

Q1. (20%)

- a. What is the running time of heapsort if all the elements have the same value?
- b. What is the exact height of a complete binary tree with n nodes?
Assume that the roots' level is 0.

Explain your answers.

- א. מה זמן הריצה של מיון ערמה אם לכל האיברים אותו הערך?
- ב. מה הגובה (המדויק) של עץ בינרי כמעט שלם בעל n צמתים? הניחי שרמת השורש היא 0.
- הסבירי את תשובתייך.

Q2. (20%)

Write a C function that receives an array of integers a of size n , which represents a complete binary tree and prints the post-order-traversal of the tree.

The header of the function is:

```
void postorder (int a[], int n)
```

The running time is $O(n)$.

כתבי פונקציה ב-C שמקבלת מערך של שלמים a בגודל n , שמייצג עץ בינרי כמעט שלם ומדפיסה סריקה בסדר סופי של העץ.

כותרת הפונקציה היא:

```
void postorder (int a[], int n)
```

זמן הריצה שלה $O(n)$.

תשובה:

הפונקציה קוראת לפונקציה רקורסיבית המבצעת את הסריקה.

```
void postorder (int a[], int n)
{
    recorder(a, n, 0);
}

void recorder (int a[], int n, int i)
{
    if (i < n)
    {
        recorder(a, n, 2*i+1);
        recorder(a, n, 2*i+2);
        printf("%i ", a[i]);
    }
}
```

Q3. (20%)

Let p be a node in a balanced binary search tree. The key of p is k . Write, in pseudocode, an algorithm that returns true if and only if p is the only node with key k .

The running time of the algorithm is $O(\lg n)$, where n is the number of nodes in the tree.

יהי p צמת בעץ חיפוש בינרי מאוזן. הערך ב- p הוא k . כתבי, בפסידוקוד, אלגוריתם המחזיר true אם ורק אם p הוא הצמת היחיד בעל הערך k .
זמן הריצה של האלגוריתם הוא $O(\lg n)$, כאשר n הוא מספר הצמתים בעץ.

תשובה:

```
find(p)
// find the succ. of p
if right[p]≠nil
then q←right[p]
    while left[q]≠nil do
        q←left[q]
else q←p
    while parent[q]≠nil and left[parent[q]]≠q do
        q←parent[q]
if q≠nil and key[q]=key[p]
then return true
// find the prev. of p
if left[p]≠nil
then q←left[p]
    while right[q]≠nil do
        q←right[q]
else q←p
    while parent[q]≠nil and right[parent[q]]≠q do
        q←parent[q]
if q≠nil and key[q]=key[p]
then return true
return false
```

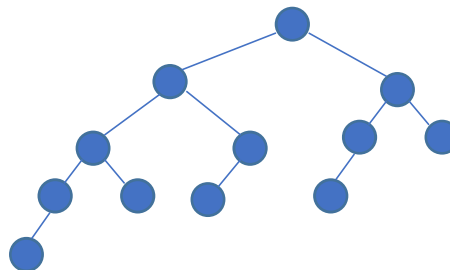
Q4. (20%)

- Draw the highest balanced binary tree with 12 nodes.
- Draw the balanced tree of height 5 (the root's level is 0) with minimal number of nodes.

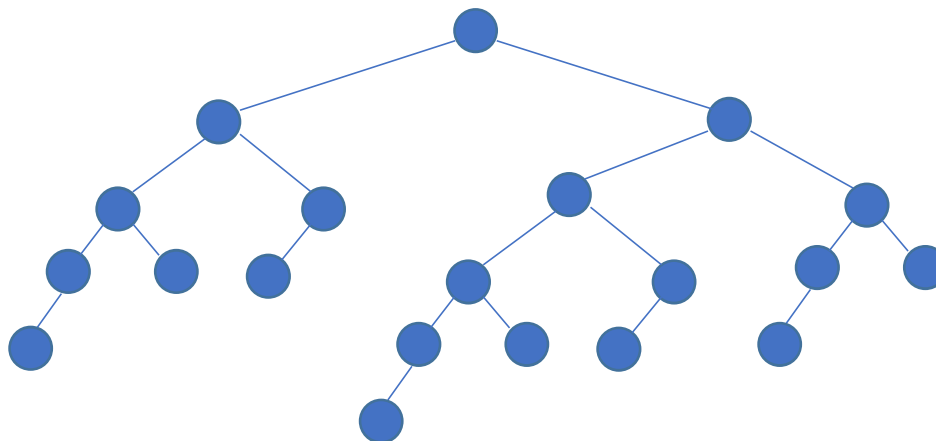
א. ציירי את העץ הבינרי המאוזן הגבוה ביותר בעל 12 צמתים.
ב. ציירי עץ מאוזן בגובה 5 (רמת השורש היא 0) בעל מספר קטן ביותר של צמתים.

תשובה:

a.



b.



לכל עץ 10 נקודות

Q5. (20%)

Whereas a stack allows insertion and deletion of elements at only one end, and a queue allows insertion at one end and deletion at the other end, a deque (double-ended queue) allows insertion and deletion at both ends. Write four $O(1)$ -time procedures, in pseudocode, to insert elements into and delete elements from both ends of a deque constructed from an array.

בעוד מחסנית מאפשרת הכנסה והוצאה של איברים מקצה אחד בלבד, ותור מאפשר הכנסה בצד אחד והוצאה מהצד השני, דו-תור מאפשר הכנסה והוצאה משני הצדדים. כתבי ארבע שגרות בפסידוקוד, שזמן ריצתן $O(1)$, להכנסה לשני הצדדים ולהוצאה משני הצדדים של דו-תור הממומש במערך.

תשובה:

המערך a יכיל את אברי המערך, כאשר $a[0]$ יכיל את מספר האיברים בדו-תור, ו- $a[1]$ יכיל את האינדקס של ראש הדו-תור.

```
inserthead(a, x)
if a[0]=size[a]-2
then OVERFLOW
a[0]←a[0]+1
a[1]←a[1]-1
if a[1]=1
then a[1]←size[a]-1
a[a[1]]←x
```

```
removehead(a)
if a[0]=0
then UNDERFLOW
x←a[a[1]]
a[0]←a[0]-1
a[1]←a[1]+1
if a[1]=size[a]
then a[1]←2
return x
```

```
inserttail(a, x)
if a[0]=size[a]-2
then OVERFLOW
i←a[1]+a[0]
if i>=size[a]
then i←i-size[a]+2
a[i]←x
a[0]←a[0]+1
```

```
removetail(a)
if a[0]=0
then UNDERFLOW
a[0]←a[0]-1
i←a[1]+a[0]
if i>=size[a]
then i←i-size[a]+2
return a[i]
```