CS 260: Homework 2

Daniel Lopez

July 14, 2017

8 July 2017

1 1

1.1 g1(n) and g3(n)

$g1(n)$ is $O(g3(n))$ for all even n >= 0.
$g1(n)$ is $\Omega(g3(n))$ for all odd n.
$g3(n)$ is $O(g1(n))$ for odd n >= 1.
$g1(n)$ is $\Omega(g3(n))$ for all even n.

1.2 g1(n) and g2(n)

$g1(n)$ is $O(g2(n))$ for all even and odd n for n > 100.
$g1(n)$ is $\Omega(g2(n))$ for all n except for even n for n > 100.
$g2(n)$ is $O(g1(n))$ for all n except for n > 100
$g2(n)$ is $\Omega(g1(n))$ for all n except for n < 100.

1.3 g2(n) and g3(n)

$g2(n)$ is $O(g3(n))$ for all n 0 <= n <= 100.
$g2(n)$ is $\Omega(g3(n))$ for n n > 100.
$g2(n)$ is $O(g2(n))$ for n > 100.
$g3(n)$ is $\Omega(g2(n))$ for all n < 100.

2 2

2.1 a

Since 17 is a constant, it is not dependent on any variable. For one input, there is one output: 17. With no other dependencies, 17 is $O(1)$.

2.2 b

By definition of $T(n)$ is $O(f(n))$: $T(n) = \frac{n(n-1)}{2}$

$$2 * \frac{n(n-1)}{2} <= 2 * \frac{1}{2} n^2 n(n-1) <= \quad n^2 \frac{n^2 - n}{n} <= \frac{n^2}{n} n - 1 <= \quad n \tag{1}$$

Thus, $\frac{n(n-1)}{2}$ is $O(n^2)$

2.3 c

By definition of $T(n)$ is $O(n^3)$ $T(n) = \max(n^3,\ 10(n^2))$ $n = 11$

$$\begin{aligned} max(11^3, 10(11^2) &<= 11^3 && (2)\\ max(1331, 1210) &<= 1331 && (3)\\ 1331 &<= 1331 && (4)\\ & && (5) \end{aligned}$$

Thus, $\max(n^3,\ 10(n^2))$ is $O(n^3)$ for values of $n >= 10$.

2.4 d

By summing, we can prove that $\sum_{i=1}^{n} i^k$ is $O(n^{k+1}$ and $\Omega(n^{k+1})$. $1^k + 2^k + ... + n^{k+1} <= n^{k+1}$ $1^k + 2^k + ... + n^{k+1} >= n^{k+1}$ Since both of these fit the definition of equal, $\sum_{i=1}^{n} i^k$ is $O(n^{k+1}$ and $\Omega(n^{k+1})$, or $\Theta(n^{k+1})$.

2.5 e

If a is a positive integer, $a_k n^k + a_{k-1} n^{k-1} + ... + a_0$. Clearly, $P(x) <= a_k n^k + a_{k-1} n^{k-1} + ... + a_0$, so $P(x)$ is $O(n^k)$. Also, since $a_k >= 0$, $P(x)$ is $\Omega(n^k)$

3 3

$\frac{1}{3}^n < 17 < log(log(n)) < log(n) < log^2(n) < \sqrt{n} < \sqrt{n} log^2(n) < \frac{n}{log(n)} < n < \frac{2}{3}^n$

4 4

The max function is called twice and is of size n/2. This means we can use the master theorem: $T(j) = 2T(\frac{n}{2}) + 1$ Solving using the master theorem:

$$a = 2b = 2c = log_2(n) = 11 <= n \tag{6}$$

The max function satisfies case 1, the definition of which states that $2T(\frac{n}{2}) + 1$ is $O(n)$ and $\Omega(n)$.

5 5

The delete function doesn't work from line 4, where it should read

while p $<$ END(L)

Additionally, the function could immediately check if the array is empty with

if not L:

return L

else

delete ..."

6 6

FIRST: n^2
END: n^2
NEXT: n^3