CECS 528, Learning Outcome Assessment 2, 9/11/2024, Dr. Ebert

LO1(b) Assume $T(k) \ge ck\log k$ for some const. $c>0$ and all $k<n$. Show $T(n) \ge cn\log n$.

Directions: show all work.

$T(n) = T(\frac{n}{3}) + T(2n/3) + n \ge \frac{cn}{3}\log(\frac{n}{3}) + \frac{2cn}{3}\log(\frac{2n}{3}) + n =$

$cn\frac{(\log n - \log 3)}{3} + \frac{2}{3}cn(\log n + \log(2/3)) + n =$

$cn\log n + cn(\frac{2}{3}\log(2/3) - \frac{\log 3}{3}) + n \ge cn\log n \iff$

$\boxed{c \le \frac{1}{(\log 3/3 - \frac{2}{3}\log 2/3)}}$ ✓ Note: $\log 3 > \log(\frac{4}{9})$

Problem

LO1. Solve the following problems.

(a) Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 5T(n/2) + n^{\log_3 27}$. Defend your answer.

$n^{\log_3 27} = n^3 = \Omega(n^{\log 5+\epsilon})$ where $\epsilon = 3 - \log 5$

(b) Use the substitution method to prove that, if $T(n)$ satisfies

Thus, $\boxed{T(n) = \Theta(n^3)}$

$$T(n) = T(n/3) + T(2n/3) + n$$

then $T(n) = \Omega(n \log n)$.

LO2. Solve each of the following problems.

# of mults

size of each number

addition shift are both linear as well as subtraction

$T(n) = 3T(n/2) + n$

(a) Consider the following algorithm of Karatsuba called `multiply` for multiplying two n-bit binary numbers x and y. In what follows, we assume n is even. Let x_L and x_R be the leftmost $n/2$ and rightmost $n/2$ bits of x respectively. Define y_L and y_R similarly. Let P_1 be the result of calling `multiply` on inputs x_L and y_L, P_2 be the result of calling `multiply` on inputs x_R and y_R, and P_3 the result of calling `multiply` on inputs $x_L + x_R$ and $y_L + y_R$. Then return the value $P_1 \times 2^n + (P_3 - P_1 - P_2) \times 2^{n/2} + P_2$. Provide a divide-and-conquer recurrence for the running time of this algorithm. Justify your choices of a, b, and $f(n)$.

(b) Recall that the `Minimum Positive Subsequence Sum (MPSS)` problem admits a divide-and-conquer algorithm that, on input integer array a, requires computing the mpss of any subarray of a that contains both $a[n/2-1]$ and $a[n/2]$ (the end of a_{left} and the beginning of a_{right}). For

a_L a_R

$T(n) = 2T(n/2) + n\log n$

$$a = 33, -34, -1, -31, 14, -20, 61, -7, 83, -10$$

provide the two sorted arrays a = SortedLeftSums and b = SortedRightSums from which the minimum positive sum $a[i] + b[j]$ represents the desired mpss (for the middle), where i in the index range of a and j is within the index range of b. Also, demonstrate how the minimum positive sum $a[i] + b[j]$ may be computed via the movement of left and right markers.

Left Sums: 14, −17, −18, −52, −19 Sorted: −52, −19, −18, −17, 14

Right Sums: −20, 41, 34, 117, 107 Sorted: −20, 34, 41, 107, 117

$a_L[0] + a_R[4] = 65$ $a_L[0] + a_R[3] = 55$ $a_L[0] + a_R[2] = -11$

$a_L[1] + a_R[2] = 22$ $a_L[1] + a_R[1] = 15$ $a_L[1] + a_R[0] = -39$

MPPS middle = 15

$a_L[2] + a_R[0] = -38$ $a_L[3] + a_R[0] = -37$

$a_L[4] + a_R[0] = -6$ Note: $T(n) = \Theta(n\log^2 n)$