Exam 1.
Name:
Student ID:

Part C (You need to answer at least **6** questions completely to pass Part C**)**

1- Compute:

a) $\sum_{i=-10}^{n}(1/2)^i + \sum_{i=200}^{n^2}(3)^i$

b) $7^{loglog4} + log_3 log^2 8$

2- Use L'Hopital's rule to determine the limit of:

$$\lim_{x \to \infty} \frac{xlnx^2 + 3x}{\sqrt{4x^2 - 1}}$$

3- What is the growth of the below function: **(What is the most accurate answer?)**

$$f(n) = 8^{logn} + \sqrt{n^6 logn} + nlog^8 n + logn^{2^{logn}}$$

a) $\Theta(n^3)$
b) $\Theta(n^3 logn)$
c) $\Theta(n^3\sqrt{logn})$
d) $\Theta\ (nlogn)$
e) Neither!

4- What is the growth of the below function: **(What is the most accurate answer?)**

$$f(n) = 2^{loglogn} + 3loglog^6 n + 5log^2 n + logn^{10}$$

a) $\Theta(logn)$
b) $\Theta(loglog^6 n)$
c) $\Theta(logn^{10})$
d) $\Theta\ (log^2 n)$
e) Neither!

5- Suppose a machine on average takes 10^{-6} seconds to execute a single algorithm step. When does the machine finish executing the below code when *n* = 100?

```
for(i=0; i < n*n; i++)
  for(k=0; k < i; k++)
          selection_sort(a); //a.length == n
```

6- Assume you want to write a code to calculate the *multiplication* of two numbers. Provide the running time for your algorithm, assuming the inputs are two n-digit numbers. **Explain your answer**.

7- Sort the below numbers using radix sort: (**show the steps**)

14, 10, 32, 50, 1, 54, 12, 45

<u>Part B (</u><u>You need to answer at least **3** questions completely to pass Part B</u>**<u>)</u>**

8- Prove that $f(n) = log^2 n - 6loglogn \quad + logn^4$ is *O(log*^{3}*n),* provide the appropriate *C* and *k* constants.

9- Prove that if $f(n) = \Theta\big(h(n)\big), g(n) = \Theta\big(k(n)\big)$ then $f(n)g(n) = \Theta(h(n)k(n))$

10- Compare the growth of $f(n) = \sqrt{n}log^2 n, g(n) = n^{2+sinn}$.

11- What is the growth of ln1+ 2ln2 + 3ln3 + · · · + n^2lnn^2?

<u>Part A (</u><u>You need to answer at least **2** questions completely to pass Part A</u>**<u>)</u>**

12- Prove that if *f(n)* is monotonically decreasing, then

$$\sum_{i=1}^{n} f(i) = \Omega(\int_{1}^{n} f(x)dx)$$

13- Prove or disprove: if *f(n) = O(g(n))* and *f(n) ≥ 1* and *log(g(n)) ≥ 1* for sufficiently large *n,* then *log(f(n)) = O(log(g(n))).*

14- Prove or disprove:

$$(logn)^{2log^3 n} = \omega((n!)^2)$$

15- Given a sorted array with n integers, provide an algorithm with the running time of O(*logn*) that checks if there is an i for which a[i]= i. (e.g. a = [1 1.5 **2** 5 10 21] >> true because a[2] = 2) (**Explain your answer in details**)