

3.4: Exercises

Exercise 3.4.1

Traditionally, computers keep track of the time/date using a format known as [Unix time](#), which counts the number of seconds that have elapsed since 00:00:00 UTC on Thursday, 1 January 1970. But there's a problem if we track Unix time using a fixed-width integer, since that has a maximum value. Beyond this date, the Unix time counter will roll-over, wreaking havoc on computer systems. Calculate the roll-over date for:

1. Ordinary (signed) 32-bit integers
2. *Unsigned* 32-bit integers, which do not reserve a bit for the sign (and thus store only non-negative numbers).
3. Signed 64-bit integers
4. Unsigned 64-bit integers

Exercise 3.4.2

Find the runtime of each of the following Python code samples (e.g. $O(1)$ or $O(N)$). Assume that the arrays `x` and `y` are of size N :

- a. `z = x + y`
- b. `x[5] = x[4]`
- c. `z = conj(x)`
- d. `z = angle(x)`
- e. `x = x[::-1]` (this reverses the order of elements).

Exercise 3.4.3

Write a Python function `uniquify_floats(x, epsilon)`, which accepts a list (or array) of floats `x`, and deletes all "duplicate" elements that are separated from another element by a distance of less than `epsilon`. The return value should be a list (or array) of floats that differ from each other by at least `eps`.

Exercise 3.4.4

(Hard) Suppose a floating-point representation uses one sign bit, N fraction bits, and M exponent bits. Find the density of real numbers which can be represented exactly by a floating-point number. Hence, show that floating-point precision decreases exponentially with the magnitude of the number.

This page titled [3.4: Exercises](#) is shared under a [CC BY-SA 4.0](#) license and was authored, remixed, and/or curated by [Y. D. Chong](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.