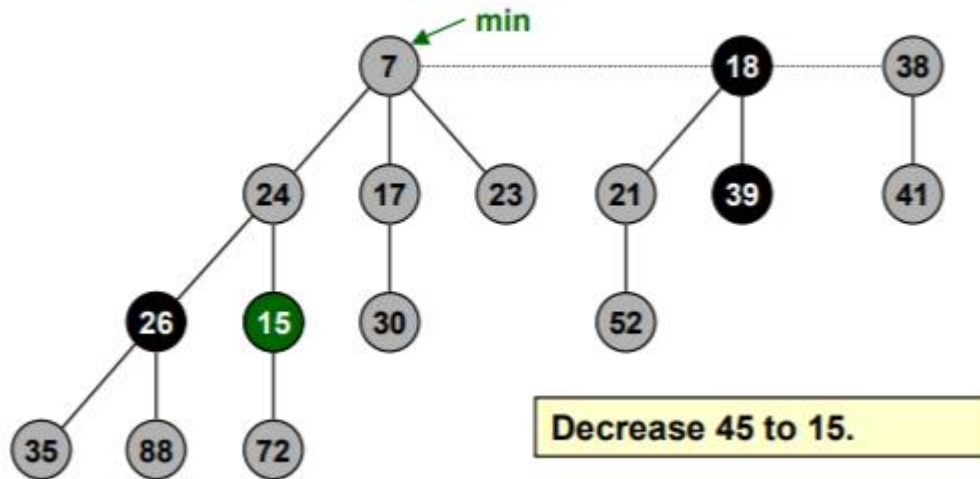
# DECREASE KEY(X)

הסדר מופר וצריך לתקן:

מקרה ראשון:  $p(x)$  לא מסומן. (הערה:  $p(x)$  הוא ההורה של  $x$ )

1. שנה את ערכו של  $x$ .
2. נתקן אותו מאביו.
3. הפוך את  $x$  לשורש.
4. סמן את  $p(x)$
5. עדכן מינימום אם ישנו צורך.

# DECREASE KEY(X)



1. שנה את ערכו של  $x$ .

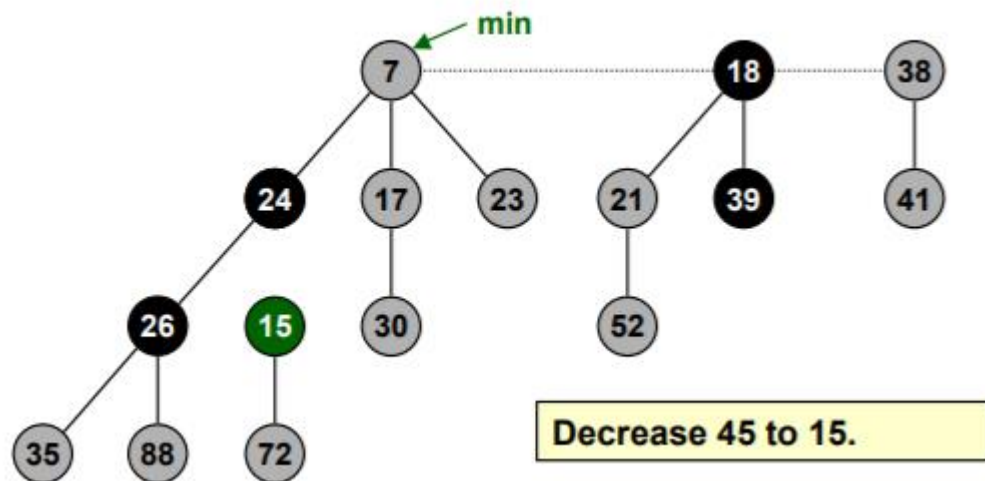
2. נתקן אותו מאביו.

3. הפוך את  $x$  לשורש.

4. סמן את  $p(x)$

5. עדכן מינימום אם ישנו צורך.

# DECREASE KEY(X)



1. שנה את ערכו של  $x$ .

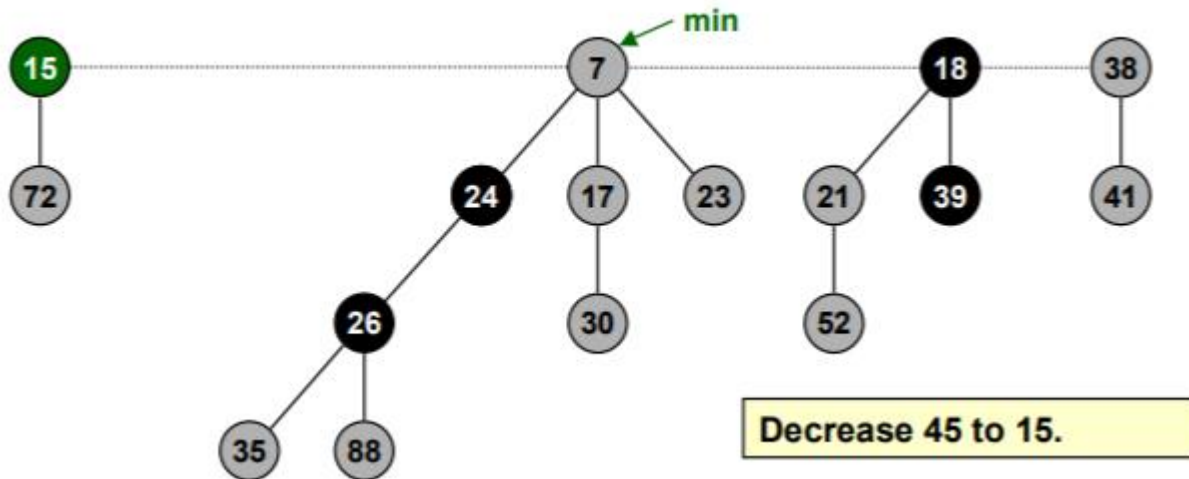
2. נתקן אותו מאביו.

3. הפוך את  $x$  לשורש.

4. סמן את  $p(x)$

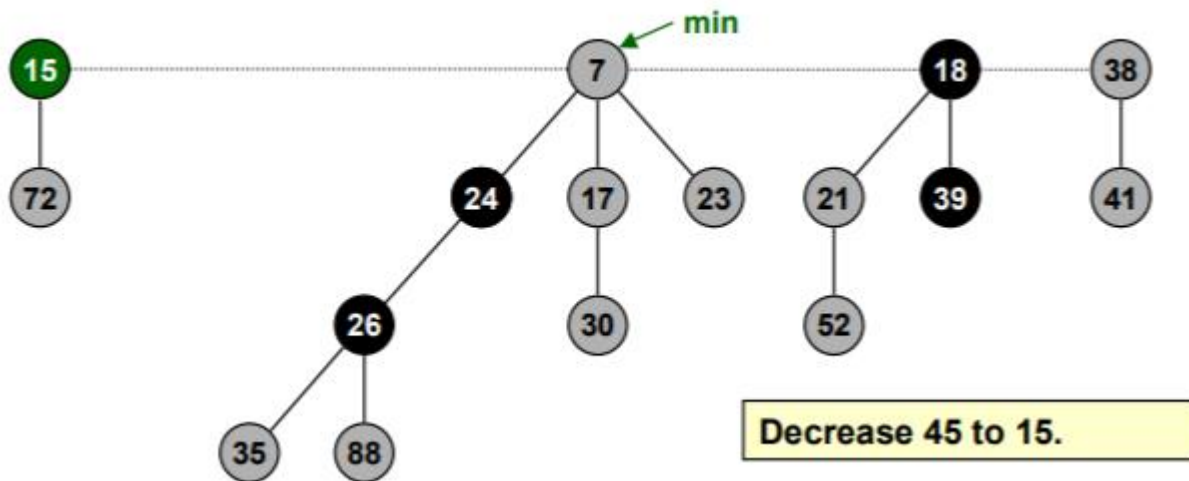
5. עדכן מינימום אם ישנו צורך.

# DECREASE KEY(X)



1. שנה את ערכו של  $x$ .
2. נתקן אותו מאביו.
3. הפוך את  $x$  לשורש.
4. סמן את  $p(x)$ .
5. עדכן מינימום אם ישנו צורך.

# DECREASE KEY(X)



1. שנה את ערכו של  $x$ .
2. נתקן אותו מאביו.
3. הפוך את  $x$  לשורש.
4. סמן את  $p(x)$ .
5. עדכן מינימום אם ישנו צורך.

# DECREASE KEY(X)

הסדר מופר וצריך לתקן:

מקרה ראשון:  $p(x)$  (הערה:  $p(x)$  הוא ההורה של  $x$ )

1. שנה את ערכו של  $x$ .

2. נתק אותו מאביו.

3. הפוך את  $x$  לשורש.

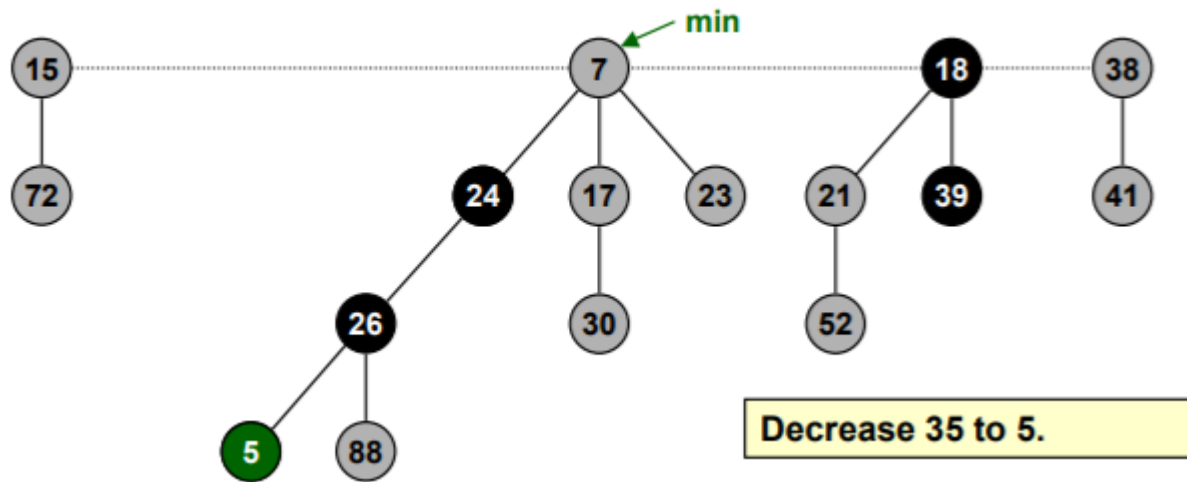
4. נתק את  $p(p(x))$

4.5 הסר סימן.

5. אם  $p(p(x))$  לא מסומן – סמן. אם לא – חזור על התהליך רקורסיבית.

6. עדכן  $\min$

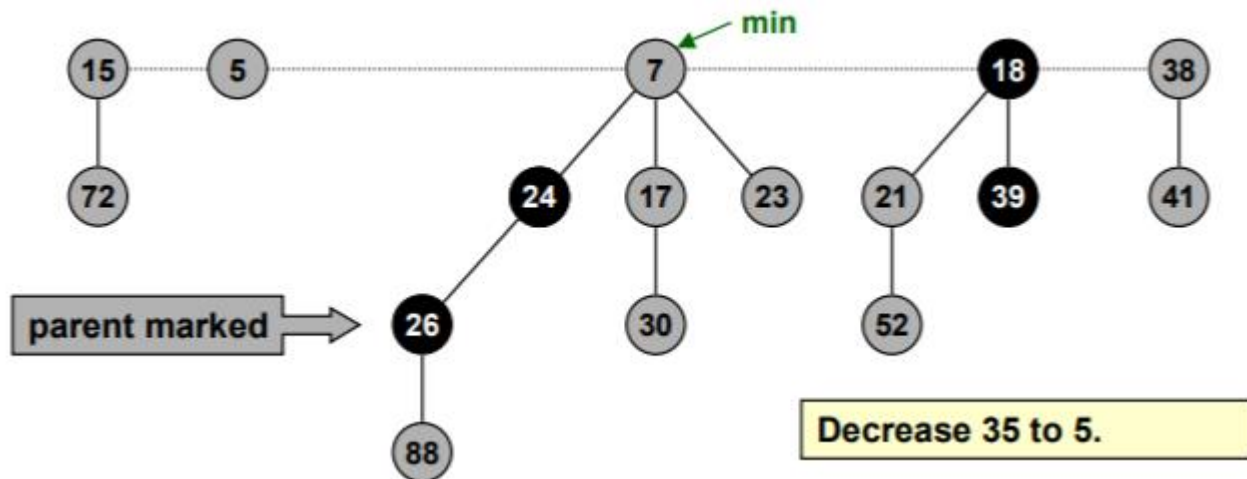
# DECREASE KEY(X)



1. שנה את ערכו של  $x$ .
2. נתקן אותו מאביו.
3. הפוך את  $x$  לשורש.
4. נתקן את  $p(p(x))$ .
- 4.5 הסר סימן.
5. אם  $p(p(x))$  לא מסומן - סמן. אם לא -
6. עדכן  $min$ .



# DECREASE KEY(X)



1. שנה את ערכו של  $x$ .

2. נתקן אותו מאביו.

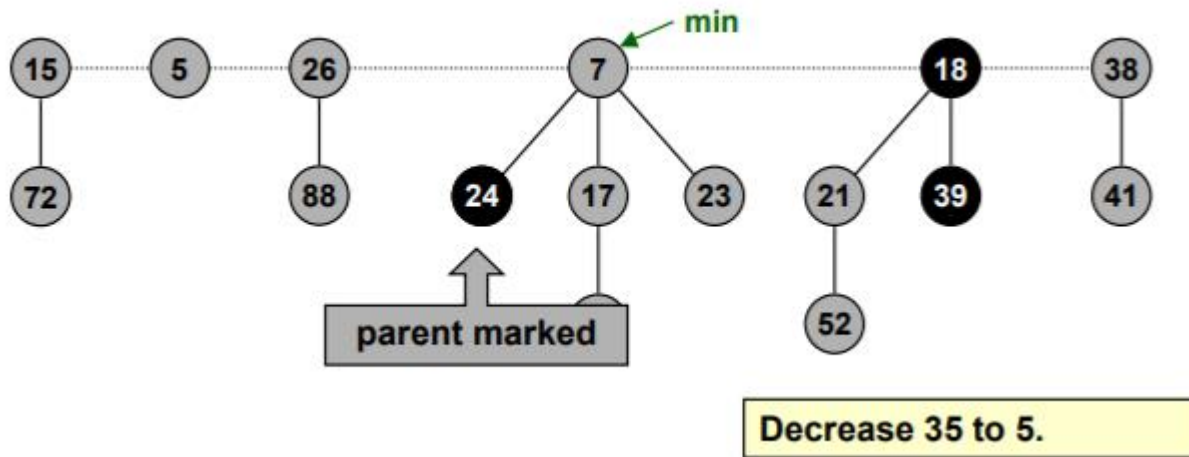
3. הפוך את  $x$  לשורש.

4. נתקן את  $p(p(x))$

4.5 הסר סימן.

5. אם  $p(p(x))$  לא מסומן - סמן. אם לא - חז

6. עדכן  $min$



1. שנה את ערכו של  $x$ .

2. נתקן אותו מאביו.

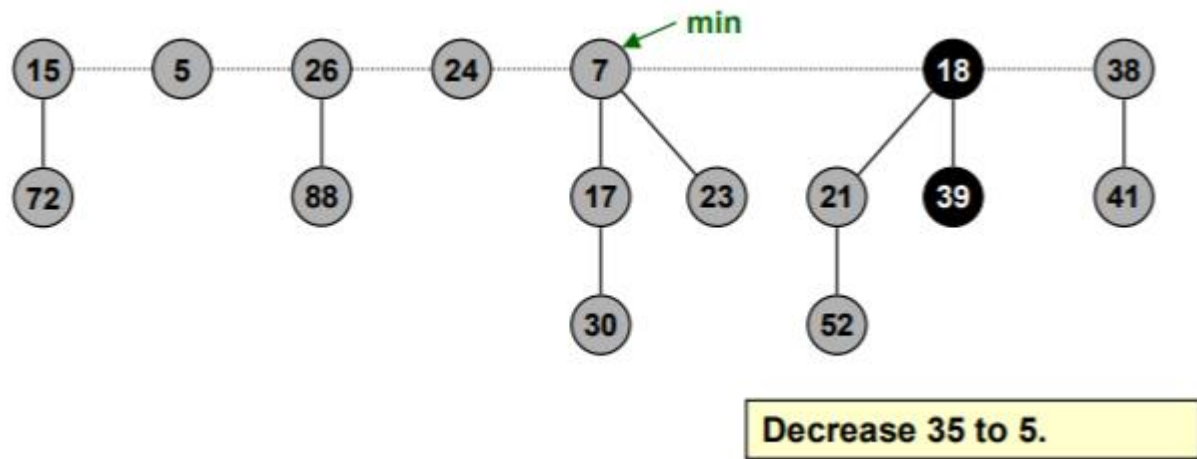
3. הפוך את  $x$  לשורש.

4. נתקן את  $p(p(x))$

4.5 הסר סימן.

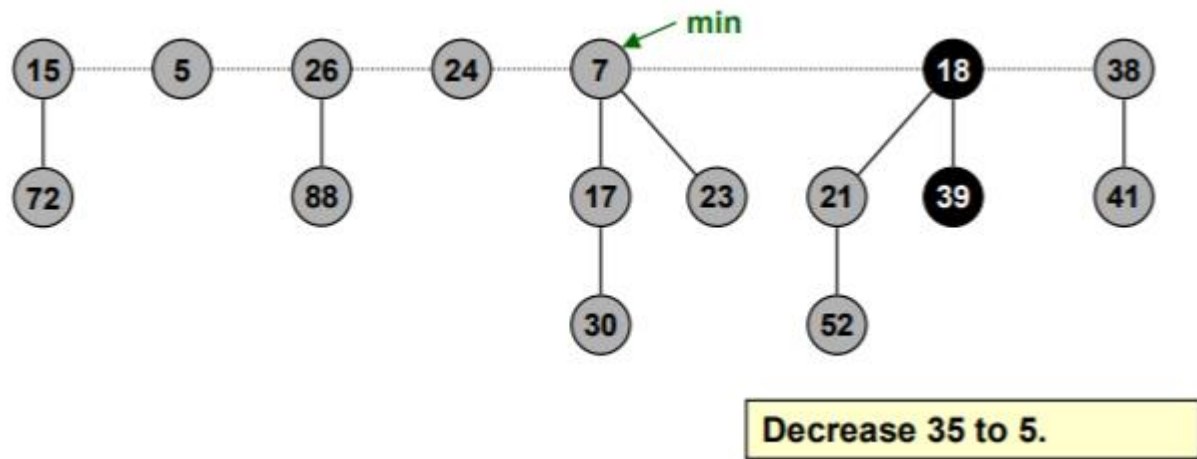
5. אם  $p(p(x))$  לא מסומן - סמן. אם לא - חזור על התהליך רקורסיבית.

6. עדכן  $min$



36

1. שנה את ערכו של  $x$ .
2. נתקן אותו מאביו.
3. הפוך את  $x$  לשורש.
4. נתקן את  $p(p(x))$ .
- 4.5 הסר סימן.
5. אם  $p(p(x))$  לא מסומן - סמן. אם לא - חזור על התהליך רקורסיבית.
6. עדכן  $\min$ .



36

1. שנה את ערכו של  $x$ .
2. נתקן אותו מאביו.
3. הפוך את  $x$  לשורש.
4. נתקן את  $p(p(x))$ .
- 4.5 הסר סימן.
5. אם  $p(p(x))$  לא מסומן - סמן. אם לא - חזור על התהליך רקורסיבית.
6. עדכן  $min$



# מחזיקה

שנה את הערך ל  $-\infty$   
מחק את המינימום.