Author: Parth Shah

Title: Chapter 3 Solutions

Notes

Θ-Notation Asymptotic Tight Bound $\Theta(g(n)) = \{f(n)$: there exist positive constants c_1, c_2, and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0\}$

O-Notation Asymptotic Upper Bound $O(g(n)) = \{f(n)$:there exist positive constants c and n_0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0\}$

Ω-Notation Asymptotic Lower Bound $\Omega(g(n)) = \{f(n)$: there exist positive constants c and n_0 such that $0 \leq cg(n) \leq f(n)$ for all $n \geq n_0\}$

o-Notation Not Tight Asymptotic Upper Bound $O(g(n)) = \{f(n)$: for any positive constant c there exists a n_0 such that $0 \leq f(n) < cg(n)$ for all $n \geq n_0\}$

ω-Notation Not Tight Asymptotic Lower Bound $\omega(g(n)) = \{f(n)$: for any positive constant c there exists a n_0 such that $0 \leq cg(n) < f(n)$ for all $n \geq n_0\}$

3.1 Growth of Functions

Problem 3.1-1 Let $f(n)$ and $g(n)$ be asymptotically nonnegative functions. Using basic definition of Θ-notation, prove that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

Solution 3.1-1 WLOG assume $f(n) < g(n)$. Therefore $\max(f(n), g(n)) = g(n)$. Since $f(n) < g(n)$ we also have $\frac{1}{2}f(n) + \frac{1}{2}g(n) < g(n) < f(n) + g(n) \rightarrow \max(f(n), g(n)) = \Theta(f(n) + g(n))$.

Problem 3.1-2 Show that for any real constant a and b, where $b > 0$, $(n + a)^b = \Theta(n^b)$

Solution 3.1-2 We can use the definition of Θ notation to prove this. Set constants $c_1 = 1$ and $c_2 = 2$. We see that for all n, $(n + a)^b > c_1 n^b$. Furthermore for all $n > a$, $(n + a)^b < c_2 n^b$. Therefore $(n + a)^b = \Theta(n^b)$.

Problem 3.1-3 Explain why saying an algorithm is at least $O(f(n))$ is meaningless.

Solution 3.1-3 Big O means that an algorithm is upper bounded by function $f(n)$. However, saying it is at least $f(n)$ means the upper bound can be any function greater than $f(n)$. Therefore the upper bound does not exist and the algorithm can be of any runtime.

Problem 3.1-4 Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$.

Solution 3.1-4 $2^{n+1} = O(2^n)$ is true because $2^{n+1} = 2 \cdot 2^n$ and 2 is a constant. $2^{2n} = O(2^n)$ is not true because $2^{2n} = 2^n \cdot 2^n$ and 2^n is not a constant.

Problem 3.1-5 Prove that $f(n) = \Theta(g(n))$ iff $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

Solution 3.1-5 $f(n) = O(g(n)) \to \exists c_1$ such that $0 \le f(n) \le c_1 g(n)$. $f(n) = \Omega(g(n)) \to \exists c_2$ such that $0 \le c_2 g(n) \le f(n)$. Combining these we get $0 \le c_2 g(n) \le f(n) \le c_1 g(n)$.

Problem 3.1-6 Prove that an algorithm is $\Theta(g(n))$ iff worst case is $O(g(n))$ and best case is $\Omega(g(n))$

Solution 3.1-6 This is straight from the definition. If the best case is lower bounded by $g(n)$ and worst case is upper bounded by $g(n)$ the entire algorithm is tight bounded by $g(n)$.

Problem 3.1-7 Prove that $o(g(n)) \cap \omega(g(n))$ is the empty set.

Solution 3.1-7 Proof by contradiction. Assume the set is not empty. Then there exists some function $f(n)$ such that $f(n) = o(g(n) = \omega(g(n))$. This implies that there exists a constant n_1 and n_2 for a given constant c such that $f(n) > cg(n)$ for $n \ge n_1$ and $f(n) < cg(n)$ for $n \le n_2$. This is a clear contradiction. Therefore the intersection is empty.

Problem 3.1-8 O-Notation Asymptotic Upper Bound $O(g(n,m)) = \{f(n,m)$: there exist positive constants c, n_0, and m_0 such that $0 \le f(n,m) \le cg(n,m)$ for all $n \ge n_0$ or $m \ge m_0\}$

Solution 3.1-8 Θ-Notation Asymptotic Tight Bound $\Theta(g(n,m)) = \{f(n,m)$: there exist positive constants c_1, c_2, n_0, and m_0 such that $0 \le c_1 g(n,m) \le f(n,m) \le c_2 g(n,m)$ for all $n \ge n_0$ or $m \ge m_0\}$

Ω-Notation Asymptotic Lower Bound $\Omega(g(n,m)) = \{f(n,m)$: there exist positive constants c, n_0, and m_0 such that $0 \le cg(n,m) \le f(n)$ for all $n \ge n_0$ or $m \ge m_0\}$

3.2 Standard Notations and Common Functions

Problem 3.2-1 Show that if $f(n)$ and $g(n)$ are monotonically increasing functions, then so are the functions $f(n) + g(n)$ and $f(g(n))$ and if $f(n)$ and $g(n)$ are in addition nonnegative, then $f(n)g(n)$ is monotonically increasing.

Solution 3.2-1 If $f(n)$ and $g(n)$ are monotonically increasing functions then for all n and n_0, $f(n + n_0) \ge f(n)$ and $g(n + n_0) \ge g(n)$. Then clearly $f(n) + g(n)$ is also monotonically increasing because

$f(n+n_0) + g(n+n_0) \geq f(n) + g(n)$. Similarly $f(g(n+n_0)) = f(k)$. Since $k \geq g(n)$, $f(k) \geq f(g(n))$. Therefore $f(g(n))$ is monotonically increasing. $f(n)g(n)$ is monotonically increasing if both are nonnegative because $f(n+n_0)g(n+n_0) \geq f(n)g(n+n_0) \geq f(n)g(n)$.

Problem 3.2-2 Prove $a^{log_b c} = c^{log_b a}$

Solution 3.2-2 Take the log base b of both sides. We get $(log_b a)(log_b c) = (log_b c)(log_b a)$.

Problem 3.2-3 Prove $\lg(n!) = \Theta(n \lg(n))$. Prove $n! = \omega(2^n)$ and $n! = o(n^n)$.

Solution 3.2-3 $lg(n!) = \sum\limits_{i=1}^{n} lg(i) = \Theta(n \lg(n))$. $n! = \omega(2^n)$. For $n > 4$ we can see that $i \cdot (n-i) > 4$. By pairing each i and $n-i$ we see that the product is greater than 4. Therefore it is easy to see $n! = \omega(2^n)$. Finally we show that $n! = o(n^n)$. $n! = \prod\limits_{i=1}^{n} i < \prod\limits_{i=1}^{n} n$.

Problem 3.2-4 Is the function $\lceil \lg n \rceil!$ polynomially bounded? Is the function $\lceil \lg \lg n \rceil!$

Solution 3.2-4 $\lceil \lg n \rceil! = O((\lg n)^{\lg n})$. If it is polynomially bounded then $(\lg n)^{\lg n} = O(n^a)$ for some constant a. Taking the log of both sides we get $(\lg n)(\lg \lg n) = a(\lg lgn)$. Clearly a is nonconstant. Therefore $\lceil \lg n \rceil!$ is not polynomially bounded.

$\lceil \lg \lg n \rceil! = O((\lg \lg n)^(\lg \lg n))$. If it is polynomially bounded then $(\lg \lg n)^{\lg \lg n} = O(n^a)$ for some constant a. Taking the log of both sides we get $(\lg \lg \lg n)(\lg \lg n) = a(\lg lgn)$. Clearly a is nonconstant. Therefore $\lceil \lg \lg n \rceil!$ is not polynomially bounded.

Problem 3.2-5 Which is asympotically larger: $\lg(\lg *n)$ or $\lg *(\lg n)$.

Solution 3.2-5 Assume $\lg *n = k$. Then $\lg *(\lg n) = k - 1$ because one lg factor is already introduced. Meanwhile $\lg(\lg *n) = \lg(k)$. Clearly $\lg *(\lg n)$ is asymptotically larger.

Problem 3.2-6 Show that the golden ratio ϕ and its conjugate $\bar{\phi}$ satisfy the equation $x^2 = x + 1$.

Solution 3.2-6 $\phi = (1+\sqrt{5})/2$. $\phi^2 = (3+\sqrt{5})/2 = 1+\phi$. Therefore ϕ satisfies the equation. Now for $\bar{\phi}$. $\bar{\phi}^2 = (3-\sqrt{5})/2 = 1+\bar{\phi}$.

Problem 3.2-7 Prove by induction that the ith Fibonacci number satisfies the equality: $F_i = \frac{\phi^i - \bar{\phi}^i}{\sqrt{5}}$.

Solution 3.2-7 $F_0 = 0$ and $F_1 = (1+\sqrt{5})/2 + (1-\sqrt{5})/2 = 1$. By induction if it is true for F_{i-2} and F_{i-1}. Can we show it is true for $F_i = F_{i-2} + F_{i-1} = \frac{\phi^{i-2}}{\sqrt{5}}(\phi+1) + \frac{\phi^{\bar{i-2}}}{\sqrt{5}}(\bar{\phi}+1)$. Since ϕ and $\bar{\phi}$ are solutions to the equation $x^2 = x+1$ we can substitute ϕ^2 and $\bar{\phi}^2$ for $(\phi+1)$ and $(\bar{\phi}+1)$ respectively. This reduces the original equation to: $F_i = F_{i-2} + F_{i-1} = \frac{\phi^{i-2}}{\sqrt{5}}(\phi^2) + \frac{\phi^{\bar{i-2}}}{\sqrt{5}}(\bar{\phi}^2) = \frac{\phi^i}{\sqrt{5}} + \frac{\bar{\phi}^i}{\sqrt{5}}$. This satisfies the inductive statement.

Problem 3.2-8 Show that $k \ln k = \Theta(n)$ implies $k = \Theta(n/\ln n)$.

Solution 3.2-8 Substituting we get $k \ln k = (n/\ln n)(\ln n - \ln\ln n) = (n/\ln n)\Theta(\ln n) = \boxed{\Theta(n)}$.

Problems

Problem 3-1 Let $p(n)$ be a polynomial of degree-d. Prove the following

a) If $k \geq d$ then $p(n) = O(n^k)$.

Solution If $k \geq d$ then n^k is an upperbound on any polynomial of degree d. We can quickly show this. For $n > a_d$, $2n^d > p(n)$. Therefore $2n^k > 2n^d > p(n)$ for $n > a_d$ so the big O bound is well defined.

b) If $k \leq d$ then $p(n) = \Omega(n^k)$.

Solution If $k \leq d$ then n^k is an lowerbound on any polynomial of degree d. We can quickly show this. For $n > a_d$, $n^d < p(n)$. Therefore $n^k < n^d < p(n)$ for $n > a_d$ so the Ω bound is well defined.

c) If $k = d$ then $p(n) = \Theta(n^k)$.

Solution If $k = d$ then n^k is an upperbound on any polynomial of degree d. We can quickly show this. For $n > a_d$, $2n^d > p(n)$. For $n > a_d$, $n^d < p(n)$. Therefore $n^k < n^d < p(n)$ for $n > a_d$ and $2n^k > 2n^d > p(n)$ for $n > a_d$ so the Θ bound is well defined.

d) If $k > d$ then $p(n) = o(n^k)$.

Solution If $k > d$ then n^k is an upperbound on any polynomial of degree d. We can quickly show this. For a constant c, $cn^k > p(n)$ for all $n \geq n_0$. Let a_max be the max coefficient. So clearly for $n > 2a_max$, $cn^k > p(n)$.

e) If $k < d$ then $p(n) = \omega(n^k)$.

Solution If $k < d$ then n^k is a lowerbound on any polynomial of degree d. We can quickly show this. For a constant c, $cn^k < p(n)$ for all $n \geq n_0$. Let a_d be the leading degree coefficient coefficient. So clearly for $n > c \cdot a_d$, $cn^k < p(n)$.

Problem 3-2 Determine the asymptotic relationships. Multiple may apply. Choose from $O, o, \Omega, \omega, \Theta$.

a) $A = \lg^k n$ and $B = n^\epsilon$

Solution Depends on ϵ and k.

b) $A = n^k$ and $B = c^n$.

Solution $A = O(B) = o(B)$

c) $A = \sqrt{n}$ and $B = n^{sin(n)}$

Solution $None$

d) $A = 2^n$ and $B = 2^{n/2}$

Solution $A = \omega(B) = \Omega(B)$

e) $A = n^{\lg c}$ and $B = c^{\lg n}$

Solution $A = \Theta(B) = O(B) = \Omega(B)$.

f) $A = \lg(n!)$ and $B = \lg(n^n)$

Solution $A = O(B) = o(B)$

Problem 3-3 Rank the following functions in decreasing asymptotic complexity.

a) $\lg(\lg^* n)$; $2^{\lg^* n}$; $(\sqrt{2})^{\lg n}$; n^2; $n!$; $(\lg n)!$; $(3/2)^n$; n^3; $\lg^2 n$; $\lg(n!)$; 2^{2^n}; $n^{1/\lg n}$; $\ln \ln n$; $\lg^* n$; $n2^n$; $n^{\lg \lg n}$; $\ln n$; 1; $2^{\lg n}$; $(\lg n)^(\lg n)$; $1)!$; $\sqrt{\lg n}$; $\lg^*(\lg n)$; $2^{\sqrt{(2)} \lg n}$; n; 2^n; $n \lg n$; 2^{2^n+1}

Solution $1 < n^{1/\lg n} < \lg(\lg^* n) < \lg^*(\lg n) < \lg^* n < \ln \ln n < \sqrt{\lg n} < \ln n < \lg^2 n < n < n \lg n < n^2 < n^3 < \lg(n!) < n^{\lg \lg n} < 2^{\lg^* n} < (\sqrt{2})^{\lg n} < 2^{\lg n} < 2^{\sqrt{(2)} \lg n} < 4^{\lg n} < 2^n < e^n < (3/2)^n < n2^n < (\lg n)! < (\lg n)^(\lg n) < n! < (n+1)! < 2^{2^n} < 2^{2^n+1}$

b) Give a function that is neither upper nor lower bounded by any of the functions in a)

Solution $(n^{n^n})^(sinn)$. Oscillates between extremely large and extremely small, therefore cannot be asymptotically bounded.

Problem 3-4 Prove or disprove the following,

a) $f(n) = O(g(n)) \to g(n) = O(f(n))$

Solution False. Just because $g(n)$ is an upperbound on $f(n)$ does not imply theat $f(n)$ is an upperbound on $g(n)$. In fact this is only possible if they are tight bounds.

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

Solution False. It is the maximum not the minimum. For example if $g(n) = 1$ and $f(n)$ is some polynomial then this equation is not true.

c) $f(n) = O(g(n)) \to lg(f(n)) = O(lg(g(n))$ where $lg(g(n)) \geq 1$ and $f(n) \geq 1$.

Solution True. As long as the function's log are nonnegative this is true.

d) $f(n) = O(g(n)) \to 2^{f(n)} = O(2^{g(n)})$.

Solution True as long as the functions are nonnegative. Exponentiation preserves asymptotic relationships.

e) $f(n) = O(f(n)^2)$.

Solution False. If $f(n) < 1$ this is not true.

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

Solution True. If $g(n)$ is an upper bound on $f(n)$ then $f(n)$ is a lower bound on $g(n)$

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

Solution False. Counterexample is $f(n) = 2^n$.

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

Solution True because $o(f(n)) + f(n) \leq c_0 f(n)$ for some c_0 and $n \geq n_0$. Furthermore $f(n) + o(f(n))$ is clearly lower bounded by $f(n)$. Therefore since $f(n) + o(f(n)) = O(f(n)) = \Omega(f(n))$, we have $f(n) + o(f(n)) = \Theta(f(n))$.

Problem 3-5 $\overset{\infty}{\Omega}$ is defined such that $f(n) = \overset{\infty}{\Omega}(g(n))$ if there exists a positive constant c such that $f(n) \geq cg(n) \geq 0$ for infinitely many integers n.

a) Show that one of the following two statements must be true: $f(n) = O(g(n))$ and $f(n) = \overset{\infty}{\Omega}(g(n))$.

Solution Proof by contradiction. Let us say both statements are false. If $f(n) \neq \overset{\infty}{\Omega}(g(n))$ then there are a finite number of values for which $f(n) \geq cg(n)$. Let us say the max of this set of values is n_0. Then for all n greater than n_0, $f(n) \leq cg(n)$ which implies $f(n) = O(g(n))$. This contradicts the assumption that both $f(n) = O(g(n))$ and $f(n) = \overset{\infty}{\Omega}(g(n))$ are false.

b) Describe the advantages of using $\overset{\infty}{\Omega}$ instead of Ω.

Solution Advantages are that all pairs of functions can now be categorized by either O or the modified Ω. Disadvantage is that there is no guarantee on asymptotic behavior. For example, oscillatory functions are not actually bounded asymptotically but they will show up as satisfying the modified Ω bound.

c) O' can be defined as $f(n) = O'(g(n))$ iff $|f(n)| = O(g(n))$. What happens to the iff in theorem 3.1 if we use O and O': For any two functions $f(n)$ and $g(n)$, we have $f(n) = \Theta(g(n))$ iff $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

Solution The iff only applies if $f(n) \geq 0$.

d) O is bigO notation with logarithmic factors ignored. $O(g(n)) = \{f(n) :$ there exist positive constants c, k and n_0 such that $0 \leq f(n) \leq cg(n) \lg^k(n)$ for all $n \geq n_0\}$. Define the same for Ω and Θ.

Solution $\Omega(g(n)) = \{f(n) :$ there exist positive constants c, k and n_0 such that $0 \leq cg(n) \lg^k(n) \leq f(n)$ for all $n \geq n_0\}$. $\Theta(g(n)) = \{f(n) :$ there exist positive constants c_1, c_2, k and n_0 such that $0 \leq c_1 g(n) \lg^k(n) \leq f(n) \leq c_2 g(n) \lg^k(n)$ for all $n \geq n_0\}$.

Problem 3-6 $f_c^*(n) = \min\{i \geq 0 : f^{(i)}(n) \leq c\}$. Solve the following iterated functions.

a) $f(n) = n - 1$ and $c = 0$.

Solution $f(n)^k = n - k \leq 0 \rightarrow \boxed{k = \lceil n \rceil}$.

b) $f(n) = \lg n$ and $c = 1$

Solution This is the definition of $lg^* n$.

c) $f(n) = n/2$ and $c = 1$

Solution $\boxed{\lceil \lg n \rceil}$.

d) $f(n) = n/2$ and $c = 2$

Solution $\boxed{\lceil \lg n \rceil - 1}$.

e) $f(n) = \sqrt{n}$ and $c = 2$

Solution $n^{1/2^k} < 2 \rightarrow 2^{2^k} > n \rightarrow \boxed{k = \lceil \lg \lg n \rceil}$.

f) $f(n) = \sqrt{n}$ and $c = 1$

Solution $\boxed{\infty}$ if $n > 1$

g) $f(n) = n^{1/3}$ and $c = 2$

Solution $n^{1/3^k} < 2 \rightarrow 2^{3^k} > n \rightarrow \boxed{k = \lceil \lg \log_3 n \rceil}$.

h) $f(n) = n / \lg n$ and $c = 2$

Solution $f^{(1)}(n) = \frac{n}{(\lg n)(\lg n - \lg \lg n)}$. This is $\boxed{\Omega(\lg^* n)}$.