**Note:** Please deposit your solutions in the appropriate box **by 4 p.m. on the due date**. Late assignments or assignments placed into incorrect boxes will not be marked. Use a cover sheet available from outside the Resource Centre.

PLEASE SHOW ALL WORKING.

**1.** Show how we can split the polynomials into three parts and obtain an algorithm for multiplying polynomials of degree $n$ that runs in time $O(n^{\log_3 5})$.

**2.** Prove by constructive induction that the recurrence inequality

$$t(n) \leq t(\lceil n/5 \rceil) + t(\lceil 7n/10 \rceil) + dn \qquad (n \geq 10)$$

that we saw in the analysis of the median-finding algorithm has its solution in $O(n)$ for every fixed positive constant $d$ and any nonnegative initial values $t_0, t_1, \ldots, t_9$.

**3.** Solve the given recurrence exactly, assuming that $n$ is a power of 2. Describe what kind of algorithm would give rise to this sort of recurrence for its running time. If $T$ is known to be increasing, what can you say about its asymptotic growth rate?

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 2T(n/2) + \lg n & \text{if } n > 1. \end{cases}$$

**4.** Consider the sorting algorithm that uses mergesort for input above a certain threshold size and selection sort below that threshold. Considering only the number of comparisons made as your measure of cost, estimate the optimal value of this threshold using the methods shown in lecture.

**5.** The obvious algorithm to find the minimum and maximum elements of an array is to first scan through for the maximum and then scan through for the minimum. This takes $2(n-1)$ comparisons. Give a divide-and-conquer algorithm that finds the minimum and maximum using at most $cn$ comparisons where $c$ is some constant less than 2. Find the smallest value you can for $c$.

**6.** Do Exercise 6.1 page 312 of the textbook.

**7.** Use the Subset-Sum algorithm given in Section 6.4 for the following input: $W = 30$ with $w_1 = 4$, $w_2 = 6$, $w_3 = 7$, $w_4 = 9$, $w_5 = 11$, $w_6 = 12$, $w_7 = 14$. (Give a table like the one on page 270 of the textbook.)

**8.** Do Exercise 6.8 page 319 of the textbook.

**9.** Give an example of input of size $n = 5$ for the Knapsack Problem that will cause the greedy algorithm to fail. Justify your answer.

**10.** Do Exercise 6.11 page 323 of the textbook.