150015-Semester II - 5781

Data Structures I

**Homework Assignment #2**

**Question 1**

Arrange the following functions in ascending asymptotic order. Explain the reasons for the order.

Prove 2 cases.

5, lg2n, nlgn, nn, (n+1)!, lglgn, n2.5, n2, 5n, 5n

1. Θ(loglogn)
2. Θ(5)
3. Θ(log²n)
4. Θ(nlogn)
5. Θ(n²)
6. Θ(n^2.5)
7. Θ(5n)
8. Θ(5^n)
9. Θ((n+1)!)

Loglogn is going from negativity to lim = -0. It never raises above 0

(5) never raises, even now it is highest, because it never raises its point it becomes more and more middle.

**Question 2**

For each pair of functions, determine whether f(n) = O(g(n)), or f (n) = Ω(g(n)), or

Θ(g(n)). Prove your answer.

1. *f*(*n*) = *g*(*n*) =

O(g(n)) since n^1+n^logn is going one n number higher as n^logn

b. *f*(*n*) = *g*(*n*) =

from n E (0, 1)U(~4,6, +inv) f(n) = omega(g(n)) but between n E (1, ~4,6) f(n)= w(g(n))

c. *f*(*n*) = *g*(*n*) =

f(n) = Θ(g(n)), since both go to parallel lines

**Question 3**

Analyze the run time complexity of the following function?

Θ(1)

Θ(1)

Θ(n-2)

Θ(1)+Θ(1)

State the run time complexity as a function of n using θ, and explain your answer. You should write the function as simple as possibille.

Θ(2)+ Θ((n-2)x(2))

**Question 4**

Prove or disprove the following:

*f*(*n*), *g*(n) are positive functions

* 1. if *f*(*n*) = *o*(*g*(*n*)) then 2*f*(*n*) = *O*(2*g*(*n*))

No, o(g(n)) means f(n) <g(n) and when we power it both 2 and the power of the function 2^f(n) and 2^g(n) than (f(n)=o(g(n))) is 2 power with it less higher than with 2^f(n) and 2^g(n)

Meaning: 2^f(n) = o(2^g(n))

* 1. *f*(*n*) = *O*(*f*(*n*)2)

yes, if function of f(n) is n=1 than resultion would be for f(n)=1 and f(n)² will be 1 == stays the same. But when it goes higher if n>1 it immediately f(n)² becomes higher value as f(n)

* 1. *f*(*n*) = Θ(*f*(*n*/2))

*No, the value of f(n) and the value of g(n/2) (if f(n/2) = g(n/2)) than f(n) = omega(g(n/2)) since only by n = 0 it is the same but if n>0 than f(n) > g(n/2).*

* 1. *n*1/3 = Θ(*n*1/4)

*yes, n^1/3 is parallel of n^1/4*

* 1. *n*! = Θ(*n*lg *n*)

incorrect, since n! its line goes in a very strange way, but in n log n it goes in a bounded wave between 0,5 and -0,5