Growth of functions (Ω, O, Θ, o, ω)

Last updated: 2/24/2023

MC = multiple choice

P1 (MC) For all the questions below (except for the True or False questions), the **answer can be none, one, some or all of the choices**. Write your answers on the **LEFT** side. **No justification needed.** (3 points each question)

a) If *f(N) = O(g(N))*, then *f(N) = Θ(g(N))*. True or False.

b) b) Mark all answers that are correct for this summation: 1+2+3 …+i +…+ N
A) Θ(2^N) **B)** Ω(lg(N)*lg(N)) **C)** O(N) **D)** O($N\sqrt{N}$) **E)** none of the these

c) Give a function f(N) (other than N^3) that is $O(N^3)$: f(N) = ……

d) Which of the following is **always** a correct description of the time complexity of the code below (**regardless** of what `someFunction` does)?

A. Θ(N) B. O(N) C. Ω(N) D. O(NlgN)

```
int k;
for(k=1; k <= N; k++) {
   someFunction(k);
}
```

e) You are given the option to choose one of three algorithms with time complexities:

A. Θ(N^2) B. O(N^2) C. Ω(N^2)

You want to choose the algorithm most likely to be the fastest (takes less time). Which one will you choose?

P2.

a) What can you tell about the time complexity of the code below (**regardless** of what `someFunction` does)? Give a lower, upper or tight bound (using **Ω, O,** or **Θ**)**. Justify your answer.**

```
int k;
for(k=1; k <= N; k++) {
     someFunction(N);
}
```

b) What can you tell about the time complexity of the code below (**regardless** of what `someFunction` does)? Give a lower, upper or tight bound (using **Ω, O,** or **Θ**)**. Justify your answer.**

```
int k;
for(k=1; k <= N; k++) {
     return someFunction(100);
}
```

P3. $5N^3 + N^2 = O(N^3)$ True or False? ~~Justify your answer~~.

P4. $5N^3 + N^2 = \Theta(N^3)$ True or False? ~~Justify your answer~~.

P5. $500\lg N = \Theta(N)$ True or False? ~~Justify your answer~~.

P6. Consider the function lgN+300. Select all options below that are also true about this function.

For example if you select $O(N^2)$, you are saying that this function is $O(N^2)$.

a. $O(1)$ b. $O(\lg N)$ c. $O(N)$ d. $O(N2)$

e. $\Theta(1)$ f. $\Theta(\lg N)$ g. $\Theta(N)$ h. $\Theta(N^2)$

i. $\Omega(1)$ j. $\Omega(\lg N)$ k. $\Omega(N)$ l. $\Omega(N^2)$

Extra problems, not part of any examination.

Extra1. Let $T(N) = \sum_{k=0}^{N} \left(\frac{5}{7}\right)^k = \left(\frac{5}{7}\right)^0 + \left(\frac{5}{7}\right)^1 + \left(\frac{5}{7}\right)^2 + \ldots + \left(\frac{5}{7}\right)^N$. To which of the sets below does *T(N)* belong?

A. $\Theta(1)$ B. $\Theta(N)$ C. $\Theta(N^2)$ D. $\Theta(N\lg N)$ E. $\Theta(\lg N)$

Extra2. Given summation: $1 + 2^6 + 3^6 + \ldots + N^6$ Can you solve this in terms of Θ, Ω or O ?

Extra3. **– Hard – for math-lovers.**

Suppose that f(N) > 0 for all N >= 0. Suppose that $g(N) = f(N)/2 + \sqrt{N}$. For each of the following, specify if it is "**definitely true**", "**definitely false**", or "**possibly true and possibly false**". **Justify** your answer (using limits or other properties). If you answer "possibly true and possibly false", provide at least one specific example of f(N) that makes the answer "true" and one specific example of f(N) that makes the answer "false".

a) $f(N) = O(g(N))$

b) $f(N) = \Theta(g(N))$

c) $f(N) = \Omega(g(N))$