

# דפים מדורגים לתלמידי "עיצוב תוכנה"

## הסרת איבר מעץ חיפוש בינארי

### גרסת ג'אוה

צוות הפיתוח:

שרה וגנר

קרן קוטלובסקי

אהובה שפרלינג

רקע:

הגישה מבוססת על סדרת דפים, שכל אחד מהם נותן לתלמיד רמז או עזרה "צנועה" המסייעים לו באופן מדורג בפתרון המשימה.

המורה מחלק לתלמידים דף (מספר 1) ועליו תיאור של משימה מורכבת שעליהם לבצע. על שולחן המורה נמצאים דפים נוספים: דף מספר 2, דף מספר 3, וכן הלאה. בכל דף כזה ניתנים לתלמיד רמז קל או עזרה "צנועה" שיכולים לקדם אותו מעט בביצוע המשימה. (לא פתרון מלא של כל המשימה, אלא רק דחיפה בכיוון המתאים).

השיעור מתנהל כך שכל התלמידים מקבלים את הדף הראשון אבל כל תלמיד מחליט באופן עצמאי מתי (ואם בכלל) הוא ניגש לשולחן ולוקח את הדף הבא. מותר לתלמיד לקחת גם מספר דפים בו-זמנית.

## הסרת איבר מעץ חיפוש בינארי

### דף מספר 1

תזכורת: עץ חיפוש בינארי הוא עץ בינארי (כלומר שלכל צומת יש עד שני בנים: בן ימני ובן שמאלי), שבו כל צומת מקיים את התכונה שכל ערכי הצמתים בתת העץ הימני שלו גדולים או שווים מערך הצומת, וכל ערכי הצמתים בתת העץ השמאלי שלו קטנים מערך הצומת.

#### משימה 1:

לפניך כותרת פעולה המקבלת עץ חיפוש בינארי (tree) והפניה לצומת בעץ (node) ומוחקת את הצומת הנתון מהעץ תוך שמירת תכונות עץ החיפוש הבינארי. הפעולה תחזיר הפנייה לעץ לאחר מחיקת הצומת המבוקש.

```
public static BinTreeNode<Integer>
```

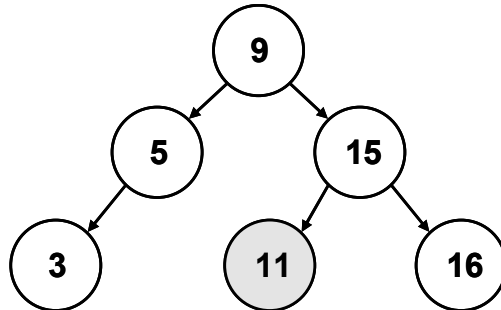
```
delete (BinTreeNode<Integer> tree, BinTreeNode<Integer> node)
```

צור מחלקה חדשה בשם DeleteMethod וממש את הפעולה delete.

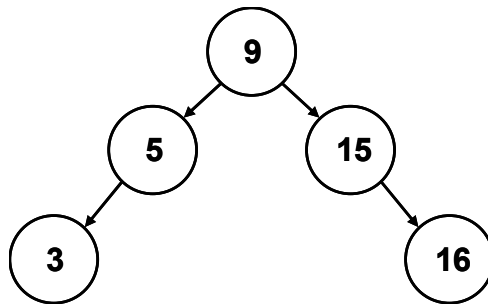
## דפים מדורגים – הסרת איבר מעץ חיפוש בינארי

### דף מספר 2

לפניך עץ חיפוש בינארי:



לאחר הסרת הצומת בעל הערך 11 נקבל את עץ החיפוש הבינארי הבא:



### משימה 2:

כתוב פעולה בשם delete המקבלת עץ חיפוש בינארי (tree) והפניה לצומת כלשהו בעץ שהוא עלה

(node) ומסירה אותו מהעץ תוך שמירה על תכונות עץ החיפוש הבינארי.

הפעולה תחזיר הפנייה לעץ לאחר מחיקת הצומת. במידה והעץ היה עץ עלה - הפעולה תחזיר null.

**הערה:** השתמש בפעולות העזר בדף המצורף (ראה בסוף הקובץ).

דפים מדורגים – הסרת איבר מעץ חיפוש בינארי.

### דף מספר 3

לפניך פיתרון המשימה המופיעה בדף 2:

פעולת המחיקה:

```
public static void delete (BinTreeNode<Integer> tree, BinTreeNode<Integer> node )
{
    BinTreeNode<Integer> parent = parent(tree,node);
    if (parent!=null)
    {
        if(parent.getLeft()==node)
            parent.setLeft(null);
        else
            parent.setRight(null);
    }
    else
    {
        tree=null;
    }
    return tree;
}
```

### משימה 3:

ענה על השאלות הבאות:

1. כיצד יודעים אם צומת הוא עלה?
2. הוסף במקום המתאים בפעולה delete בדיקה האם הצומת הוא אכן עלה.

דפים מדורגים – הסרת איבר מעץ חיפוש בינארי.

## דף מספר 4

לפניך פיתרון המשימה המופיעה בדף 3:

```
public static void delete (BinTreeNode<Integer> tree, BinTreeNode<Integer> node )
{
    BinTreeNode<Integer> parent = parent(tree,node);
    if(isLeaf(node))
    {
        if (parent!=null)
        {
            if(parent.getLeft()==node)
                parent.setLeft(null);
            else
                parent.setRight(null);
        }
        else
        {
            tree=null;
        }
        return tree;
    }
}
```

### משימה 4:

מקרה נוסף הקיים בעץ הוא המקרה שבו לצומת שרוצים להסיר יש בן יחיד.

להלן העץ לאחר הסרת הצומת בעל הבן היחיד:	לפניך עץ חיפוש בינארי נוסף. ברצוננו להסיר את הצומת בעל הערך 12

הוסף לפעולה delete במקום המתאים הסרה של צומת בעל בן יחיד.

דפים מדורגים – הסרת איבר מעץ חיפוש בינארי.

## דף מספר 5

### משימה 5:

הוסף לפרויקט SearchBinTree את המחלקה DeleteMethod בה נמצאת הפעולה delete שכתבת במשימה 4. שים לב כי במחלקה זו אמורות להופיע כל פעולות העזר שהפעולה delete משתמשת בהן.

בפרויקט זה תמצא מחלקה בשם SearchBinTree. מחלקה זו כוללת פעולה ראשית שהוגדר בה עץ חיפוש בינארי וזהו לעץ המופיע בדף מספר 4.

פעולה זו מזמנת את הפעולה delete ומציגה את העץ לפני ואחרי הסרת הצומת בעל הערך 12.

הרץ את הפרויקט וודא כי העץ שקיבלת לאחר זימון הפעולה אכן תואם לציפיותיך.

אם כן עבור לדף מספר 7.

אחרת עבור לדף מספר 6.

**דפים מדורגים – הסרת איבר מעץ חיפוש בינארי**

## **דף מספר 6**

### **משימה 6:**

#### **ענה על השאלות הבאות:**

בקוד שכתבת במשימה 4, מה הצומת שהסרת? האם היה לו בן שמאלי או בן ימני?  
האם התייחסת בנפרד למקרה שיש לצומת שרוצים להסיר רק בן שמאלי או ורק בן ימני או שמא  
התייחסת למקרה אחד מבין השניים?

**כעת, נסה לתקן את הקוד והרץ שוב את המחלקה על פי ההנחיות בדף מספר 5.**

## דף מספר 7

### לפניך פיתרון המשימה המופיעה בדף 4:

```
public static BinTreeNode<Integer> delete (BinTreeNode<Integer> tree, BinTreeNode<Integer> node )
{
    BinTreeNode<Integer> parent = parent(tree, node);
    if (isLeaf(node))
    {
        if (parent != null)
        {
            if (parent.getLeft() == node)
                parent.setLeft(null);
            else
                parent.setRight(null);
        }
        else
            tree = null;
        return tree;
    }
    else
    {
        if (node.getRight() == null)
        {
            if (parent != null)
            {
                if (parent.getLeft() == node)
                    parent.setLeft(node.getLeft());
                else
                    parent.setRight(node.getLeft());
            }
            else
                tree = node.getLeft();
            return tree;
        }
        else
        {
            if (node.getLeft() == null)
            {
                if (parent != null)
                {
                    if (parent.getLeft() == node)
                        parent.setLeft(node.getRight());
                    else
                        parent.setRight(node.getRight());
                }
                else
                    tree = node.getRight();
                return tree;
            }
        }
    }
}
```

המשך בדף הבא...

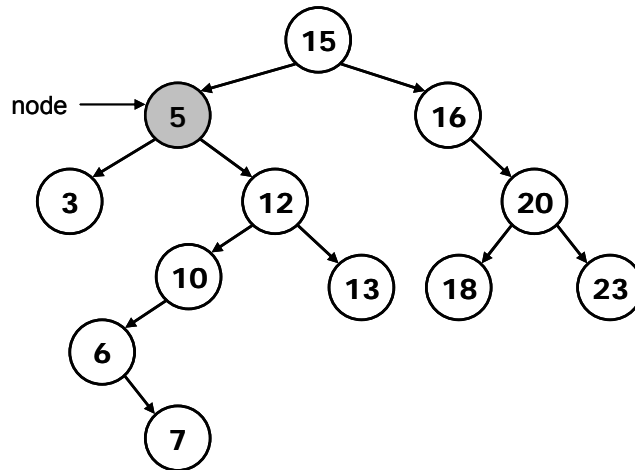


דפים מדורגים – הסרת איבר מעץ חיפוש בינארי

## דף מספר 7 – המשך

### משימה 7:

המקרה האחרון בו נטפל הוא המקרה שבו לצומת שברצוננו להסיר יש 2 בנים. ההסרה נעשית ע"י החלפת הערך בצומת שנרצה להסיר עם הערך העוקב לו בעץ. נתון העץ הבא (עץ חיפוש בינארי) וצומת node הוא הצומת שברצוננו להסיר. מהו הערך שצריך להיכנס לצומת node בעץ:



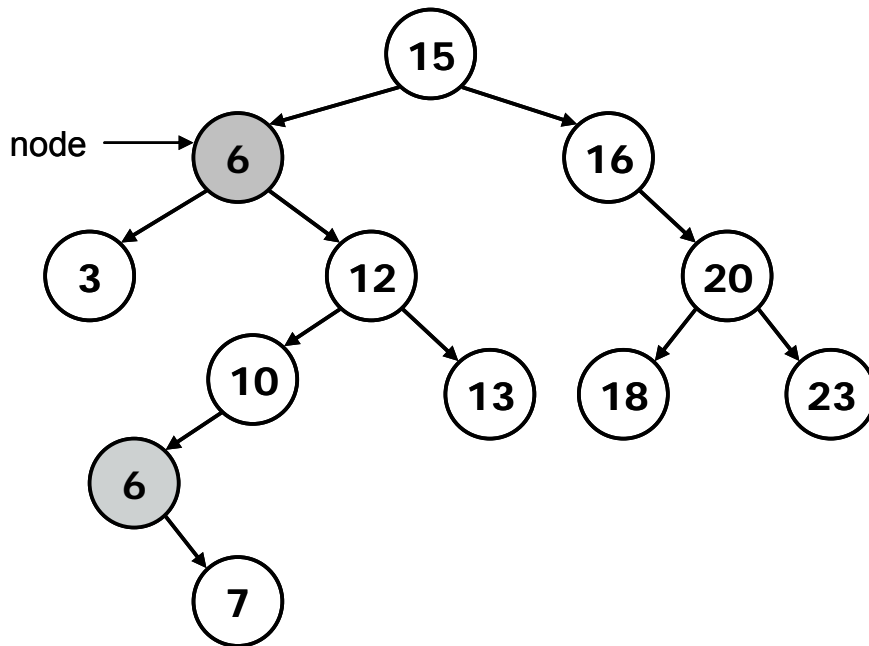
דפים מדורגים – הסרת איבר מעץ חיפוש בינארי

## דף מספר 8

### פיתרון משימה 7:

הערך שיחליף את ערך הצומת שברצוננו להסיר הוא 6.

להלן העץ לאחר שינוי ערך הצומת:



הערך שהושם בצומת הוא הערך הקטן ביותר בתת העץ הימני.

שים לב: העץ המצויר מסמל רק את שלב שינוי הערך של הצומת שאנו רוצים להסיר.

### משימה 8:

1. הוסף פעולת עזר המקבלת הפניה לצומת שרוצים להסיר (node) ומחזירה הפניה לצומת בתת

העץ הימני שערכו הוא הקטן ביותר.

2. הוסף לפעולה delete קוד המוצא את הצומת שערכו יחליף את הערך בצומת node ובצע את

ההחלפה. השתמש בפעולה שכתבת והוסף לקוד את הקטע שמשנה את שני הערכים. כלומר,

משנה את ערך הצומת שרוצים להסיר לערך של הצומת שהוחזר.

## דף מספר 9

לפניך פיתרון המשימה המופיעה בדף 8: פעולת delete מעודכנת:

```
public static BinTreeNode<Integer> delete (BinTreeNode<Integer> tree, BinTreeNode<Integer> node )
{
    BinTreeNode<Integer> parent = parent(tree,node);
    if(isLeaf(node))
    {
        if (parent!=null)
        {
            if(parent.getLeft()==node)
                parent.setLeft(null);
            else
                parent.setRight(null);
        }
        else
            tree=null;
        return tree;
    }
    else
    {
        if (node.getRight()==null)
        {
            if(parent!=null)
            {
                if(parent.getLeft()==node)
                    parent.setLeft(node.getLeft());
                else
                    parent.setRight(node.getLeft());
            }
            else
                tree=node.getLeft();
            return tree;
        }
        else
        {
            if (node.getLeft()==null)
            {
                if(parent!=null)
                {
                    if(parent.getLeft()==node)
                        parent.setLeft(node.getRight());
                    else
                        parent.setRight(node.getRight());
                }
                else
                    tree = node.getRight();
                return tree;
            }
            else
            {
                BinTreeNode<Integer> tmpNode = smallestNode(node);
                node.setInfo(tmpNode.getInfo());
            }
        }
    }
}
```

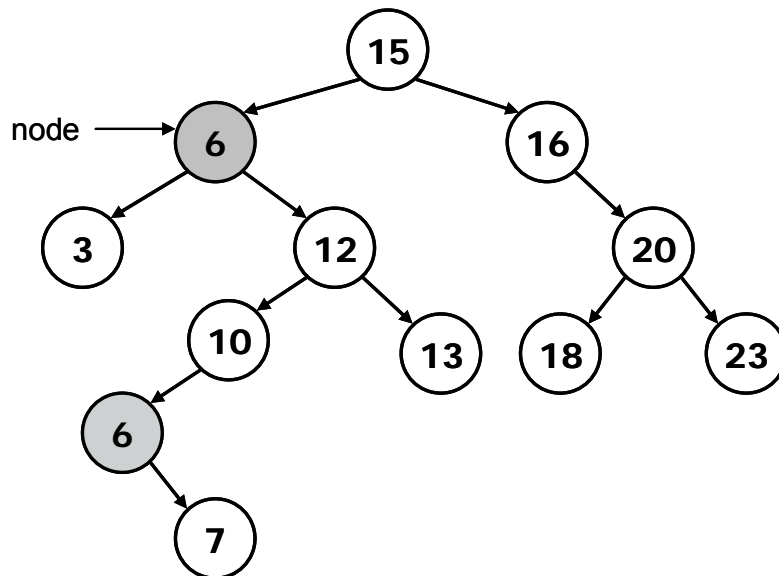
## פעולת העזר:

```
public static BinTreeNode<Integer> smallestNode (BinTreeNode<Integer> node)
{
    if(node.getRight()!=null)
        return smallestNodeHelp(node.getRight());
    else
        return node.getLeft();
}

public static BinTreeNode<Integer> smallestNodeHelp (BinTreeNode<Integer> node)
{
    if(node.getLeft()==null)
        return node;
    else
        return smallestNodeHelp(node.getLeft());
}
```

## משימה 9:

לפניך עץ החיפוש הבינארי לאחר שינוי ערך הצומת שרצינו להסיר:



בעץ לאחר השינוי מופיע הערך 6 פעמיים. פעם בצומת המינימאלי ופעם בצומת שרצינו להסיר. כדי להשלים את תהליך ההסרה יש להסיר את הצומת המינימאלי. הוסף בפעולה delete במקום המתאים הוראה שתסיר את הצומת המינימאלי מעץ החיפוש. (רמז: השתמש בפעולה עצמה).

## דפים מדורגים – הסרת איבר מעץ חיפוש בינארי

### דף מספר 10

תזכורת למשימה מספר 1:

לפניך כותרת פעולה המקבלת עץ חיפוש בינארי (tree) והפניה לצומת בעץ (node) ומוחקת את הצומת הנתון מהעץ תוך שמירת תכונות עץ החיפוש הבינארי.

```
public static void delete (BinTreeNode<Integer> tree, BinTreeNode<Integer> node)
```

לפניך הפיתרון הרקורסיבי המלא:

```
public static BinTreeNode<Integer> delete (BinTreeNode<Integer> tree, BinTreeNode<Integer> node )
{
    BinTreeNode<Integer> parent = parent(tree, node);
    if (isLeaf(node))
    {
        if (parent != null)
        {
            if (parent.getLeft() == node)
                parent.setLeft(null);
            else
                parent.setRight(null);
        }
        tree = null;
        return tree;
    }
    else
    {
        if (node.getRight() == null)
        {
            if (parent != null)
            {
                if (parent.getLeft() == node)
                    parent.setLeft(node.getLeft());
                else
                    parent.setRight(node.getLeft());
            }
            else
                tree = node.getLeft();
            return tree;
        }
        else
        {
            if (node.getLeft() == null)
            {
                if (parent != null)
                {
                    if (parent.getLeft() == node)
                        parent.setLeft(node.getRight());
                    else
                        parent.setRight(node.getRight());
                }
                else
                    tree = node.getRight();
                return tree;
            }
            else
            {
                BinTreeNode<Integer> tmpNode = smallestNode(node);
                node.setInfo(tmpNode.getInfo());
                return delete(tree, tmpNode);
            }
        }
    }
}
```

#### משימות מתקדמות

##### משימה 1:

פעולת ההסרה המלאה השתמשה

ברקורסיה.

שנה את הפעולה כך שנסיר את הצומת

המבוקש ללא רקורסיה.

##### משימה 2:

הצע דרך סימטרית למחיקת איבר (בדומה

למתואר בדפים) שבה מחליף את ערך

הצומת איבר מתת העץ השמאלי של

הצומת.

## דפים מדורגים – הסרת איבר מעץ חיפוש בינארי

### דף עזר

טענת כניסה: הפעולה מקבלת הפנייה לצומת בעץ.

טענת יציאה: הפעולה מחזירה true אם הצומת עלה, אחרת מחזירה false.

```
public static boolean isLeaf (BinTreeNode<Integer> node)
{
    return ((node.getLeft()==null)&&(node.getRight()==null));
}
```

טענת כניסה: הפעולה מקבלת הפנייה לעץ והפנייה לצומת בעץ.

טענת יציאה: הפעולה מחזירה הפנייה לאב של הצומת או null אם הצומת הוא השורש.

```
public static BinTreeNode<Integer> parent
    (BinTreeNode<Integer> tree, BinTreeNode<Integer> node)
{
    if((tree == null) || (node == tree.getLeft())
        || (node == tree.getRight()))
        return tree;

    BinTreeNode<Integer> tmpNode = parent(tree.getLeft(),node);

    if(tmpNode == null)
        return parent(tree.getRight(),node);
    else
        return node;
}
```