בס״ד

המרכז האקדמי לב

שנה״ל **תשע״ח**, סמסטר **ב**, מועד **א**

שאלון בחינה בקורס: מבנה נתונים - Data Structures I

מספר קורס: **150015**

Student ID Number: _____ Campus: Lev To be filled by student

Instructor's Name: Meir Komar
Date of Exam: 26/3/2018 י' ניסן תשע"ח
Length of Exam (in minutes): 180
Materials Allowed During Exam: None
This exam has 6 questions each worth 20 points. Answer any 5 of them. If you answer 6 only the first 5 will be marked.

Answers must be written on answer sheets provided. Draft notes will not be checked.

1. **If for any reason you are not certain if you understand the intent of the instructor with regard to a question, you are to indicate at the beginning of your answer how you understand the question and answer accordingly. The instructor has the right to determine how many points your understanding and answer are worth.**
2. **You must return the test form with the notebook (in the case where there is a notebook). If the form is not attached the test will not be graded.**
3. **It is your responsibility to understand the institutional regulations regarding exams. Any deviation from those rules may lead to the exam being disallowed and other actions taken against student.**
4. **Please note that points will be taken off not only for mistakes but for irrelevant material in answer, lack of adequate explanation for an answer, an unclear and/or ambiguous answer.**

Good Luck!

1	2	3	4	5	6	Total

המרכז האקדמי לב
שנה״ל **תשע״ח**, סמסטר **ב**, מועד **א**
שאלון בחינה בקורס: מבנה נתונים - Data Structures I
מספר קורס: **150015**

General comments:

1. Unless otherwise stated, when asked for run time, the meaning is the worst case order of complexity.
2. Anywhere written without any additional memory, you are allowed to use constant size memory.
3. In questions where you need to write algorithms, you can use algorithms taught in the lecture and you do not need to implement them.

Question 1

Answer only 2 of the following 3 parts (10 points for each part)
Prove or disprove:

a. $n^{1+logn} = O(n^{logn})$

b. $\log((n^2)!) = \theta((nlogn)^2)$

c. $(\sqrt{n})^n = \omega(n^{\sqrt{n}})$

Question 2 (5 pts for each part)

a. Suppose we have a maxheap H and two values v1 and v2, such that all values are distinct. Let H12 be the heap you get if you insert v1 and then v2 into H, and H21 be the heap you get if you insert v2 and then v1 into H. Give an example of H, v1 and v2 such that H12 = H21. No justification needed, just draw the heaps H, H12 and H21 and specify which is v1 and which is v2.

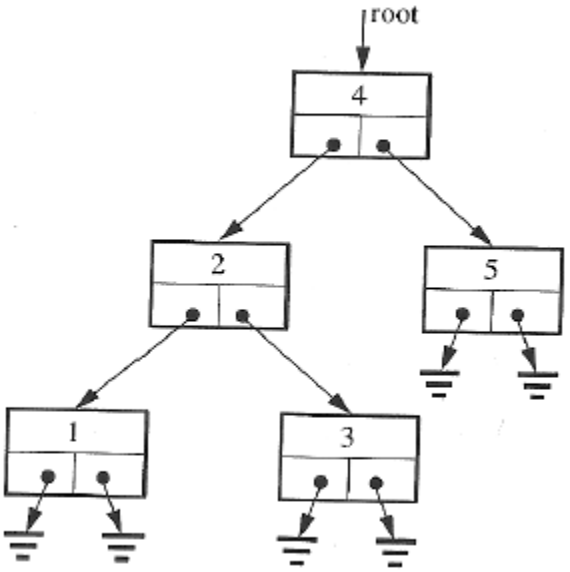
b. Suppose we have a maxheap of height h, such that all values are distinct. At what level will you find the successor of the successor of the minimal value in the heap (the third smallest value)? Explain your answer.

c. Suppose we have a maxheap H containing n keys, such that all keys are distinct. Given: A is the series of keys of a preorder transversal of H.
Complete the maximal number for the following sentence.
The minimal key is definitely not in the first _____ keys in A.
Be exact in your answer
Explain your answer:

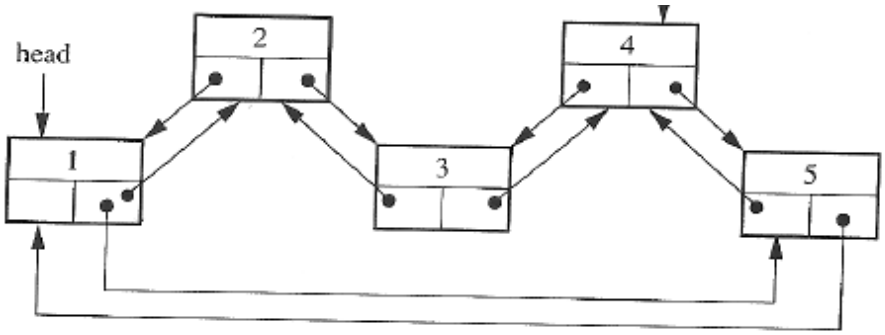
d. Given: a fully binary tree with n nodes. We know that at level 3 there are (n+1)/4 nodes.
What is the height of the tree? _____
Explain your answer.

המרכז האקדמי לב
שנה״ל **תשע״ח**, סמסטר **ב**, מועד **א**
שאלון בחינה בקורס: מבנה נתונים - Data Structures I
מספר קורס: **150015**

Question 3
Given: A binary search tree. We want to create a circular double linked list in ascending order according to the values in the tree in $\Theta(n)$ time.
For example, for the following binary search tree



The following circular double linked list will be created.



You can assume that the binary tree is comprised of nodes. Each node contains a key and pointers to the left son (left) and a pointer to the right son (right).
In addition, the data structure, circular double linked list contains links. Each link is comprised of a key, a pointer to the next link (next) and a pointer to the previous link (prev), and the function head(L) returns a pointer to the head of the list.

a. (16 points) Write in pseudocode a function TreeToList that receives, root, the root of the binary search tree and L an empty circular double linked list.
   - The function must run in $\Theta(n)$ time where n is the number of nodes in the tree.
   - The initial call to the function will be: TreeToList(root, L) where:
     - root is a pointer to the root of the binary search tree
     - L is an empty circular double linked list
   - On completion, the links of the list will contain the keys of the binary search tree in ascending order.
   - If needed you may add additional functions for the tree or the list as long as they run in $\Theta(n)$ time.

   Explain your function.

b. Explain what is the run time complexity of your function.

Question 4

a. (10 points) What is the run time complexity of following function? Explain your answer in terms of n using Θ.

```
x=0
    for ( i = 1 ; i ≤ n ; i++)
      for  (j = 2i ; j ≤ n ; j++)
           x++
```

b. (10 points) Given: The following 2 functions:

```
func1(n)
    Q is a new queue
    enqueue(Q,0)
    enqueue(Q,1)
    func2(Q,n)
    x = dequeue(Q)
    return x

func2(Q,n)
    if n>0
        a = dequeue(Q)
        b = dequeue(Q)
        enqueue(Q,b)
        enqueue(Q,a + b)
        func2(Q,n-1)
```

What will func1(n) return? _____
Explain your answer

Question 5

a. (5 points) What is the maximal height of a decision tree for a comparison based sort of an array containing n elements? Prove your answer.
Reminder : $log(n!) = \theta(nlogn)$

For the following parts: You are given a **single linked list** of size n, containing natural numbers in the range of 1 to n (including). Note: not all the numbers have to be in the list and the same number can be more than once.

b. (5 points) Is there any sorting algorithm that we learnt that can sort the list in Θ(n) time in the worst case using only O(1) additional space? _____
Explain.

c. (5 points) Is there any sorting algorithm that we learnt that can sort the list in Θ(nlogn) time in the worst case using only O(1) additional space? _____
Explain.

d. (5 points) Suggest an algorithm that finds the number of distinct values in the list in Θ(n) time.

Question 6

a. (7 points) Build an AVL tree with the following values:
5, 10, 9, 2, 8, 6, 11, 4, 1, 3, 7
Draw the tree after each insertion and if you do a rotation specify which rotation you did.

b. (3 points) Delete from the tree the following values: 5, 7, 6
Draw the tree after each deletion and if you do a rotation specify which rotation you did.

c. (10 points) What is the run time complexity of the following function. Explain your answer in terms of n using Θ.

```
func(n)
   x  = 0
   i=1
   while( i< (log n)^n )
            x = x+1
            i = i*2
```

המרכז האקדמי לב
שנה״ל **תשע״ח**, סמסטר **ב**, מועד **א**
שאלון בחינה בקורס: מבנה נתונים - Data Structures I
מספר קורס: **150015**

Formula Sheet

Definition of O : Given two functions f(n), g(n): $\mathbb{N} \to \mathbb{R}^+$
We say that g(n) is O(f(n)) if there are positive constants n_0 and c such that
g(n) ≤ c*f(n) for all $n \geq n_0$

Definition of Ω : Given two functions f(n), g(n): $\mathbb{N} \to \mathbb{R}^+$
We say that g(n) is Ω (f(n)) if there are positive constants n_0 and c such that
g(n) ≥ c*f(n) for all $n \geq n_0$

Definition of Θ : Given two functions f(n), g(n): $\mathbb{N} \to \mathbb{R}^+$
We say that g(n) is Θ (f(n)) if there are positive constants n_0 , c_1, c_2 and c such that
c_1f(n) ≤ g(n) ≤ c_2f(n) for all $n \geq n_0$

Definition of o : Given two functions f(n), g(n): $\mathbb{N} \to \mathbb{R}^+$
We say that g(n) is o(f(n)) if there are positive constants n_0 and c such that
g(n) < c*f(n) for all $n \geq n_0$

Definition of ω : Given two functions f(n), g(n): $\mathbb{N} \to \mathbb{R}^+$
We say that g(n) is ω (f(n)) if there are positive constants n_0 and c such that
g(n) > c*f(n) for all $n \geq n_0$

Arithmetic Series

$$\sum_{k=1}^{n} k = 1 + 2 + \ldots + n = \frac{1}{2} n(n+1)$$

Geometric Series

$$\sum_{k=0}^{n} x^k = 1 + x + x^2 + \ldots + x^n = \frac{x^{n+1} - 1}{x - 1}$$

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x} \qquad |x| < 1 \qquad \text{עבור}$$

Harmonic Series

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots + \frac{1}{n} = \sum_{k=1}^{n} \frac{1}{k} = \ln n + O(1)$$

המרכז האקדמי לב

שנה״ל **תשע״ח**, סמסטר **ב**, מועד **א**

שאלון בחינה בקורס: מבנה נתונים - Data Structures I

מספר קורס: **150015**

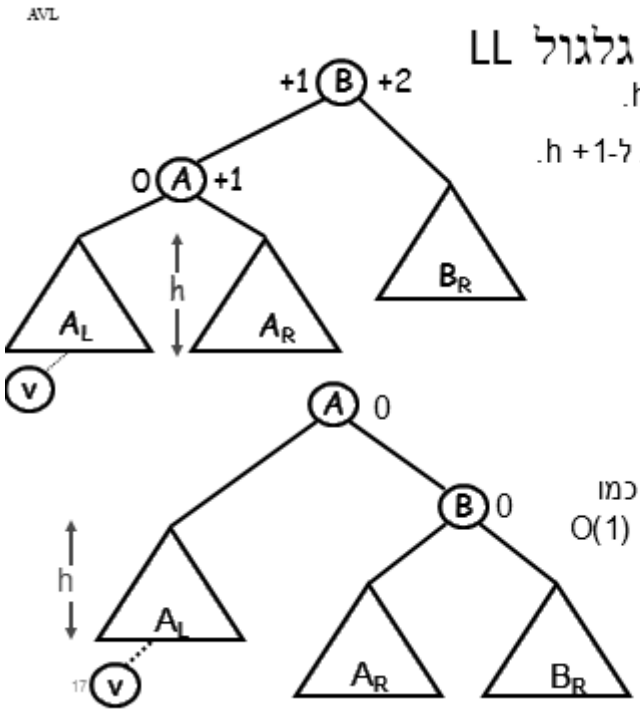
Series of Squares

סדרת הריבועים: $\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$

שינוי בסיס log: $\frac{\log_c a}{\log_c b} = \log_b a$

שינוי חזקה: $n^{\log_c a} = a^{\log_c n}$

כלל לופיטל: $\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}$



לפני הכנסת v: גובה העץ הוא h+2.

הוכנס צומת v שהגדיל את גובה A_L ל-h+1.

מצד ימין של הצמתים מסומנים גורמי האיזון שהשתנו.

גלגול LL: יעביר את A לשורש

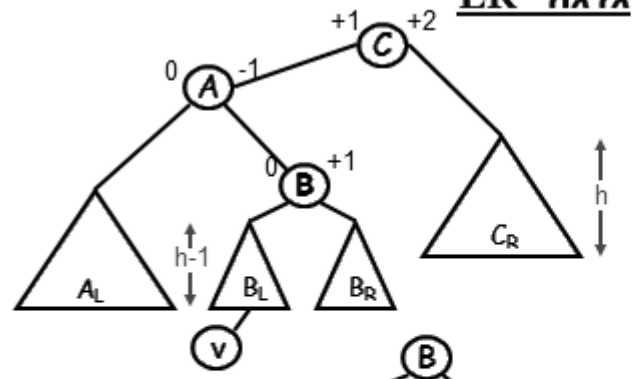
אחרי הגלגול:

גובה העץ לאחר הגלגול הוא h+2, כמו לפני ההכנסה. השורש מאוזן. שינינו O(1) מצביעים ולכן זמן הגלגול O(1).

המרכז האקדמי לב
שנה״ל **תשע״ח**, סמסטר **ב**, מועד **א**
שאלון בחינה בקורס: מבנה נתונים - Data Structures I
מספר קורס: **150015**

AVL

גלגול LR



לפני הכנסת איבר v:

הוכנס איבר ל-B_L שגרם לו להעלות את גובהו ל-h.

גלגול LR:

גובה העץ אחרי הגלגול הוא h+2, כמו לפני ההכנסה.

שינוי O(1) מצביעים ולכן זמן הגלגול O(1).