**HOME WORK**

**Homework based on Lecture 1.1**

1. When an algorithm is written in the form of a programming language, it becomes a \_\_\_\_\_\_\_\_\_  
   a) Flowchart  
   **b) Program**  
   c) Pseudo code  
   d) Syntax
2. The time that depends on the input: an already sorted sequence that is easier to sort.  
   a) Process  
   b) Evaluation  
   **c) Running**  
   d) Input
3. In computer science, algorithm refers to a special method usable by a computer for the solution to a problem.
   1. **True**
   2. False

**Homework based on Lecture 1.2**

1. Consider the following three claims

1. (n + k)m = Θ(nm), where k and m are constants

2. 2n + 1 = O(2n)

3. 22n + 1 = O(2n)

Which of these claims are correct?

1. **1 and 2**
2. 1 and 3
3. 2 and 3
4. 1, 2, and 3
5. What is the time complexity of following code:

int i, j, k = 0;

for (i = n / 2; i <= n; i++)

{

     for (j = 2; j <= n; j = j \* 2)

{

         k = k + n / 2;

     }

}

A) O(n)

**B) O(nLogn)**

C) O(n^2)

D) O(n^2Logn)

**Homework based on Lecture 1.3**

1. What is the complexity of T(n)?



1. Θ ( 1⁄n )
2. Θ ( 1⁄n2 )
3. **Θ ( 1 )**
4. Θ ( ln( n ) )
5. The substitution method is the only method to solve the recurrence relation?
6. True
7. **False**

**Homework based on Lecture 1.4**

1. Consider the recurrence relation

T(n) = T(n/4) + T(n/2) + cn2

What is the result of the recurrences which fall under Recursive tree method?

1. O(n logn)
2. O(N3)
3. **O(n2)**
4. O(log n)
5. What is the time complexity of Tower of Hanoi problem?

A) T(n) = O(sqrt(n))  
(D) T(n) = O(n^2)  
**(C) T(n) = O(2^n)**  
(D) None

**Homework based on Lecture 1.5**

1. What is the result of the recurrences which fall under first case of Master’s theorem (let the recurrence be given by T(n)=aT(n/b)+f(n) and f(n)=nc?  
   **a) T(n) = O(n^logba)**  
   b) T(n) = O(nc log n)  
   c) T(n) = O(f(n))  
   d) T(n) = O(n2)
2. We can solve any recurrence by using Master’s theorem.  
   a) true  
   **b) false**
3. What is the result of the recurrences which fall under the extended second case of Master’s theorem (let the recurrence be given by T(n)=aT(n/b)+f(n) and f(n)=nc(logn)k?

a) T(n) = O(nlogba)  
b) T(n) = O(nc log n)  
**c) T(n)= O(nc (log n)k+1**  
d) T(n) = O(n2)