

**MATH182 HOMEWORK #1**  
**DUE June 18, 2020**

Note: while you are encouraged to work together on these problems with your classmates, your final work should be written in your own words and not copied verbatim from somewhere else. You need to do at least seven (7) of these problems. All problems will be graded, although the score for the homework will be capped at  $N := (\text{point value of one problem}) \times 7$  and the homework will be counted out of  $N$  total points. Thus doing more problems can only help your homework score. For the programming exercise you should submit the final answer (a number) *and* your program source code.

**Exercise 1.** Write out the following two sums in full:

- (1)  $\sum_{0 \leq k \leq 5} a_k$
- (2)  $\sum_{0 \leq k^2 \leq 5} a_{k^2}$

**Exercise 2.** Evaluate the following summation:

$$\sum_{k=1}^n k 2^k.$$

*Hint: rewrite as a double sum.*

**Exercise 3.** Suppose  $x \neq 1$ . Prove that

$$\sum_{j=0}^n j x^j = \frac{n x^{n+2} - (n+1) x^{n+1} + x}{(x-1)^2}.$$

*Challenge: do this without using mathematical induction.*

**Exercise 4** (Recommended! Horner's rule). The following code fragment implements Horner's rule for evaluating a polynomial

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

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

```
1  y = 0
2  for i = n downto 0
3      y = ai + x · y
```

- (1) In terms of  $\Theta$ -notation, what is the running time of this code fragment for Horner's rule?
- (2) Write pseudocode to implement the naive polynomial-evaluation algorithm that computes each term of the polynomial from scratch. What is the running time of this algorithm? How does it compare to Horner's rule?
- (3) Consider the following loop invariant:

At the start of each iteration of the **for** loop of lines 2-3

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

1

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,  $y = \sum_{k=0}^n a_k x^k$ .

- (4) Conclude by arguing that the given code fragment correctly evaluates a polynomial characterized by the coefficients  $a_0, a_1, \dots, a_n$ .

**Exercise 5.** Suppose  $m, n \in \mathbb{Z}$  are such that  $m > 0$ . Prove that

$$\left\lceil \frac{n}{m} \right\rceil = \left\lfloor \frac{n + m - 1}{m} \right\rfloor$$

This gives us another reflection principle between floors and ceilings when the argument is a rational number.

**Exercise 6.** Find a necessary and sufficient condition on the real number  $b > 1$  such that

$$\lfloor \log_b x \rfloor = \lfloor \log_b \lfloor x \rfloor \rfloor$$

holds for all real numbers  $x \geq 1$ .

**Exercise 7.** Suppose  $0 < \alpha < \beta$  and  $0 < x$  are real numbers. Find a closed formula for the sum of all integer multiples of  $x$  in the closed interval  $[\alpha, \beta]$ .

**Exercise 8.** How many of the numbers  $2^m$ , for  $0 \leq m \leq M$  (where  $m, M \in \mathbb{N}$ ), have leading digit 1 when written in decimal notation? Your answer should be a closed formula.

**Exercise 9.** Suppose  $x, y, z \in \mathbb{Z}$  are such that  $y, z \geq 1$ . Prove that  $z(x \bmod y) = (zx) \bmod (zy)$ .

**Exercise 10.** Suppose  $a, b, r, s \in \mathbb{Z}$  are such that  $r, s \geq 1$ . Prove that if  $a \bmod rs = b \bmod rs$ , then  $a \bmod r = b \bmod r$  and  $a \bmod s = b \bmod s$ .

**Exercise 11.** Suppose  $b > 1$ . Express  $\log_b \log_b x$  in terms of  $\ln \ln x$ ,  $\ln \ln b$ , and  $\ln b$ .

**Exercise 12** (Programming exercise, also doable by hand). If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Find the sum of all the multiples of 3 or 5 below 1000000.

**Exercise 13** (Programming exercise). Let  $F_0, F_1, F_2, \dots$  be the sequence of Fibonacci numbers. Compute the following sum:

$$\sum_{\substack{F_n < 10^7 \\ F_n \bmod 2 = 0}} F_n$$

**Exercise 14** (Programming exercise). Determine the following number:

$$\min \{n \in \mathbb{N} : n \geq 1 \text{ and for each } k \in \{1, 2, \dots, 30\}, k|n\}$$

Note: the above number certainly exists since the above set contains the number  $30!$ , so it is non-empty and thus has a minimum element by the Well-Ordering Principle.

**Exercise 15** (Programming exercise). Let  $P_n$  denote the  $n$ th prime number. So  $P_1 = 2, P_2 = 3, P_3 = 5, P_4 = 7, \dots$ . Find  $P_{100000}$ .

**Exercise 16** (Programming exercise). A unit fraction contains 1 in the numerator. The decimal representation of the unit fractions with denominators 2 to 10 are given:

$$\begin{aligned} 1/2 &= 0.5, & 1/3 &= 0.(3), & 1/4 &= 0.25, & 1/5 &= 0.2, & 1/6 &= 0.1(6), \\ 1/7 &= 0.(142857), & 1/8 &= 0.125, & 1/9 &= 0.(1), & 1/10 &= 0.1 \end{aligned}$$

where  $0.1(6)$  means  $0.166666\dots$ , and has a 1-digit recurring cycle. It can be seen that  $1/7$  has a 6-digit recurring cycle. Find the value of  $d < 3000$  for which  $1/d$  contains the longest recurring

cycle in its decimal fraction part. Hint: first analyze by hand what you would have to do to notice that  $1/7$  has a 6-digit recurring cycle, think about it in terms of the Division Algorithm.

**Exercise 17.** Suppose  $f(n)$  and  $g(n)$  are asymptotically positive functions. Prove that  $f(n) = \Theta(g(n))$  iff  $f(n) = O(g(n))$  and  $f(n) = \Omega(g(n))$ .

**Exercise 18.** Prove that for all  $m \in \mathbb{N}$ ,  $(\ln n)^m = o(n)$ .