CS5300 Advanced Algorithms
HW #2
Kishor Kumar Andekar
700744713 (CRN: 12008)

1. Rank the following functions by order of growth.

$\lg(\lg n), n^2, e^n, n^3, n, \lg(n), 2^n, n(\lg(n)), 1$

Ranking of Functions by Order of growth from Smallest to Largest

Constant < Logarithmic < Linear < Linearithmic < Polynomials < Exponential

Therefore, Order of increasing growth rate of the given functions:

$$1 < \lg(\lg n) < \lg(n) < n < n\lg n < n^2 < n^3 < 2^n < e^n$$

2. Show that for any real constants a and b, where b>0,

$$\underbrace{(n+a)^b}_{f(n)} = O(\underbrace{n^b}_{g(n)})$$

Defination: we want to find c and n_0 $\ni$ $0 \le f(n) \le c g(n)$ $\forall$ $n \geq n_0$

$$0 \le (n+a)^b \le \overset{?}{c}(n^b) \quad \forall \; n \geq \overset{?}{n_0}$$

$$n+a \le n+n \quad \text{if } n \geq |a|$$

$$n+a \le 2n$$

$$(n+a)^b \le (2n)^b$$

$$(n+a)^b \le 2^b n^b$$

Let $c = 2^b$ and $n_0 = |a|$

3. What does the following algorithm do (Output)? Analyze its worst Case running time, and express it using "Big-oh" notation.

Algorithm Foo (a, n):

Input : two integers a and n

Output : ?

	Cost	time
K=0	C1	1
b=1	C2	1
while K<n do	C3	n+1
K=K+1	C4	n
b=b*a	C5	n
return b	C6	1

Let Assume

$a=2, n=3 \Rightarrow b=8 \Rightarrow 2^3=a^n$

$a=3, n=4 \Rightarrow b=81 \Rightarrow 3^4=a^n$

$a=4, n=5 \Rightarrow b=1024 \Rightarrow 4^5=a^n$

Worst Case:

$$T(n) = C_1(1) + C_2(1) + C_3(n+1) + C_4(n) + C_5(n) + C_6(1)$$

$$= C_1 + C_2 + C_3 n + C_3 + C_4 n + C_5 n + C_6$$

$$= (C_1 + C_2 + C_3 + C_6) + n(C_3 + C_4 + C_5)$$

$$= a + bn$$ // Where a and b are constants

Hence $T(n) = O(n)$

4. What does the following algorithm do (what is the output) ? Analyze its worst - Case running time and express it using "Big-oh" notation

Algorithm Foo (A, n)

Input : An Array A Storing $n \geq 1$ integers

Output : ??	Cost	time
K = A[0]	C_1	1
for i=1 to n-1 do	C_2	n
if K > A[i] then	C_3	n-1
K = A[i]	C_4	n-1
return K	C_5	1

Assume A = [1, 2, 3]

0	1	2
1	2	3

$K = A[0] = 1$

$K > A[i] \Rightarrow 1 > A[1] \Rightarrow 1 > 2$ # Fail

$K > A[i] \Rightarrow 1 > A[2] \Rightarrow 1 > 3$ # Fail

K=1 is the smallest value of A

A=[3,6,7] ⇒ K=3 is the smallest value of A

A=[7,8,9] ⇒ K=7 is the smallest value of A

Worst Case: $T(n) = C_1(1) + C_2(n) + C_3(n-1) + C_4(n-1) + C_5(1)$

$= C_1 + C_2 n + C_3 n - C_3 + C_4 n - C_4 + C_5$

$= (C_1 - C_3 - C_4 + C_5) + n(C_2 + C_3 + C_4)$

$T(n) = a + bn$ // where a and b are Constants

$T(n) = O(n)$

5. Determine whether each statement is true or false. Justify your answer

a) $n^n = O(2^n)$

b) If $f(n) = O(g(n))$, then $g(n) = \Omega(f(n))$

c) $2^{n+1} = O(2^n)$

a) $\underbrace{n^n}_{f(n)} = O(\underbrace{2^n}_{g(n)})$

We want to find c and $n_0 \ni 0 \le f(n) \le c\,g(n) \;\forall\; n \ge n_0$

$0 \le n^n \le c \cdot 2^n \;\forall\; n \ge n_0$

Apply log on both sides

$0 \le \lg n^n \le c \lg 2^n$

$0 \le n \lg n \le c\, n \lg_2^2$ // $\lg_2^2 = 1$

$0 \le n \lg n \le cn$

Therefore, $n \lg n$ is always greater than n

Hence $n^n \ne O(2^n)$. It is False

b) If $f(n) = O(g(n))$, then $g(n) = \Omega(f(n))$

$f(n) = a$, $g(n) = b$

According to Big oh $0 \le f(n) \le c\,g(n)$

$a \le b$

$g(n) = \Omega(f(n))$

According to Big Ω $0 \le c\,g(n) \le f(n)$

$a \le b$

According to Transpose Symmetry $b \ge a$ if $a \le b$, it is True

c. $\underset{f(n)}{2^{n+1}} = O(\underset{g(n)}{2^n})$

We want to find C and n_0 $\ni$ $0 \leq f(n) \leq Cg(n)$ $\forall$ $n \geq n_0$

$$0 \leq 2^{n+1} \leq C2^n \quad \forall \; n \geq n_0$$

$$f(n) = 2^{n+1} \; , \; g(n) = 2^n$$

If $n_0 = 0$, $0 \leq 2^{0+1} \leq C \cdot 2^0$

$$\Rightarrow 0 \leq 2 \leq C$$

If $n_0 = 1$, $0 \leq 2^{1+1} \leq C \cdot 2^1$

$C = 2$

$$\Rightarrow 0 \leq 2^2 \leq C2$$

$$\Rightarrow 0 \leq 4 \leq 2(2)$$

$$\Rightarrow 0 \leq 4 \leq 4$$

Hence $2^{n+1} = O(2^n)$. It is true

6 Show that $\underset{f(n)}{\lg n} = O(\underset{g(n)}{\sqrt{n}})$

Defination: we want to find c and $n_0 \ni$

$$0 \leq f(n) \leq c\,g(n) \quad \forall \; n \geq n_0$$

$$0 \leq \lg n \leq \overset{?}{c} \sqrt{n} \quad \forall \; n \geq \overset{?}{n_0}$$

Let $c=1$, $n_0=16$

n_0	$\lg n$	$\sqrt{n}$
1	0	1
2	1	1.4
4	2	2
8	3	2.6
16	4	4

// check $\sqrt{n}$ bigger than $\lg n$

7. Show that $\underset{f(n)}{2n^2 - 3n}$ is $\Omega(\underset{g(n)}{n^2})$

Def: We want to find c and $n_0 \ni \; 0 \leq c\,g(n) \leq f(n) \quad \forall \; n \geq n_0$

$$0 \leq \overset{?}{c}\, n^2 \leq 2n^2 - 3n \quad \forall \; n \geq \overset{?}{n_0}$$

Let $c = \frac{1}{5}$, $n_0 = 2$

If $n_0 = 2$, then $\frac{1}{5} 2^2 \leq 2 \cdot 2^2 - 3 \cdot 2 \Rightarrow \frac{4}{5} \leq 2$

8. Show that $\underset{f(n)}{n^2 + 3n} = O(\underset{g(n)}{n^2})$

Def: We Want to find c and $n_0 \ni$

$$0 \leq f(n) \leq c\,g(n) \quad \forall \; n \geq n_0$$

$$0 \leq n^2 + 3n \leq \overset{?}{c}\, n^2 \quad \forall \; n \geq \overset{?}{n_0}$$

Let $c=4$, $n_0=1$

9. Show that $\underset{f(n)}{10n^2 - 5n + 51} = o(\underset{g(n)}{n^3})$

Def: $\lim\limits_{n \to \infty} \dfrac{f(n)}{g(n)}$

$$= \lim_{n \to \infty} \frac{10n^2 - 5n + 51}{n^3}$$

$$= \lim_{n \to \infty} \left(\frac{10n^2}{n^3} - \frac{5n}{n^3} + \frac{51}{n^3} \right)$$

$$= \lim_{n \to \infty} \frac{10n^2}{n^3} - \lim_{n \to \infty} \frac{5n}{n^3} + \lim_{n \to \infty} \frac{51}{n^3}$$

$$= \lim_{n \to \infty} \frac{10}{n} - \lim_{n \to \infty} \frac{5}{n^2} + \lim_{n \to \infty} \frac{51}{n^3}$$

$$= 0 - 0 + 0 = 0$$

Hence $10n^2 - 5n + 51 = o(n^3)$

10. Show that $\underset{f(n)}{18n^3 - 4n^2 + 2n - 40} = \omega(\underset{g(n)}{n^2})$

Def: $\lim\limits_{n \to \infty} \dfrac{f(n)}{g(n)} = \lim\limits_{n \to \infty} \dfrac{18n^3 - 4n^2 + 2n - 40}{n^2}$

$$= \lim_{n \to \infty} \left(\frac{18n^3}{n^2} - \frac{4n^2}{n^2} + \frac{2n}{n^2} - \frac{40}{n^2} \right)$$

$$= \lim_{n \to \infty} 18n - \lim_{n \to \infty} 4 + \lim_{n \to \infty} \frac{2}{n} - \lim_{n \to \infty} \frac{40}{n^2}$$

$$= \infty - 4 + 0 - 0 = \infty$$

Hence $18n^3 - 4n^2 + 2n - 40 = \omega(n^2)$