

# COMP460: Algorithms and Complexity

## Solutions to Homework #1

Jose Luis Rodriguez

January 18, 2017

**1 (HW1).** Rank the following functions by the order of growth. That is, make a list of the functions such that if  $f(n)$  is on a higher line than  $g(n)$ , then  $f(n)$  is  $\Omega(g(n))$ . If  $f(n)$  is  $\Theta(g(n))$ , then  $f(n)$  and  $g(n)$  should be listed on the same line.

*Solution.* We can use L'Hopital's rule to find what function grows faster, then rank them by order of growth.

- |                         |                                  |                   |
|-------------------------|----------------------------------|-------------------|
| 1. 1 and 2              | 7. $n = 2^{\lg n}$               | 13. $n!$          |
| 2. $\ln(\ln n)$         | 8. $n \lg(n)$                    |                   |
| 3. $\sqrt{\lg n}$       | 9. $n^2 = 4^{\lg n}$             | 14. $(n+1)!$      |
| 4. $\ln(n)$ and $\lg n$ | 10. $n^3$                        | 15. $2^{2^n}$     |
| 5. $\lg^2(n)$           | 11. $\left(\frac{3}{2}\right)^n$ |                   |
| 6. $\sqrt{2}^{\lg n}$   | 12. $2^n$                        | 16. $2^{2^{n+1}}$ |

□

**2 (HW1).** Express the solution to each of the following recurrences using  $\Theta$ -notation. Show your reasoning.

We can solve the recurrences below using the Master Theorem, given that:  $T(n) = aT(n/b) + f(n)$ ,  $a \geq 1, b > 1$ . Following three cases:

1. If  $\exists \epsilon > 0$  such that  $f(n) = \mathcal{O}(n^{\log_b a - \epsilon})$ , then  $T(n) = \mathcal{O}(n^{\log_b a})$ .
2. If  $f(n) = \Theta(n^{\log_b a} \lg^k n)$ , with  $k \geq 0$ , then  $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$ .
3. If  $\exists \epsilon > 0$  such that  $f(n) = \Omega(n^{\log_b(a+\epsilon)})$ , then  $T(n) = \Theta(f(n))$ .

(a)  $T(n) = 2T(n/2) + nlg^2n$ .

*Solution.* Following a more general form of the masters theorem where  $T(n) = aT(n/b) + \Theta(n^k \log^\phi(n))$ ,  $a \geq 1$ ,  $b > 1$ , and  $\phi \in \mathbb{R}$ . We have that  $a = 2, b = 2, k = 1$  and  $f(n) = n^{\log_2 2} lg^2 n$ . For this case if  $a = b^k$  and  $\phi > -1$  then the solution to the recurrence follows case#2 of the general form, such that:  $T(n) = \Theta(n^{\log_b a} lg^{\phi+1} n) = \Theta(n^1 lg^3 n)$ .  $\square$

(b)  $T(n) = 9T(n/2) + n^3$ .

*Solution.* Given that  $a = 9$ ,  $b = 2$ ,  $f(n) = n^3$ . Since  $n^{\log_2 9}$  is polynomially larger than  $f(n) = n^3$  this recurrence follows case# 1 of the masters theorem. Such that  $f(n) = \mathcal{O}(n^{\log_b a - \epsilon})$ . Therefore the solution to the recurrence is given by:  $T(n) = \Theta(n^{lg 9})$ .  $\square$

(c)  $T(n) = 8T(\lceil n/4 \rceil) + n^2$ .

*Solution.* When solving recurrences usually we can disregard floor/ceil as we are interested in the growth of the recurrence (running time). In this case we have that  $a = 8$ ,  $b = 4$ ,  $f(n) = n^2$  and  $n^{\log_4 8} = n^{\frac{3}{2}}$ . Since  $n^{\log_4 8} \leq n^2$  the recurrence follows case# 3 of the master theorem. Such that  $f(n) = \Omega(n^{\log_b(a+\epsilon)}) = \Omega(n^2)$   $\square$