

In-class Exercises and Homework, Sept. 30

Please upload your assignment as a PDF or Word file to Canvas by Thursday, Sept. 15 at midnight. This assignment is worth 20 points as indicated.

Problem 1 (2 pts.) The IEEE-754 floating-point standard specifies that single-precision floating-point numbers be represented in a base-2 format with a 24-bit significand:

$$\hat{x} = \pm a_0.a_1a_2 \cdots a_{23} \times 2^e \quad (1)$$

We say that the nonzero quantity \hat{x} is normalized if $a_0 = 1$ and $-126 \leq e \leq 127$. We define the representation of zero as $0.00 \cdots 0 \times 2^0$

a) What is the normalized IEEE single-precision representation of the number 8.125? Express your answer in the form of Eq. (1). In other words, specify the values of $a_1, a_2, \cdots a_{23}$ and e .

b) If we change the significand of 8.125 by 1 ulp, by how much does the value of the floating-point representation change? Express your answer as a power of 2.

Problem 2 (3 pts.)

Let $x = 1/7$

a) Find the (infinite, eventually repeating) binary representation of x .

b) Find the IEEE-754 single-precision representation \hat{x} of x when rounding to nearest. That is, find the closest approximation to x by an expression of the form (1).

c) Use geometric series to find an exact expression for the error $\hat{x} - x$ due to the rounding in b).

Problem 3 (1 pts.)

What fraction is represented by the repeating decimal sequence $0.135\overline{135}$? Express your answer as p/q in lowest terms.

Problem 4 (2 pts.)

True or false: If x has a terminating base-2 expansion, then x has a terminating base-10 expansion. State a proof using a result given in lecture or find a counterexample.

Problem 5 (3 pts.)

The heart of many numerical methods, such as Newton's method, involves the solution of a linear system of equations. This problem illustrates some of the inherent mathematical problems that can exist in the solution of linear systems.

a) Let n be a positive integer and consider the linear system

$$\begin{aligned} x + y &= a \\ x + (1 + 2^{-n})y &= b \end{aligned}$$

Express x and y in terms of a , b , and 2^n .

b) Suppose that we are using IEEE single-precision arithmetic and that the value of b changes by 1 ulp. By how much do the values of x and y change?

This exercise illustrates the problem of ill conditioning in linear systems. Even if n is relatively modest, say $n = 10$, then if b is subject to roundoff error, or if the value of b is known only to a few bits of accuracy due to measurement error, then the values of x and y may have no significant digits (i.e., the computed values of x and y may be garbage). Linear systems that are nearly singular are always ill conditioned.

Problem 6 (4 pts.)

The machine epsilon for IEEE single-precision numbers is $2^{-23} \approx 1.2 \times 10^{-7}$. In this respect, single-precision IEEE floating-point is roughly equivalent to 7 decimal digits. Nevertheless, 7 decimal digits do not suffice to represent an IEEE single-precision number uniquely—in fact, you need 9 decimal digits. This exercise outlines a proof of this claim.

Consider real numbers x such that $10 \leq x < 16$, i.e., the interval $[10, 16)$.

a) The numbers in $[10, 16)$ that are exactly representable in 7 decimal digits are 10.00000, 10.00001, 10.00002, \dots , 15.99999. How many numbers are in this set?

b) The numbers in $[10, 16)$ that are exactly representable in IEEE single precision run from

$$10_{10} = 1.01 \underbrace{00 \cdots 00}_{21 \text{ bits}} \times 2^3 \quad \text{to} \quad (16 - 2^{-20})_{10} = 1.11 \underbrace{11 \cdots 11}_{21 \text{ bits}} \times 2^3$$

How many numbers are in this set?

c) Apply the pigeonhole principle to the example above to show that at least two different IEEE single-precision numbers have the same 7-digit decimal representation.

d) Explain why 8 decimal digits also are not enough to represent IEEE single-precision numbers uniquely.

Problem 7 (5 pts.)

The floating-point representation in Eq. (1) can be stored in 32 bits, as follows: 1 bit for the sign, 23 bits for a_1, a_2, \dots, a_{23} , and 8 bits for the exponent. There is no need to store a_0 since it is always 1 if $x \neq 0$.

However, Eq. (1) is not the only possible way to encode a floating-point number in 32 bits. Another approach, which we'll call the (\pm, s, l) format, uses a sign bit, a significand s , and an integer "level" l , as follows. Let s be expressed as a binary number such that $1 \leq s < 2$, and require that $-7 \leq l \leq 7$. We can express real numbers greater than or equal to 1 in magnitude in the following way. If $l = 0$, then the real number represented by $(\pm, s, 0)$ is $\pm s$. If $l = 1$, then $(\pm, s, 1)$ represents $\pm 2^s$; if $l = 2$, then $(\pm, s, 2)$ represents $\pm 2^{2^s}$; $l = 3$, $\pm 2^{2^{2^s}}$; and so on. Numbers between 0 and 1 in magnitude are represented by negative values of l : $(\pm, s, -1)$ represents $\pm 2^{-s}$; $(\pm, s, -2)$ represents $\pm 2^{-2^s}$, etc. The (\pm, s, l) format is just as compact as the IEEE format: l requires 4 bits, the sign takes 1 bit, and s can occupy the remaining bits of a 32-bit word.

a) What real number is represented by $(+, 1.5, 0)$? $(-, 1.5, 1)$? $(+, 1.5, 2)$? $(-, 1.5, -1)$?

b) What is the set of representable positive values for $l = 0, 1, 2, 3, 4$?

- c)** The number $G = 10^{100}$ is called a *googol*; 10^G is a *googolplex*. Express the largest representable value of $(+, s, 4)$ as an approximate power of G .
- d)** Is the largest representable value of $(+, s, 5)$ greater or less than a *googolplex*? Explain.
- e)** Invent your own terminology as necessary to describe the largest representable values of $(+, s, 6)$ and $(+, s, 7)$.

Although overflow and underflow are unlikely to occur in the (\pm, s, l) format, there are no simple algorithms for addition and multiplication, which is why we use formats like Eq. (1).