COMP 250 : Exercise Set #2 : Feb. 16, 2009 : Complexity

For Questions Contact: `daniel.simeone@gmail.com`

First, we recall the definition of $\mathcal{O}(g(n))$. (This may differ slightly than the one given in class, but it is equivalent.)

Definition:
The function $t(n)$ is $\mathcal{O}(g(n))$ if there exists a positive constant c such that

$$t(n) \leq c \cdot g(n)$$

for sufficiently large values of n.

The phrase " $t(n)$ is $\mathcal{O}(g(n))$ " may be written (and I will do so) as $t(n) \in \mathcal{O}(g(n))$. The symbol "$\in$" is to be read as "is in," and in this context leads to an understanding of $\mathcal{O}(g(n))$ as being the set of all functions that are of the same or lesser complexity than $g(n)$.

Using this notation, we continue with the definitions of $\Omega(g(n))$ and $\Theta(g(n))$.

Definition:
The function $t(n) \in \Omega(g(n))$ if there exists a positive constant c such that

$$c \cdot g(n) \leq t(n)$$

for sufficiently large values of n.

Definition:
The function $t(n) \in \Theta(g(n))$ if there exists positive constants c_1 and c_2 such that

$$c_1 \cdot g(n) \leq t(n) \leq c_2 \cdot g(n)$$

for sufficiently large values of n.

When doing problems, it is key to write the definition down, and to follow carefully from the definition.

Example:
Show that $0.5n$ is $\mathcal{O}(n^2)$.
This would be true if there exists a value of c such that

$$0.5n \leq c \cdot n^2$$

Divide by n.

$$0.5 \leq c \cdot n$$

Since we may take n very large, we may select $c = 1$, giving us

$$0.5 \leq n$$

for very large n, which is clearly true.

In some of the exercises that follow, it may be useful to use techinques that you should remember from calculus – exponentiation, logarithms, and taking limits. Letting n be very large, is essentially taking a limit.

1. Is $\cos(n) \in \mathcal{O}(n)$?

   `ANSWER:` $\cos(n) \leq 1 \leq n$, `so yes.`

2. Is $\cos(n) \in \mathcal{O}(1)$?

   `ANSWER:` $\cos(n) \leq 1$, `so yes.`

3. Is $0.07n + \frac{1}{3000}n^2 \in \mathcal{O}(n^2)$?

4. Is $0.07n + \frac{1}{3000}n^2 \in \Omega(n^2)$?

5. Is $0.07n + \frac{1}{3000}n^2 \in \Theta(n^2)$?

   `ANSWER: FOR 3,4, AND 5.` $\frac{1}{3000}n^2 \leq 0.07n + \frac{1}{3000}n^2 \leq n^2$ `so` $0.07n + \frac{1}{3000}n^2$ `is` $\Theta(n^2)$`, as well as` $\mathcal{O}(n^2)$ `and` $\Omega(n^2)$.

6. Is $n! \in \mathcal{O}((n+2)!)$?

   `ANSWER:` $n! \leq (n+1)(n+1) \cdot n! \Longrightarrow 1 \leq (n+1)(n+1)$ `which is clearly true, so` $n! \in \mathcal{O}((n+2)!)$

7. Is $(n+2)! \in \mathcal{O}(n!)$?

   `ANSWER:` $(n+2)(n+1)n! \leq c \cdot n! \Longrightarrow (n+1)(n+2) \leq c$ `which is clearly false, for any constant` c, `so` $(n+2)! \notin \mathcal{O}(n!)$

8. Is $12^n \in \mathcal{O}(9^n)$?

   `ANSWER:` $12^n \leq c9^n \Longrightarrow n\log 12 \leq \log c + n\log 9 \Longrightarrow \log 12 \leq \frac{\log c}{n} + \log 9 \Longrightarrow \log 12 \leq \log 9$ `FALSE, therefore` $12^n \notin \mathcal{O}(9^n)$

9. Is $9^n \in \mathcal{O}(12^n)$?

   `ANSWER: By similar reasoning as the previous question, one can see that` $9^n \in \mathcal{O}(12^n)$

10. Which of the following is $\mathcal{O}$ of the other:
    $\log n$ and $\log\log n$.
    (Assume that it is the base 2 logarithim.)

    `ANSWER:` $\log\log n \in \mathcal{O}(\log n)$. `This can be seen by simplification using base-2 exponentiation.`

11. If $g(n) \in \Theta(f(n))$ may you conclude that $f(n) \in \Omega(g(n))$.?
    Carefully explain why this is true if it is true, or if it is false, give a counterexample.

    `ANSWER: True.`

12. If $g(n) \in \mathcal{O}(f(n))$ may you conclude that $f(n) \in \Omega(g(n))$.?
    Carefully explain why this is true if it is true, or if it is false, give a counterexample.

    `ANSWER: True.`

13. If $g(n) \in \mathcal{O}(f(n))$ may you conclude that $g(n) \in \Theta(f(n))$.?
    Carefully explain why this is true if it is true, or if it is false, give a counterexample.

    `ANSWER: False. For example,` $n \in \mathcal{O}(n^2)$ `but` $n \notin \Theta(n^2)$.

14. If $f(n)$ is not $\mathcal{O}(g(n))$, then $f(n)$ is $\Omega(g(n))$.
    If this is true, prove it, or if it is false, give a counterexample. `ANSWER: False. For example, let` f `and` g `be Boolean valued functions such that` $f(n) = n$ `iff` n `is even and` $g(n) = n$ `iff` n `is odd, and both functions` 1 `otherwise.`

15. For each algorithm below, indicate the running time using the simplest and most accurate Big- $\mathcal{O}$ notation, as a function of n. Assume that all arithmetic operations can be done in constant time. The first algorithm is an example No justifications are required.

Algorithm	Running time in Big-$\mathcal{O}$Notation

Example(n)
$x \leftarrow 0$
for $i \leftarrow 1$ to n do
    $x \leftarrow x + 1$
	$\mathcal{O}$(n)

Question1(n)
$i \leftarrow n$
while $(i > 0)$ do
    $i \leftarrow \lfloor \frac{i}{2} \rfloor$
	`ANSWER:` $\mathcal{O}(\log n)$

Question2(n)
$x \leftarrow 0$
for $i \leftarrow 1$ to $\sqrt{n}$ do
    for $j \leftarrow 1$ to i do
        $x \leftarrow x + 1$
	`ANSWER:` $\mathcal{O}$`(n)`

Question3(n)
for $i \leftarrow 1$ to 1000 do
    $j \leftarrow 1$
    while $j < n^2$ do
        $j \leftarrow j + 2$
	`ANSWER:` $\mathcal{O}(n^2)$

Question4(n)
If $n \leq 0$ then return 1
else return Question4$(n-1)$
	`ANSWER:` $\mathcal{O}(n)$

BOOTNOTE: Someone in the tutorial asked if I could think of any algorithms that would actually run in $\log \log n$ time. An algorithm running in such a time complexity, since $\log \log n$ is sub-linear, would not even be able to examine all the data even once,
In the last few years, we have begun to have *very* large datasets to deal with. On example is the "web graph," the network structure of the entire Internet. Clearly, it is not efficient to consider the entire Internet with an algorithm, as the data(even of just the link structure, without any content) is too huge. However, one may answer questions about the structure by considering some sub-sampling of the nodes of the Internet that, if chosen in a rational way, can answer algorithmic questions in sub-linear time, often $\log n$. (I still can't think of a precise example of $\log \log n$.)