Big-O and other asymptotic notations

Question 1

Let $a(n) = 10^6 n^2$ and $b(n) = 10^n$. Computer A performs 10^6 operations per second; computer B performs 10^{12} operations per second. In the worst case on an instance of size n, an implementation of an algorithm α solves a problem P in $a(n)$ operations on computer A, and an implementation of an algorithm β solves P in $b(n)$ operations on computer B.

(a) Which instances of P would you solve using the implementation of α on A, and which using the implementation of β on B?

(b) Estimate how long it would take in the worst case to solve an instance of P of size 30 using α on A and using β on B.

Solution:

To determine which instances of the problem P you would solve using the implementations of algorithms α on computer A and β on computer B, we need to analyze the time complexity of each algorithm relative to the computing power of the respective computers.

Computers' performance:

- Computer A: 10^6 operations per second.
- Computer B: 10^{12} operations per second.

Time taken by each implementation:

- Implementation of α on computer A requires $a(n) = 10^6 n^2$ operations.
- Implementation of β on computer B requires $b(n) = 10^n$ operations.

Time taken to execute on each computer:

- For computer A:

$$\text{Time for } \alpha = \frac{a(n)}{10^6} = \frac{10^6 n^2}{10^6} = n^2 \text{ seconds}$$

- For computer B:

$$\text{Time for } \beta = \frac{b(n)}{10^{12}} = \frac{10^n}{10^{12}} = 10^{n-12} \text{ seconds}$$

Determining when to use each implementation: We want to find the values of n for which:

1. $n^2 < 10^{n-12}$ (favorable to use α on A).
2. $n^2 \geq 10^{n-12}$ (favorable to use β on B).

To find the transition point, we set up the inequality:

$$n^2 < 10^{n-12}$$

Taking the logarithm (base 10) of both sides:

$$\log_{10}(n^2) < n - 12$$

$$2\log_{10}(n) < n - 12$$

Now, rearranging gives:

$$n - 2\log_{10}(n) > 12$$

This inequality can be evaluated for different values of n:

1. **Trial with small n:**
   - For $n = 1$:

     $$1 - 2 \cdot 0 = 1 \quad (\text{not } > 12)$$

   - For $n = 2$:

     $$2 - 2 \cdot 0.301 = 1.398 \quad (\text{not } > 12)$$

   - For $n = 3$:

     $$3 - 2 \cdot 0.477 = 2.046 \quad (\text{not } > 12)$$

   - For $n = 10$:

     $$10 - 2 \cdot 1 = 8 \quad (\text{not } > 12)$$

   - For $n = 20$:

     $$20 - 2 \cdot 1.301 = 17.398 \quad (> 12)$$

2- Finding the exact the transition point n=15. Use computer A when n>=15 and computer B when n<15.

b.

For Algorithm α on Computer A:

1. **Operations required**:

$$a(30) = 10^6 \cdot 30^2 = 10^6 \cdot 900 = 9 \times 10^8 \text{ operations}$$

2. **Time taken**: Computer A performs 10^6 operations per second, so:

$$\text{Time for } \alpha = \frac{a(30)}{10^6} = \frac{9 \times 10^8}{10^6} = 900 \text{ seconds}$$

For Algorithm β on Computer B:

1. **Operations required**:

$$b(30) = 10^{30} \text{ operations}$$

2. **Time taken**: Computer B performs 10^{12} operations per second, so:

$$\text{Time for } \beta = \frac{b(30)}{10^{12}} = \frac{10^{30}}{10^{12}} = 10^{18} \text{ seconds}$$

Summary of Results:

- **Time to solve using α on A**: 900 seconds (or 15 minutes).
- **Time to solve using β on B**: 10^{18} seconds, which is approximately 31.7 billion years.

Thus, in the worst case:

- Using α on computer A will take **900 seconds**.
- Using β on computer B will take **approximately 10^{18} seconds**, which is impractically long.

Question 2

- Suppose that k is a positive integer. Show that if $f = O(n^k)$ then there are constants a, $b > 0$ such that $f(n) \leq an^k + b$ for all $n \geq 0$.

Solution:

If $f = O(n^k)$ then there are $c > 0$ and n_0 such that for all $n \geq n_0$ we have $f(n) \leq cn^k$. Take $a = c$ and $b = 1 + \max\{ f(n) : n < n_0 \}$ where $\max \emptyset = 0$, then $f(n) \leq an^k + b$ for all $n \geq 0$.

Question 3

Give yes/no answers to the following:

	$f(n)$	$g(n)$	$f = O(g)$?	$f = \Omega(g)$?	$f = \Theta(g)$?
$a.$	$n - 100$	$n - 200$			
$b.$	$n^{1/2}$	$n^{2/3}$			
$c.$	$100n + \log n$	$n + (\log n)^2$			
$d.$	$n \log n$	$10n \log 10n$			
$e.$	$\log 2n$	$\log 3n$			
$f.$	$n^{0.1}$	$(\log n)^{10}$			
$g.$	$\sqrt{n}$	$(\log n)^3$			
$h.$	$n2^n$	3^n			
$i.$	2^n	2^{n+1}			
$j.$	$(\log n)^{\log n}$	$2^{(\log n)^2}$			

Solution: we use rules in the course

a. Yes, Yes, Yes
b. Yes, No, No
c. Yes, Yes, Yes,
d. Yes, Yes, Yes
e. Yes, Yes, Yes
f. No, Yes, No
g. No, Yes, No
h. Yes, No, No,
i. Yes, Yes, Yes,
j. Yes, No, No,

Exercise 4. *Relative asymptotic growths*

Determine weather A is O(B), Ω(B) or Θ(B)

A	B
$\lg^k n$	n^ϵ
n^k	c^n
$\sqrt{n}$	$n^{\sin n}$
2^n	$2^{n/2}$
$n^{\log c}$	$c^{\log n}$
$\log(n!)$	$\log(n^n)$

Solution:

A	B	O	o	Ω	ω	Θ
$\lg^k n$	n^ϵ	yes	yes	no	no	no
n^k	c^n	yes	yes	no	no	no
$\sqrt{n}$	$n^{\sin n}$	no	no	no	no	no
2^n	$2^{n/2}$	no	no	yes	yes	no
$n^{\log c}$	$c^{\log n}$	yes	no	yes	no	yes
$\log(n!)$	$\log(n^n)$	yes	no	yes	no	yes

Question 5

Show that $\log(n!) = \Theta(n \log n)$.

Solution:

Another approach to show $\log(n!) = \mathcal{O}(n\log(n))$

$$\begin{aligned} n! &= (n-0)(n-1)(n-2)...(n-(n-1)) \\ &= n(1-\frac{0}{n})\cdot n(1-\frac{1}{n})\cdot n(1-\frac{2}{n})\cdot ...\cdot n(1-\frac{n-1}{n}) \\ &= n^n\cdot(1-\frac{0}{n})\cdot(1-\frac{1}{n})\cdot(1-\frac{2}{n})\cdot ...\cdot(1-\frac{n-1}{n}) \\ &= n^n\prod_{k=0}^{n-1}(1-\frac{k}{n}) \end{aligned}$$

$$\begin{aligned} \log(n!) &= \log(n^n\prod_{k=0}^{n-1}(1-\frac{k}{n})) \\ &= \log(n^n)+\log(\prod_{k=0}^{n-1}(1-\frac{k}{n})) \\ &= n\log(n)+\log(\prod_{k=0}^{n-1}(1-\frac{k}{n})) \\ &= \mathcal{O}(n\log(n)) \end{aligned}$$

Exercise 6

Asymptotically rank the following functions:

n, n1/2, log(n), log(log(n)), log2(n), $\frac{1}{3}^n$, 4, $\frac{3}{2}^n$, n!

Solution:

$\frac{1}{3}^n$ < 4 < log(log(n)) < log(n) < log2(n) < √(n) < n <$\frac{3}{2}^n$ < n!

Exercise 7.

For every given $f(n)$ and $g(n)$ prove that $f(n) = \Theta(g(n))$

a) $g(n) = n^3$, $f(n) = 3n^3 + n^2 + n$

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

c) $g(n) = \ln(n)$, $f(n) = \log_{10}(n) + \log_{10}(\log_{10} n)$

Solution:

For all the given $f(n)$ and $g(n)$ we can prove that $f(n) = \Theta(g(n))$ using the limit test and/or proving the following statement:

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \, \forall n \geq n_0 \tag{1}$$

a) $g(n) = n^3$, $f(n) = 3n^3 + n^2 + n$

Solution 1 Using equation 1 we get

$$\begin{array}{rccccccccl} & 0 & \leq & c_1(g(n)) & \leq & f(n) & \leq & c_2(g(n)) & \forall n \geq n_0 \\ = & 0 & \leq & c_1 n^3 & \leq & 3n^3 + n^2 + n & \leq & c_2 n^3 & \forall n \geq n_0 \\ & & & & & \textit{dividing by } n^3 & & & \\ = & 0 & \leq & c_1 & \leq & 3 + \frac{1}{n} + \frac{1}{n^2} & \leq & c_2 & \forall n \geq n_0 \end{array}$$

Choosing $c_1 = 3$, $c_2 = 5$, and $n_0 = 1$ helps us in proving the relations of the equation

$$\begin{array}{rcccccccc} = & 0 & \leq & 3 & \leq & 3+1+1 & \leq & 5 \\ = & 0 & \leq & 3 & \leq & 5 & \leq & 5 \end{array}$$

Therefore, $f(n) = \Theta(g(n))$

Solution 2

$$\begin{aligned} \lim_{n \to \infty} \frac{f(n)}{g(n)} &= \frac{3n^3+n^2+n}{n^3} \\ &= 3 + \frac{1}{n} + \frac{1}{n^2} \end{aligned}$$

$$= 3 + 0 + 0$$

$$= 3 \in R^+$$

Therefore, $f(n) = \Theta(g(n))$

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

Solution 1 Using equation 1 we get

$$\begin{array}{lllllllll} & 0 & \leq & c_1(g(n)) & \leq & f(n) & \leq & c_2(g(n)) & \forall n \geq n_0 \\ = & 0 & \leq & c_1 2^n & \leq & 2^{n+1} & \leq & c_2 2^n & \forall n \geq n_0 \end{array}$$

$$\textit{dividing by } 2^n$$

$$\begin{array}{lllllllll} = & 0 & \leq & c_1 & \leq & 2 & \leq & c_2 & \forall n \geq n_0 \end{array}$$

Choosing $c_1 = 2$, $c_2 = 2$, and $n_0 = 1$ helps us in proving the relations of the equation

$$= 0 \leq 2 \leq 2 \leq 2$$

Therefore, $f(n) = \Theta(g(n))$.

Solution 2

$$\begin{aligned} \lim_{n\to\infty} \frac{f(n)}{g(n)} &= \frac{2^{n+1}}{2^n} \\ &= \frac{2.2^n}{2^n} \\ &= 2 \in R^+ \end{aligned}$$

c) $g(n) = \ln(n)$, $f(n) = \log_{10}(n) + \log_{10}(\log_{10} n)$

Solution 1

$$\begin{aligned} \lim_{n\to\infty} \frac{f(n)}{g(n)} &= \frac{\log_{10}(n) + \log_{10}(\log_{10}(n))}{\ln(n)} \\ &= \frac{\infty}{\infty} 1 \end{aligned}$$

$$Using\ L'H\hat{o}pital's\ rule$$

$$\begin{aligned} \lim_{n\to\infty} \frac{f'(n)}{g'(n)} &= \frac{\frac{1}{n\ln(10)}}{\frac{1}{n}} + \frac{\frac{1}{\log_{10}(n)\ln(10)} \cdot \frac{1}{n\ln(10)}}{\frac{1}{n}} \\ &= \frac{n}{n\ln(10)} + \frac{n}{\ln^2(10) n \log_{10}(n)} \\ &= \frac{1}{\ln(10)} + \frac{1}{\ln^2(10)\log_{10}(n)} \\ &= 0.434(3d.p) + 0 \in R^+ \end{aligned}$$

Therefore, $f(n) = \Theta(g(n))$

Solution 2

$$\begin{array}{lllllllll} & 0 & \leq & c_1(g(n)) & \leq & f(n) & \leq & c_2(g(n)) & \forall n \geq n_0 \\ = & 0 & \leq & c_1 \ln(n) & \leq & \log_{10}(n) + \log_{10}(\log_{10}(n)) & \leq & c_2 \ln(n) & \forall n \geq n_0 \end{array}$$

$$\textit{dividing by } \ln(n)$$

$$= \quad 0 \quad \leq \quad c_1 \qquad \leq \quad \log_{10}(e) + \frac{\log_{10}(\log_{10}(n))}{\ln(n)} \quad \leq \quad c_2 \qquad \forall n \geq n_0$$

Choosing $c_1 = 0.434, c_2 = 0.5,$ and $n_0 = 10$ helps us in proving the relations of the equation

$$= \quad 0 \quad \leq \quad 0.434 \quad \leq \quad 0.434 \quad \leq \quad 0.5$$

Therefore, $f(n) = \Theta(g(n))$

Exercise 8. *Asymptotic notation properties*

Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove or disprove each of the following conjectures.

a. $f(n) = O(g(n))$ implies $g(n) = O(f(n))$.

b. $f(n) + g(n) = \Theta(\min\{f(n), g(n)\})$.

c. $f(n) = O(g(n))$ implies $\lg f(n) = O(\lg g(n))$, where $\lg g(n) \geq 1$ and $f(n) \geq 1$ for all sufficiently large n.

d. $f(n) = O(g(n))$ implies $2^{f(n)} = O\left(2^{g(n)}\right)$.

e. $f(n) = O\left((f(n))^2\right)$.

f. $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$.

g. $f(n) = \Theta(f(n/2))$.

h. $f(n) + o(f(n)) = \Theta(f(n))$.

Solution:

a. False. Counterexample: $n = O(n^2)$ but $n^2 \neq O(n)$.

b. False. Counterexample: $n + n^2 \neq \Theta(n)$.

c. True. Since $f(n) = O(g(n))$ there exist c and n_0 such that $n \geq n_0$ implies $f(n) \leq cg(n)$ and $f(n) \geq 1$. This means that $\log(f(n)) \leq \log(cg(n)) = \log(c) + \log(g(n))$. Note that the inequality is preserved after taking logs because $f(n) \geq 1$. Now we need to find d such that $\log(f(n)) \leq d \log(g(n))$. It will suffice to make $\log(c) + \log(g(n)) \leq d \log(g(n))$, which is achieved by taking $d = \log(c) + 1$, since $\log(g(n)) \geq 1$.

d. False. Counterexample: $2n = O(n)$ but $2^{2n} \neq 2^n$ as shown in exercise 3.1-4.

e. False. Counterexample: Let $f(n) = \frac{1}{n}$. Suppose that c is such that $\frac{1}{n} \leq c\frac{1}{n^2}$ for $n \geq n_0$. Choose k such that $kc \geq n_0$ and $k > 1$. Then this implies $\frac{1}{kc} \leq \frac{c}{k^2c^2} = \frac{1}{k^2c}$, a contradiction.

f. True. Since $f(n) = O(g(n))$ there exist c and n_0 such that $n \geq n_0$ implies $f(n) \leq cg(n)$. Thus $g(n) \geq \frac{1}{c} f(n)$, so $g(n) = \Omega(f(n))$.

g. False. Counterexample: Let $f(n) = 2^{2n}$. By exercise 3.1-4, $2^{2n} \neq O(2^n)$.

h. True. Let g be any function such that $g(n) = o(f(n))$. Since g is asymptotically positive let n_0 be such that $n \geq n_0$ implies $g(n) \geq 0$. Then $f(n) + g(n) \geq f(n)$ so $f(n) + o(f(n)) = \Omega(f(n))$. Next, choose n_1 such that $n \geq n_1$ implies $g(n) \leq f(n)$. Then $f(n) + g(n) \leq f(n) + f(n) = 2f(n)$ so $f(n) + o(f(n)) = O(f(n))$. By Theorem 3.1, this implies $f(n) + o(f(n)) = \Theta(f(n))$.