These problems will help you understand the Analysis of Algorithms (read CLRS)

Problem 2. (True or False): No comparison-based sorting algorithm can do better than *Ω(n log n)* in the worst-case

True because the algorithms must perform Ω(n log n) to sort n elements in the worst case. Lower bounds like Ω(n log n) help us figure out how to get the best solution. The other sorting algorithm worst case other than O(n log n) is O(n^2), which takes more longer than O(n log n). Also, this would exclude sorts like counting and radix.

Problem 3. We can extend our notion to the case of two parameters n and m that can go to infinity independently at different rates. For a given function *g(n, m)*, we donate by *O(g(n, m))* the set of functions

O(g(n,m)) = { *f(n, m)* : there exist positive constants *c*, n_0, and m_0, such that
0 <= f(n, m) <= cg(n, m)
for all $n >= n_0$ or $m >= m_0$}

Give corresponding definitions for *Ω(g(n, m))* and *Θ(g(n, m))*

Ω(g(n, m)) = { f(n, m) : there exist positive constants *c*, n_0, and m_0, such that
0 ≤ cg(n, m) ≤ f(n, m)
for all $n >= n_0$ or $m >= m_0$ }

Θ(g(n, m)) = { f(n, m) : there exist positive constants c_1, and c_2, n_0, and m_0, such that
c_1g(n, m) ≤ f(n, m) ≤ c_2g(n, m)
for all $n >= n_0$ or $m >= m_0$ }.

Problem 4. Consider functions *f(n)* and *g(n)* as given below. Use the most precise asymptotic notation to show how function *f* is related to function g in each case (i.e. , *f* ∈ *?(g)*). For example, if you were given the pair of functions $f(n) = n$ and $g(n) = n^2$ then the correct answer would be: *f* ∈ *o(g)*. To avoid any ambiguity between *O(g)* and *o(g)* notations due to writing, use *Big-O(g)* instead of *O(g).*

1. **f = Θ(g)**
Because for c1 = 1, c2 = 200, and n >= 0,
0 <= g(n) <= f(n) <= 200 * g(n)

2. **f = Ω(g)**
Because for large n values,
0 <= g(n) <= f(n)

3.**f = Ω(g)**
Because for n >= 0,
$\log_2 n > \log_3 n$

4. **f = Big-O(g)**
Because for n >= 0,
2^n <= 3n

5. **f = Big-O(g)**
Because for n >= 0,
0.5^n < 1

Problem 5. Indicate, for each pair of expressions (A, B) in the table below, whether A is O, Ω, Θ, o, ω of B. Assume that k>= 1, ϵ > 0 and *c* >1 are constant. Your answer should be in the form of the table with "yes" or "no" written in each box.

A	B	O	Ω	Θ	o	ω
(lg^k)n	n^(ϵ)	yes	no	no	yes	no
n^k	c^n	yes	no	no	yes	no
sqrt(n)	n^(sin(n))	no	no	no	no	no
2^n	2^(n/2)	no	yes	no	no	yes
n^(lgc)	c^(lg(n))	yes	yes	yes	no	no
lg(n!)	lg(n^n)	yes	yes	yes	no	no