8.3 Divide-and-Conquer Algorithms and Recurrence Relations

8.3 pg. 535 # 9

Suppose that $f(n) = f(n/5) + 3n^2$ when n is a positive integer divisible by 5, and $f(1) = 4$. Find

a $f(5)$.

b $f(125)$.

c $f(3125)$.

8.3 pg. 535 # 11

Give a big-O estimate for the function $f(n) = f(n/2)+1$ if f is an increasing function and $n = 2^k$.

8.3 pg. 535 # 13

Give a big-O estimate for the function $f(n) = 2f(n/3) + 4$ if f is an increasing function and $n = 3^k$.

8.3 pg. 535 # 17

Suppose that the votes of n people for different candidates (where there can be more than two candidates) for a particular office are the elements of a sequence. A person wins the election if this person receives a majority of the votes.

a Devise a divide-and-conquer algorithm that determines whether a candidate received a majority and, if so, determine who this candidate is. [*Hint:* Assume that n is even and split the sequence of votes into two sequences, each with $n/2$ elements. Note that a candidate could not received a majority of votes without receiving a majority of votes in at least one of the two halves.]

b Use the master theorem to give a big-O estimate for the number of comparisons needed by the algorithm you devised in part (a).