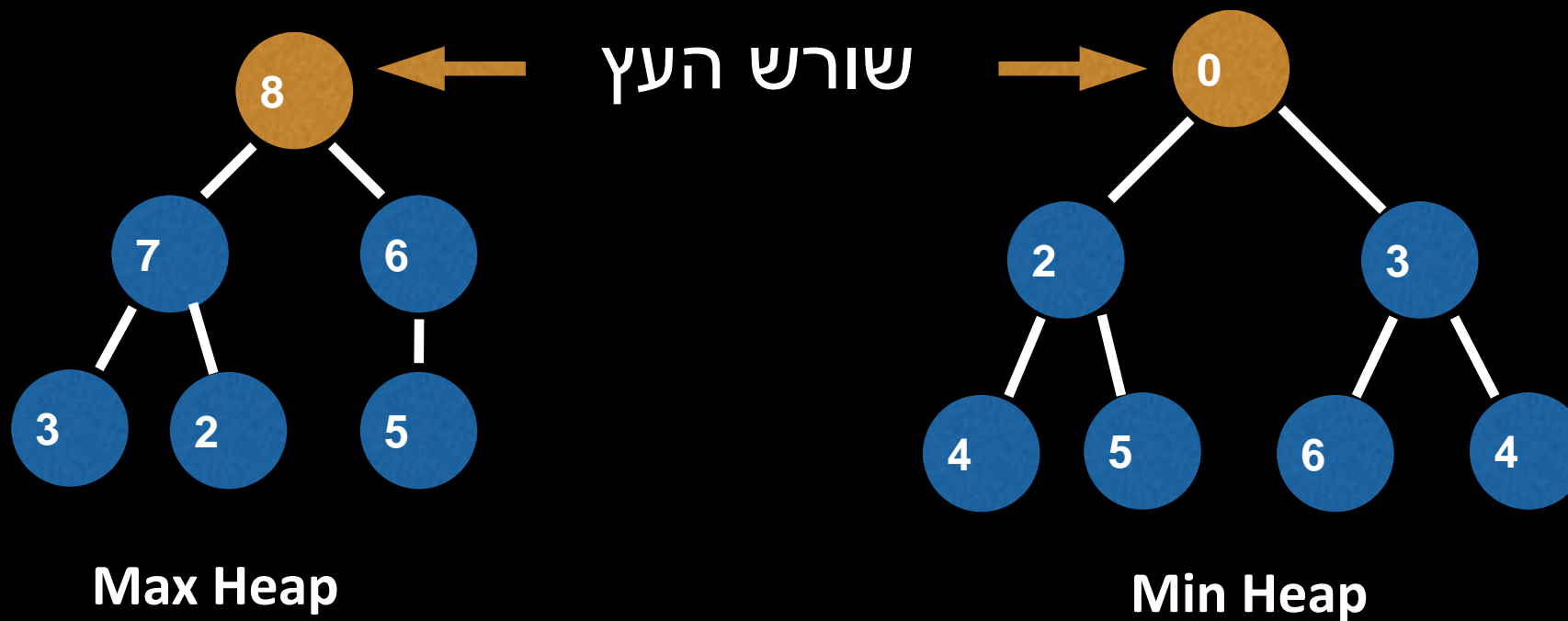


מבני נתונים

תרגול 6 – ערמה

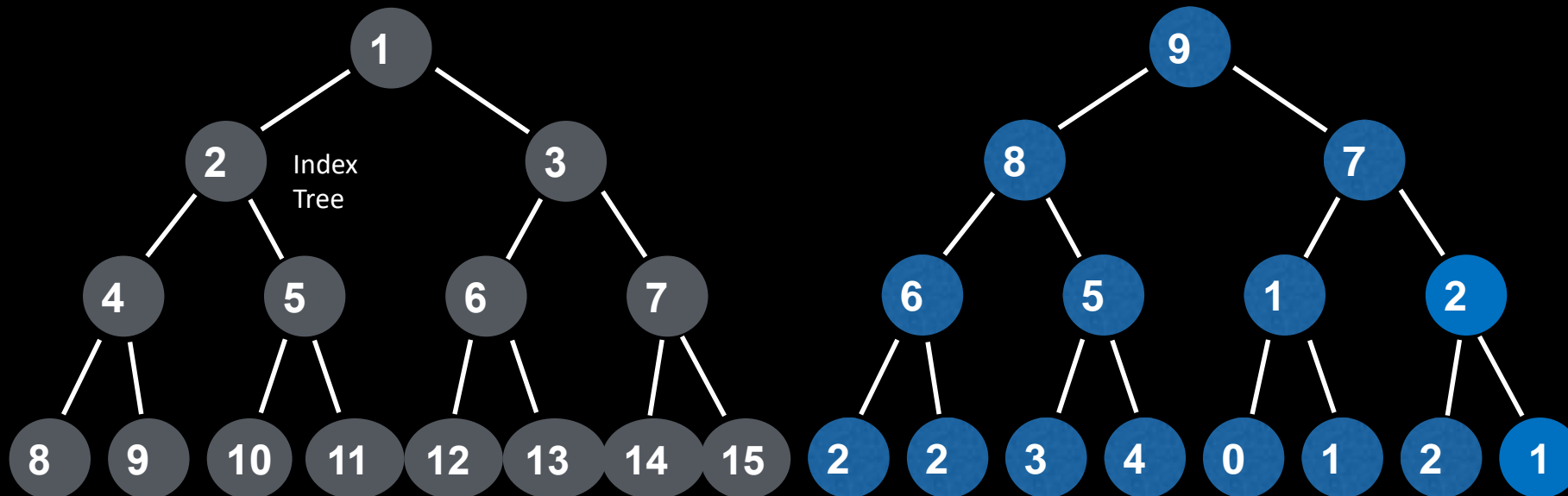
What is a Binary Heap?



עץ בינארי כמעט שלם

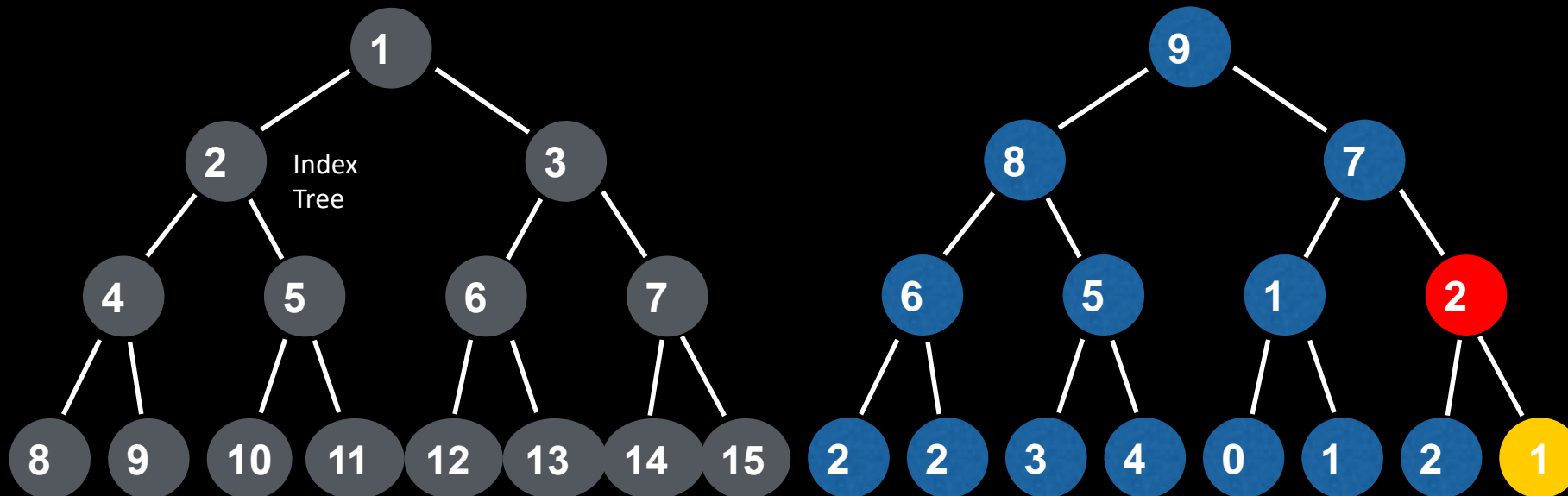
Binary Heap Representation

value	9	8	7	6	5	1	2	2	2	3	4	0	1	2	1
index	1	3	3	4	5	6	7	8	9	10	11	12	13	14	15



Binary Heap Representation

value	9	8	7	6	5	1	2	2	2	3	4	0	1	2	1
index	1	3	3	4	5	6	7	8	9	10	11	12	13	14	15



Binary Heap Representation

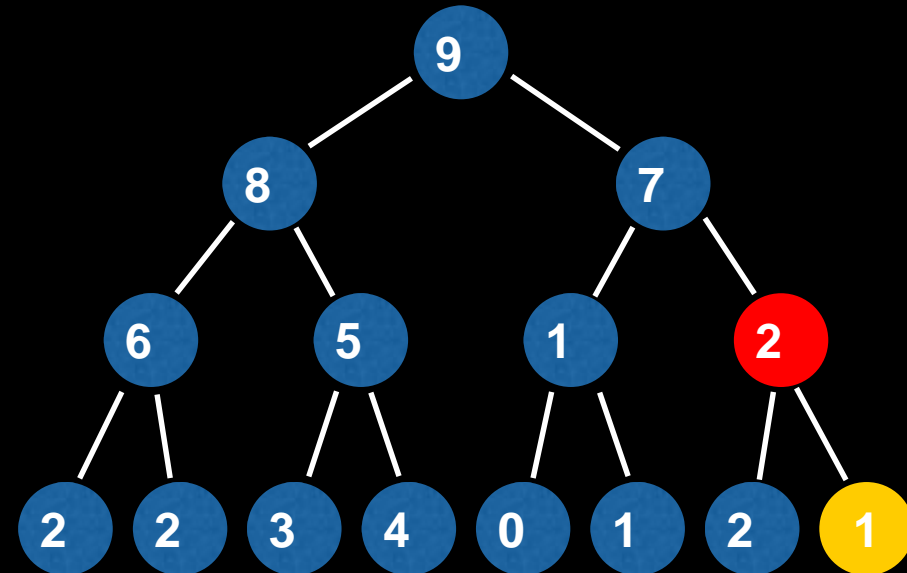
value	9	8	7	6	5	1	2	2	2	3	4	0	1	2	1
index	1	3	3	4	5	6	7	8	9	10	11	12	13	14	15

Let i be a node index
(one based)

Left child index: $2i$

Right child index: $2i + 1$

parent of i index: $\left\lfloor \frac{i}{2} \right\rfloor$



Binary Heap Representation

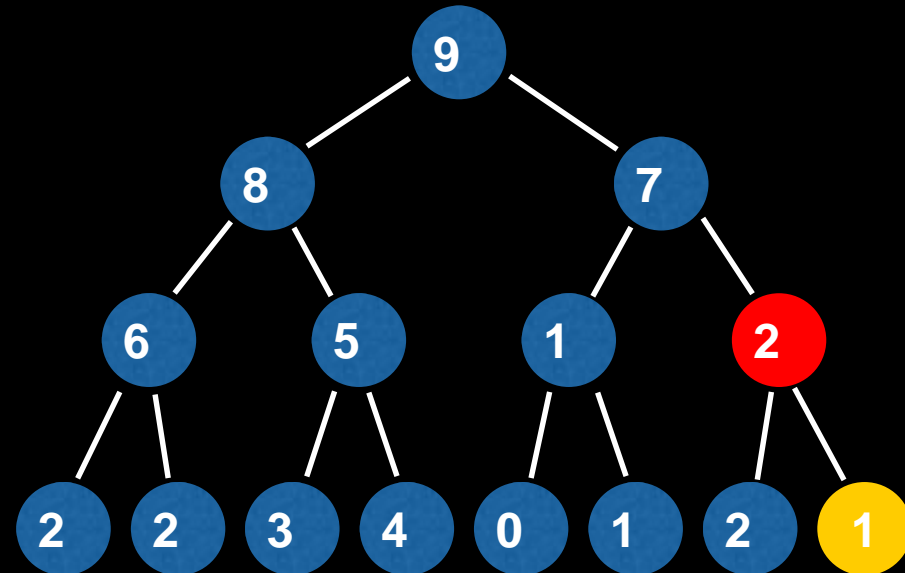
value	9	8	7	6	5	1	2	2	2	3	4	0	1	2	1
index	1	3	3	4	5	6	7	8	9	10	11	12	13	14	15

Let i be a node index
(one based)

Left child index: $2i$

Right child index: $2i + 1$

parent of i index: $\left\lfloor \frac{i}{2} \right\rfloor$



<https://www.cs.usfca.edu/~galles/visualization/Heap.html>

```
private void SwapUp(int index) //  $O(\log n)$  - Insert
```

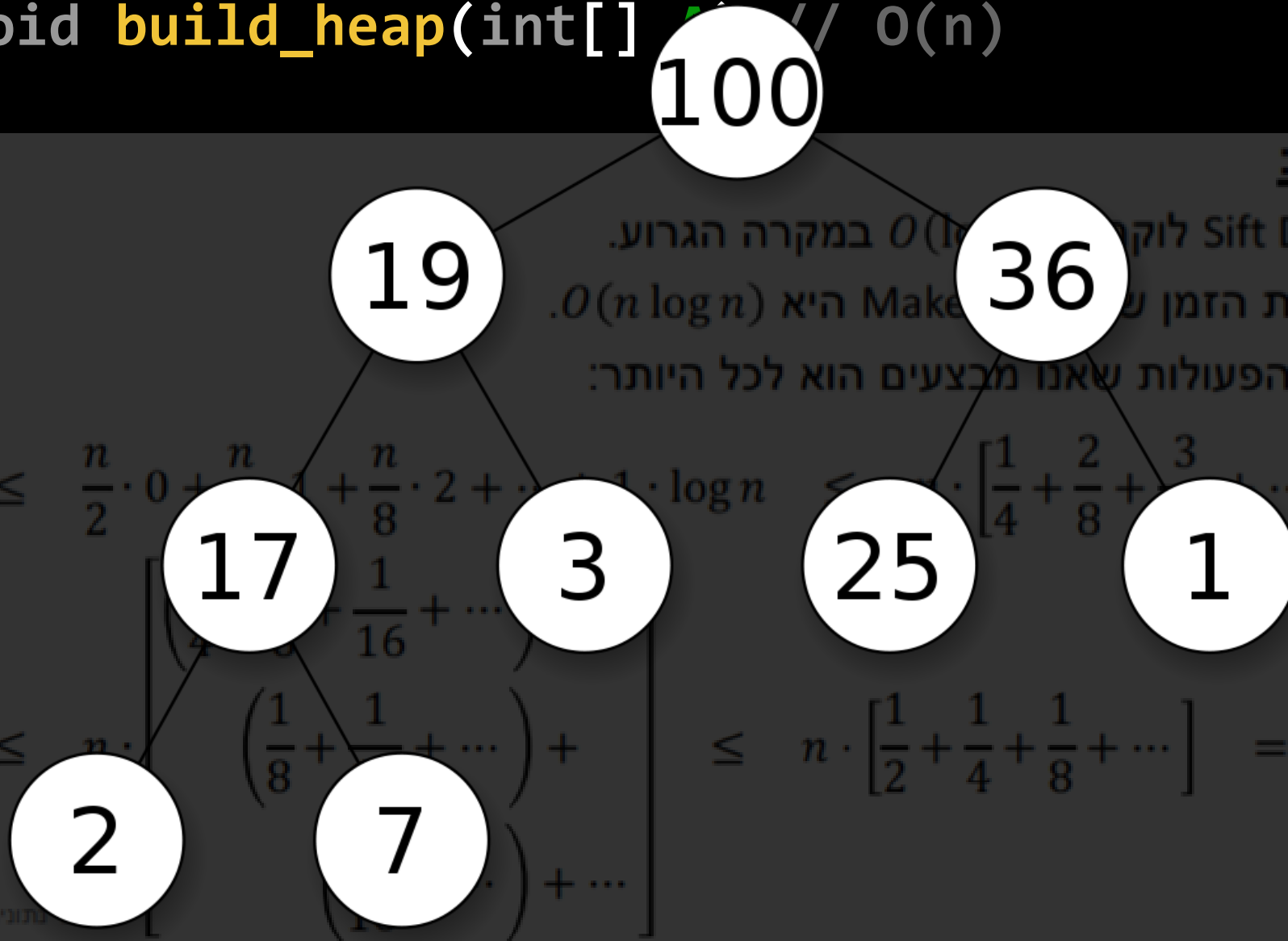
```
public void Heapify(int i) //  $O(\log n)$  - Remove
```

```
public int getMin() //  $O(1)$ 
```

```
public void build_heap(int[] A) //  $O(n)$ 
```

```
public void heapsort(int[] A) //  $O(n \log n)$ 
```

```
public void build_heap(int[] arr) // O(n)
```



ניתוח סיבוכיות:

כל פעולת Sift Down לוקחת $O(\log n)$ במקרה הגרוע.
לכאורה, סיבוכיות הזמן של MakeHeap היא $O(n \log n)$.
למעשה, מספר הפעולות שאנו מבצעים הוא לכל היותר:

$$T(n) \leq \frac{n}{2} \cdot 0 + \frac{n}{4} \cdot 1 + \frac{n}{8} \cdot 2 + \dots + 1 \cdot \log n \leq n \cdot \left[\frac{1}{4} + \frac{2}{8} + \frac{3}{16} + \dots \right] \leq n \cdot \left[\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots \right] = n$$

מִימוֹשׁ MaxHeap.java

```
public class MaxHeap  
{
```



```
...  
}
```

שאלה 1

נתון מערך של מספרים שלמים: 12, 19, 10, 4, 23, 7, 45, 8, 15.
סרטט Max-Heap עבור המספרים האלה.

שאלה 2 סרטט Min-Heap עבור המספרים של שאלה 1.

<http://btv.melezinek.cz/binary-heap.html>

מאינדקס 1

שאלה 3

- (א) כמה עליים יש לעץ ערמה? $\left\lceil \frac{n}{2} \right\rceil$
- (ב) מהו האינדקס של העלה הראשון? $\left\lceil \frac{n}{2} \right\rceil + 1 = \left\lceil \frac{n}{2} \right\rceil$
- (ג) האם המערך שבו נשמר עץ ערמה הוא חייב להיות ממוין? **לא**

שאלה 5 (20 נקודות)

א. המערך הבא מציג עץ ערמה (משמאל לימין): {8, 18, 21, 25, 28, 22, 30, 37, 36, 39, 29}.

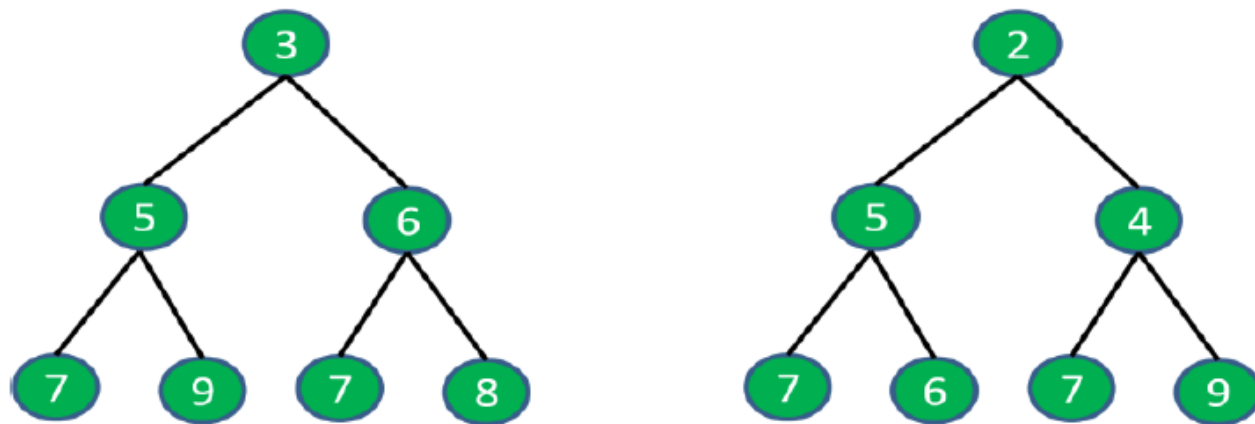
צייר את העץ. **תרגיל בית**

ב. צייר את העץ ואת המערך אחרי מחיקת האיבר הקטן. מהו זמן הריצה של מחיקה? **תרגיל בית**

ג. נניח ש-A, B הם שני עצי ערמה, ובמקרה העצים של שניהם שלמים וגם הגבהים שלהם זהים. כתוב אלגוריתם יעיל לאיחוד A, B לעץ ערמה חדש C, מהו זמן הריצה של האלגוריתם שכתבת? **הוכח!**

שימו לב: לצורך האלגוריתם, יש להניח שעצי-הערמה מיוצגים כעצים, ולא במערך. ניתן להניח שכל צומת מכיר את שני הבנים שלו, והאלגוריתם מקבל את השורשים של שני העצים.

דוגמה:



```
// Merges max heaps a[] and b[] into arr[]
public static void mergeHeaps(int[] arr, int[] a,
                             int[] b, int n, int m) {

    for (int i = 0; i < n; i++) {
        arr[i] = a[i];
    }
    for (int i = 0; i < m; i++) {
        arr[n + i] = b[i];
    }
    n = n + m;

    // Builds a max heap of given arr[0..n-1]
    for (int i = n / 2 - 1; i >= 0; i--) {
        maxHeapify(arr, n, i);
    }
}
```

שאלה 1 (20 נקודות). כתוב מחלקה MaxHeap שמהווה עץ ערמה של מספרים שלמים, ממומש על ידי מערך פנימי.

1. הגדר את המחלקה ואת השדות (המשתנים) שלה.

2. הוסף בנאי שמקבל גודל למערך הפנימי:
`public MaxHeap(int capacity)`

3. הוסף פונקציה להכנסת איבר:
`public boolean add(int x)`

4. הוסף פונקציה שמחזירה ומוחקת את האיבר הגדול באוסף:
`public int remove()`

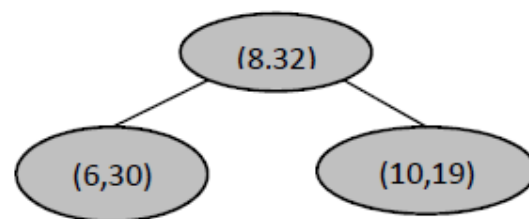
```
public class MaxHeap {}
```

שאלה 3 (20 נקודות)

ערימה הוא עץ בינרי בו הערך של כל צומת גדול או שווה לערך בנו. (שימו לב – לא דרשנו בהגדרה זו עץ בינרי כמעט שלם).

יהא T עץ בינרי, שבו כל צומת מכיל 2 ערכים: מפתח k ועדיפות p . העץ T נקרא **Treap** אם T הוא עץ חיפוש בינרי ביחס למפתחות, וערימה ביחס לעדיפויות.

לדוגמה:



כאשר הערך השמאלי בכל צומת הוא המפתח, והערך הימני הוא העדיפות (k,p) .

א. נתונה הקבוצה הבאה של זוגות סדורים (האיבר הראשון (השמאלי) בכל זוג הוא המפתח, והאיבר השני (הימני) בכל זוג הוא העדיפות):

$\{(5,34), (2,13), (8,26), (6,19), (7,38), (9,14), (11,27), (10,22)\}$

צייר Treap עבור הקבוצה הנ"ל.

ב. בהנתן קבוצה של זוגות סדורים המורכבים ממפתח ועדיפות בה כל המפתחות שונים זה מזה, וכל העדיפויות שונות זו מזו, כתוב אלגוריתם לבניית Treap.

$(2,13), (9,14), (6,19), (10,22),$
 $(8,26), (11,27), (5,34), (7,38)$

$(7,38)$
 $(5,34) \quad (11,27)$
 $(2,13) \quad (8,26) \quad (6,19)$
 $(10,22)$
 $(9,14)$