

## WEEK 07

### 1. PREPARATION FOR ASSIGNMENT

If, and *only if* you can truthfully assert the truthfulness of each statement below are you ready to start the exercises.

#### 1.1. Reading Comprehension Self-Check.

- I know that **input enhancement** is the idea of preprocessing some or all of a problem's input, and storing the additional information obtained to accelerate solving the problem afterward.
- I know why it is **false** to say that the two principal resources of time and space compete with each other in **all** design situations.
- We know why sorting by distribution counting is more efficient than quick-sort.
- I know why it is **false** to say that data compression is a typical space-time tradeoff.
- I know that **prestructuring** creates structures that allow faster and/or more flexible access to data.
- I know why it is **false** to say that these pre-structured structures typically use less space than otherwise required.
- I know that hashing enables, on average, constant-time searching, insertion, and deletion.

1.2. **Memory Self-Check.** For  $k$  being a letter of some alphabet, let the function  $ord :: k \rightarrow a :: Integer$  have a value that is the position of  $k$  in that alphabet.

Which hash function below produces a hash with fewer potential collisions?

(1)

$$hash :: [k] \rightarrow a :: Integer$$

$$hash :: [h \mid t] \rightarrow \\ ord\ h + hash\ t$$

, or

---

*Date:* November 10, 2021.

- (2) Let  $c$  be greater than any element of the alphabet and  $\text{mod} :: a \rightarrow b$  be the modulo function  $a \bmod b$ .

$$\text{hash} :: [k] \rightarrow a :: \text{Integer}$$

$$\begin{aligned} \text{hash} :: [h \mid t] \rightarrow \\ \text{mod } h \cdot c + (\text{ord } h) \, m \end{aligned}$$

Why?

## 2. WEEK 07 EXERCISES

- 2.1. **Exercise 1 on page 274.**
- 2.2. **Exercise 2 on page 274.**
- 2.3. **Exercise 3 on page 275.**
- 2.4. **Exercise 4 on page 275.**
- 2.5. **Exercise 3 on page 279.**
- 2.6. **Exercise 6 on page 279.**

## 3. WEEK 07 PROBLEMS

- 3.1. **Not in the Book.** Consider a significant algorithm you have implemented in the past. Review it to see if there is a way to use additional memory to speed up the calculation of the result.