Show All Solutions

Rosen, Discrete Mathematics and Its Applications, 7th edition
Extra Examples
Section 8.3—Divide-and-Conquer Algorithms and Recurrence Relations

Extra Examples — *Page references correspond to locations of Extra Examples icons in the textbook.*

---

p.528, icon at Example 1
#1. Suppose $f(n) = 3f(n/2) + 4$ and $f(1) = 5$. Find $f(8)$.

See Solution

---

p.528, icon at Example 1
#2. Suppose $f(n) = 2f(n/3) - 1$ and $f(1) = 2$. Find $f(9)$.

See Solution

---

p.528, icon at Example 1
#3. Suppose $f(n) = 5f(n/2) + 2n - 1$ and $f(4) = 40$. Find $f(1)$.

See Solution

---

p.531, icon at Example 6
#1. Suppose $f(n) = 2f(n/3) + 3$. Find a big-oh function for f.

See Solution

p.531, icon at Example 6

#2. A recursive algorithm for finding the maximum of a list of numbers divides the list into three equal (or nearly equal) parts, recursively finds the maximum of each sublist, and then finds the largest of these three maxima. Let $f(n)$ be the total number of comparisons needed to find the maximum of a list of n numbers (n a power of 3). Set up a recurrence relation for $f(n)$ and give a big-oh estimate for f.

See Solution