# 4231 Homework 1

Eumin Hong (eh2890)

February 9, 2022

## Problem 1

Exercise 3.1-1 (Page 52) and 3-1 (a) and (d) (Page 61) of the textbook. For Exercise 3.1-1, you can assume both functions to take nonnegative values.

### 3.1-1

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

### 3-1

Let

$$p(n) = \sum_{i=0}^{d} a_i n^i,$$

where $a_d > 0$, be a degree-$d$ polynomial in $n$, and let $k$ be a constant. Use the definitions of asymptotic notations to prove the following properties.

***a.*** If $k \geq d$, then $p(n) = O\left(n^k\right)$.

***d.*** If $k > d$, then $p(n) = o\left(n^k\right)$.

# Problem 2

Problem 2-3 (Page 41). Skip (b).

## 2-3

The following code fragment implements Horner's rule for evaluating a polynomial

$$\begin{aligned} P(x) &= \sum_{k=0}^{n} a_k x^k \\ &= a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + x a_n) \cdots)), \end{aligned}$$

given the coefficients $a_0, a_1, \ldots, a_n$ and a value for $x$:

1: $y = 0$
2: **for** $i = n \ldots 0$ **do**
3: $\quad y = a_i + x \cdot y$

***a.*** In terms of $\Theta$-notation, what is the running time of this code fragment for Horner's rule?

***c.*** Consider the following loop invariant:

At the start of each iteration of the for loop of lines 2–3,

$$y = \sum_{k=0}^{n-(i+1)} a_{k+i+1} x^k.$$

Interpret a summation with no terms as equaling 0. Following the structure of the loop invariant proof presented in this chapter, use this loop invariant to show that, at termination, $y = \sum_{k=0}^{n} a_k x^k$.

***d.*** Conclude by arguing that the given code fragment correctly evaluates a polynomial characterized by the coefficients $a_0, a_1, \ldots, a_n$.

# Problem 3

(Wait for class on Jan 26) Exercise 2.3-7 (Page 39).

## 2.3-7

Describe a $\Theta(n \lg n)$-time algorithm that, given a set $S$ of $n$ integers and another integer $x$, determines whether or not there exist two elements in $S$ whose sum is exactly $x$.

# Problem 4

(Wait for class on Jan 26) Exercise 4.4-4 (page 93). Use the substitution method to prove your upper bound.

## 4.4-4

Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 2T(n-1) + 1$. Use the substitution method to verify your answer.

# Problem 5

(Wait for class on Jan 26) Problem 4-3 (a), (c) and (j) (Page 108): Use Master theorem on (a) and (c). For (j) it suffices to draw its recursion tree and conclude with your best guess.

## 4-3

Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for sufficiently small $n$. Make your bounds as tight as possible, and justify your answers.

***a.*** $T(n) = 4T(n/3) + n \lg n$.

***c.*** $T(n) = 4T(n/2) + n^2\sqrt{n}$.

***j.*** $T(n) = \sqrt{n}T(\sqrt{n}) + n$.