Exercises in Section 3.1

1. Modify the lower-bound argument for insertion sort to handle input sizes that are not necessarily a multiple of 3.

   Solution

   Based on Figure 3.1,

   - For $n = 3k$, each of the k largest values moves through each of these k positions to somewhere in these k positions: $(n/3)(n/3) = n^2/9 = \Omega(n^2)$.

   - For $n = 3k+1$, each of the k largest values moves through each of these $k+1$ positions to somewhere in these k positions: $((n-1)/3)((n+2)/3) = n^2/9 + n/9 - 2/9 = \Omega(n^2)$.

   - For $n = 3k+2$, each of the k largest values moves through each of these $k+2$ positions to somewhere in these k positions: $((n-2)/3)((n+1)/3) = n^2/9 - n/9 - 2/9 = \Omega(n^2)$.

2. Using reasoning similar to what we used for insertion sort, analyze the running time of the selection sort algorithm from Exercise 2.2-2.

   Solution

   The two **for** loops must be executed $(n-1)(n-1) = \Theta(n^2)$ times in any case. Therefore the running time of the selection sort is $\Theta(n^2)$.

3. Suppose that α is a fraction in the range $0 < \alpha < 1$. Show how to generalize the lower-bound argument for insertion sort to consider an input in which the α largest values start in the first αn positions. What additional restriction do you need to put on α? What value of α maximizes the number of times that the α largest values must pass through each of the middle $(1 - 2\alpha)n$ array positions?

   Solution

   Additional restriction: $\alpha < 1/2$.

To maximize $T(n) = \alpha(1 - 2\alpha)n^2$, find the partial derivative:

$$\frac{\partial T}{\partial \alpha} = (1 - 4\alpha)n^2,$$

which equals zero when $\alpha = 1/4$.

Exercises in Section 3.2

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

   Solution

   Proof. Since $f(n)$ and $g(n)$ are asymptotically nonnegative, take n_0 such that $f(n) \geq 0$ and $g(n) \geq 0$. Take $c_1 = 1/2$ and $c_2 = 1$. It follows that

   $$0 \leq \frac{f(n) + g(n)}{2} \leq \max\{f(n), g(n)\} \leq f(n) + g(n)$$

   for all $n \geq n_0$. $\qquad\square$

2. Explain why the statement, "The running time of algorithm A is at least $O(n^2)$," is meaningless.

   Solution

   $O(n^2)$ is an upper bound. You can't say a number is at least bounded above by 42.

3. Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?

   Solution

   - Yes, $2^{n+1} = 2 \cdot 2^n = O(2^n)$.

   - No, $2^{2n} = \omega(2^n)$. For any constant $c > 0$, take $n_0 = \lceil \lg c \rceil + 1$, it follows that

     $$0 \leq c \cdot 2^n < 2^n \cdot 2^n = 2^{2n}$$

     for all $n \geq n_0$.

4. Prove Theorem 3.1.

   Solution

Recall Theorem 3.1: For any two functions $f(n)$ and $g(n)$, we have $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

Proof. The forward direction is trivial. For the reverse direction, assume $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$. Let c_1, n_1 be constants such that $0 \leq f(n) \leq c_1 \cdot g(n)$ for all $n \geq n_1$ and c_2, n_2 be constants such that $0 \leq c_2 \cdot g(n) \leq f(n)$ for all $n \geq n_2$. It follows that

$$0 \leq c_2 \cdot g(n) \leq f(n) \leq c_1 \cdot g(n)$$

for all $n \geq \max\{n_1, n_2\}$, so $f(n) = \Theta(g(n))$. □

5. Prove that the running time of an algorithm is $\Theta(g(n))$ if and only if its worst-case running time is $O(g(n))$ and its best-case running time is $\Omega(g(n))$.

Solution

Proof. The forward direction is trivial. For the reverse direction, let $T(n)$ be the running time of the algorithm, and assume $W(n) = O(g(n))$ and $B(n) = \Omega(g(n))$ be the worst-case and best case running times, respectively. Take n_1 such that $W(n) \geq T(n)$ for all $n \geq n_1$ and n_2 such that $B(n) \leq T(n)$ for all $n \geq n_2$.

Since $W(n) = O(g(n))$ and $B(n) = \Omega(g(n))$, there are constants c_1, c_2 and n_3, n_4 such that

$$0 \leq W(n) \leq c_1 \cdot g(n),$$

for all $n \geq n_3$, and

$$0 \leq c_2 \cdot g(n) \leq B(n)$$

for all $n \geq n_4$. It follows that

$$0 \leq c_2 \cdot g(n) \leq B(n) \leq T(n) \leq W(n) \leq c_1 \cdot g(n)$$

for all $n \geq \max\{n_1, n_2, n_3, n_4\}$, so $T(n) = \Theta(g(n))$. □

6. Prove that $o(g(n)) \cap \omega(g(n))$ is the empty set.

Solution

Proof: Suppose there are function $h(n)$ and $g(n)$ such that $h(n) = o(g(n))$ and $h(n) = \omega(g(n))$. Take n_1, n_2 such that $0 \leq h(n) < g(n)$ for all $n \geq n_1$ and $0 \leq g(n) < h(n)$ for all $n \geq n_2$. Let $n_0 = \max\{n_1, n_2\}$, we would have $h(n_0) < g(n_0) < h(n_0)$, which is a consideration. □

7. We can extend our notation to the case of two parameters n and m that can go to ∞ independently at different rates. For a given function $g(n, m)$, we denote by $O(g(n, m))$ the set of functions

$$O(g(n, m)) = \{f(n, m) : \text{there exist positive constants } c, n_0, m_0$$
$$\text{such that } 0 \leq f(n, m) \leq cg(n, m)$$
$$\text{for all } n \geq n_0 \text{ or } m \geq m_0\}.$$

Give corresponding definitions for $\Omega(g(n, m))$ and $\Theta(g(n, m))$.

Solution

$$\Omega(g(n, m)) = \{f(n, m) : \text{there exist positive constants } c, n_0, m_0$$
$$\text{such that } 0 \leq cg(n, m) \leq f(n, m)$$
$$\text{for all } n \geq n_0 \text{ or } m \geq m_0\}.$$

$$\Theta(g(n, m)) = \{f(n, m) : \text{there exist positive constants } c_1, c_2, n_0, m_0$$
$$\text{such that } 0 \leq c_1 g(n, m) \leq f(n, m) \leq c_2 g(n, m)$$
$$\text{for all } n \geq n_0 \text{ or } m \geq m_0\}.$$

Exercises in Section 3.3

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 $f(n) \cdot g(n)$ is monotonically increasing.

   Solution

   Proof.

   - If $m < n$, then

   $$(f + g)(m) = f(m) + g(m) \le f(n) + g(n) = (f + g)(n).$$

   - If $m > n$, then

   $$(f \circ g)(m) = f(g(m)) \le f(g(n)) = (f \circ g)(n).$$

   - Given that $f(n)$ and $g(n)$ are nonnegative, if $m > n$, then

   $$(fg)(m) = f(m) \cdot g(m) \le f(n) \cdot g(n) = (fg)(n).$$

   $\square$

2. Prove that $\lfloor \alpha n \rfloor + \lceil (1 - \alpha)n \rceil = n$ for any integer n and real number α in the range $0 \le \alpha \le 1$.

   Solution

   Proof.

   $$\lceil (1 - \alpha)n \rceil = \lceil n - \alpha n \rceil = n + \lceil -\alpha n \rceil = n - \lfloor \alpha n \rfloor.$$

   $\square$

3. Use equation (3.14) or other means to show that $(n + o(n))^k = \Theta(n^k)$ for any real constant k. Conclude that $\lceil n \rceil^k = \Theta(n^k)$ and $\lfloor n \rfloor^k = \Theta(n^k)$.

   Solution

Proof. We only consider $k > 0$. Let $f(n) \in o(n)$. Then, there exists n_0 such that

$$0 \le f(n) < \frac{1}{k}n$$

for all $n \ge n_0$. Therefore,

$$0 \le (n + f(n))^k < (n + \frac{1}{k}n)^k = (1 + \frac{1}{k})^k n^k < en^k$$

for all $n \ge n_0$, which implies that $(n + f(n))^k = \Theta(n^k)$. $\quad\square$

4. Prove the following:

 a. Equation (3.21).

 b. Equation (3.26)-(3.28).

 c. $\lg(\Theta(n)) = \Theta(\lg n)$.

Solution

 a. *Proof.* Let $d = \log_b c$, then $c = b^d$. Therefore,

$$c^{\log_b a} = (b^d)^{\log_b a} = (b^{\log_b a})^d = a^d = a^{\log_b c}.$$

$\quad\square$

 b.   • *Proof.* For any constant $c > 0$, take $n_0 = \lceil 1/c \rceil + 1$, it follows that

$$0 \le n! = \frac{n!}{n^n}n^n \le \frac{1}{n}n^n < \frac{1}{\lceil \frac{1}{c} \rceil}n^n \le cn^n$$

for all $n \ge n_0$. $\quad\square$

   • *Proof.* For any constant $c > 0$, take $n_0 = \lceil 4c \rceil + 1$, it follows that

$$n! = \frac{n!}{2^n} \cdot 2^n \ge \frac{n}{2} \cdot \frac{1}{2} \cdot 2^n = \frac{n}{4} \cdot 2^n > \frac{\lceil 4c \rceil}{4} \cdot 2^n \ge c \cdot 2^n$$

for all $n \ge n_0$. $\quad\square$

   • *Proof.* Equation (3.26) implies

$$\lg(n!) = O(\lg(n^n)) = O(n \lg n).$$

7

On the other hand, we have

$$\lg(n!) = \lg\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\alpha_n}\right), \quad \text{where } \frac{1}{12n+1} < \alpha_n < \frac{1}{12n}.$$

$$= \lg\sqrt{2\pi n} + n\lg n - n\lg e + \alpha_n \lg e$$

$$= n\lg n - n\lg e$$

$$\geq \frac{n}{2}\lg n + n\lg\sqrt{n} - n\lg e$$

$$\geq \frac{n}{2}\lg n$$

for all $n \geq 8$, which implies $\lg(n!) = \Omega(n\lg n)$. □

c. *Proof.* Let $f(n) = \Theta(n)$. Take c_1, c_2, n_0 such that $0 \leq c_1 n \leq f(n) \leq c_2 n$ for all $n \geq n_0$. Take logarithm on all sides, we get

$$\lg c_1 + \lg n \leq \lg(f(n)) \leq \lg c_2 + \lg n.$$

For all $n \geq c_1^{-2}$, we have $\lg n \geq -2\lg c$, therefore $\lg c \geq -(1/2)\lg n$. On the other hand, for all $n \geq c_2$, we have $\lg c_2 \leq \lg n$. Hence,

$$0 \leq \frac{1}{2}\lg n \leq \lg(f(n)) \leq 2\lg n$$

for all $n \geq \max\{n_0, c_1^{-2}, c_2\}$. □

5. Is the function $\lceil \lg n \rceil!$ polynomially bounded? Is the function $\lceil \lg \lg n \rceil!$ polynomially bounded?

Solution

There is an obvious equivalence: $f(n)$ is polynomially bounded if and only if $\lg(f(n)) = O(\lg n)$.

- $\lceil \lg n \rceil!$ is not polynomially bounded because

$$\lceil \lg n \rceil! = \Theta(\lg n \lg \lg n) = \omega(\lg n).$$

- $\lceil \lg \lg n \rceil!$ is polynomially bounded because

$$\lceil \lg \lg n \rceil! = \Theta(\lg \lg n \lg \lg \lg n) = O((\lg \lg n)^2) = O(\lg n).$$

6. Which is asymptotically larger: $\lg(\lg^* n)$ or $\lg^*(\lg n)$?

Solution

$\lg^*(\lg n)$ is asymptotically larger because

$$\lg^*(\lg n) = \lg^* n - 1 = \Theta(\lg^* n) = \omega(\lg(\lg^* n)).$$

7. Show that the golden ratio ϕ and its conjugate $\hat{\phi}$ both satisfy the equation $x^2 = x + 1$.

Solution

$$\phi^2 = \left(\frac{1 + \sqrt{5}}{2}\right)^2 = \frac{6 + 2\sqrt{5}}{4} = 1 + \frac{1 + \sqrt{5}}{2} = 1 + \phi.$$

$$\hat{\phi}^2 = \left(\frac{1 - \sqrt{5}}{2}\right)^2 = \frac{6 - 2\sqrt{5}}{4} = 1 + \frac{1 - \sqrt{5}}{2} = 1 + \hat{\phi}.$$

8. Prove by induction that the ith Fibonacci number satisfies the equation

$$F_i = (\phi^i - \hat{\phi}^i)/\sqrt{5},$$

where ϕ is the golden ratio and $\hat{\phi}$ is its conjugate.

Solution

Proof.

- Base case: $(\phi^0 - \hat{\phi}^0)/\sqrt{5} = 0 = F_0, \quad (\phi^1 - \hat{\phi}^1)/\sqrt{5} = 1 = F_0$

- Inductive step: Assume that the equation holds for $i = k$ and $i = k + 1$. Then,

$$F_{k+2} = F_k + F_{k+1}$$

$$= \frac{\phi^k - \hat{\phi}^k}{\sqrt{5}} + \frac{\phi^{k+1} - \hat{\phi}^{k+1}}{\sqrt{5}}$$

$$= \phi^k \frac{1 + \phi}{\sqrt{5}} - \hat{\phi}^k \frac{1 + \hat{\phi}}{\sqrt{5}}$$

$$= \phi^k \frac{\phi^2}{\sqrt{5}} - \hat{\phi}^k \frac{\hat{\phi}^2}{\sqrt{5}}$$

$$= \frac{\phi^{k+2} - \hat{\phi}^{k+2}}{\sqrt{5}}.$$

$\square$

9. Show that $k \lg k = \Theta(n)$ implies $k = \Theta(n/\lg n)$.

Solution

Proof. Using the property of Exercise 3.3-4(c), it follows that

$$\lg k + \lg \lg k = \lg(k \lg k) = \lg(\Theta(n)) = \Theta(\lg n).$$

Since $\lg \lg k = o(\lg k)$, we have $\lg k = \Theta(\lg n)$. Therefore,

$$k = \frac{k \lg k}{\lg k} = \frac{\Theta(n)}{\Theta(\lg n)} = \Theta(n/\lg n).$$

□