AMATH 583 Homework 2

Jesse Akes

2024-04-16

Problem 1

Write a C ++ program that finds a practical measure of your machine's SP (32 bit) and DP (64 bit) floating point precision by taking the difference of 2 numbers and comparing these to zero in the same precision. What value do you obtain for ϵ machine in both precisions?

Listing 1 `precision.cpp`

```cpp
#include <iostream>
#include <cmath>

int main() {
    // Compare SP
    float zero_sp = 0.0f;
    for (int i = 0; i < 32; ++i) {
        float out = (1.0f + 1.0f/std::pow(2.0f, float(i))) - 1.0f;
        std::cout << "Iteration: " << i << " SP: " << out << std::endl;
        if (out == 0.0f) {
            break;
        }
    }

    // Compare DP
    double zero_dp = 0.0;
    for (int i = 0; i < 64; ++i) {
        double out = (1.0 + 1.0/std::pow(2.0, double(i))) - 1.0;
        std::cout << "Iteration: " << i << " DP: " << out << std::endl;
        if (out == 0.0) {
            break;
        }
    }
}
```

For the single precision floating point, I found a vaue of 24 bits and for double precision, I found a value of 53 bits. This means that the machine precision ϵ for single precision is 2^{-23} and for double precision is 2^{-52}.

Problem 2

What are the largest and smallest SP (32 bit) and DP (64 bit) numbers that can be represented in IEEE floating point representation? Show work in terms of sign, mantissa, and exponent.

The IEEE standard for floating point numbers is

$$(-)^S \times M \times 2^E$$

where S is the sign bit, M is the significand or mantissa (23 bit for SP float or 52 for DP) and E is the exponent (8 bit for SP and 11 bit for DP). When normalized, $E = e - bias$ where the bias is $2^{k-1} - 1$ (k being the number of bits in the exponent). For SP, the bias is 127 and for DP the bias is 1023.

For 32 bit floating point representation, the largest number is represented by

- Sign bit: 0 (1 bit)
- Exponent: 11111110 (8 bits)
- Mantissa: 11111111111111111111111 (23 bits) This gives us the largest number as:

$$(2 - 2^{-23}) \times 2^{254-127} \approx 3.4028... \times 10^{38}$$

The smallest number is represented by the same thing, just with the sign bit set

- Sign bit: 1 (1 bit)
- Exponent: 11111110 (8 bits)
- Mantissa: 11111111111111111111111 (23 bits) This gives us the smallest number as:

$$(2 - 2^{-23}) \times -2^{254-127} \approx -3.4028... \times 10^{38}$$

For brevity, we have the same pattern with doubles, just more bits.

For 64 bit floating point representation, the largest number is represented by

- Sign bit: 0 (1 bit)
- Exponent: 11111111110 (11 bits)
- Mantissa: 111111111111111111111111111111111111111111 (52 bits) This gives us the largest number as:

$$(2 - 2^{52}) \times 2^{2046-1023} \approx 1.79769... \times 10^{308}$$

The smallest number is represented by the same thing, just with the sign bit set

- Sign bit: 0 (1 bit)
- Exponent: 11111111110 (11 bits)
- Mantissa: 111111111111111111111111111111111111111111 (52 bits) This gives us the smallest number as:

$$(2 - 2^{-52}) \times -2^{2046-1023} \approx -1.79769... \times 10^{308}$$

Problem 3

*Write a C++ program to multiply the integers 200 * 300 * 400 * 500 on your computer? What is the result? Name the effect you observe.*

Listing 2 multiply.cpp

```
#include <iostream>

int main() {
    std::cout << "200 * 300 * 400 * 500 = " << 200 * 300 * 400 * 500 <<
     ↪  std::endl;
    return 0;
}
```

This resulted in an integer overflow error.

Problem 4

Given C++ code segment below, what is the final value of counter?

Listing 3 unsigned.cpp

```
#include <iostream>

int main() {

    unsigned int counter = 0;
    for (int i = 0; i < 3; ++ i) --counter;

    std::cout << "Counter: " << counter << std::endl;
    return 0;
}
```

The final value of the counter is 4294967293. This is due to it being an unsigned integer and the decrement operation causing it to wrap around.

Problem 5

Count and report how many IEEE SP (32 bit) normalized and denormalized floating point numbers there are. Please count and label infinities and NANs as well. Show work.

Like we saw before, a 32 bit floating point number is composed of the following parts:

- Sign bit (1 bit)
- Exponent (8 bits)
- Mantissa/Significand (23 bits)

A noramlized floating point number does not contain all zeros or all ones in the exponent, while a denormalized number does contain all zeros.

Let's look at the easy cases first:

- Infinity is reserved as exponenet all 1's and Significand all 0's. With the sign bit, we have two choices: $\pm\infty$.
- NaN is reserved as an exponent of all 1's, then some non-zero number in the significand. Thus there are $2 * 2^{23} - 2 = 16777214$ options for Nan (sign bit has 2 options, 23 bits in the significand, and subtract out the two zero cases)

For normalized and denormalized numbers, we can follow a similar pattern:

- Normalized numbers have $2 * (2^8 - 2) * 2^{23} = (2^9 - 2^2) * 2^{23} = 2^{32} - 2^{25}$ options (sign bit has 2 options, 8 bits in the exponent, less the all 0 and all 1 cases reserved for zero and infinity/NaN, and 23 bits in the significand)
- Denormalized numbers have $2 * 2^{23} - 2 = 16777214$ options for denormalized numbers (sign bit has 2 options, all 0's for the exponent, and then 23 bits for the siginificand all less the two options for 0)

Problem 6

Consider a 6 bit floating point system with $s = 1$ (1 sign bit), $k = 3$ (3 bit exponent field), and $n = 2$ (2 bit mantissa).

With $k = 3$ bits in the exponent, we have a bias of $2^{k-1} - 1 = 2^2 - 1 = 3$

a) *Calculate by hand all the representable normalized numbers. Show work* $2*(2^3-2)*2^2 = (2^4-2^2)*2^2 = 2^6-2^4 = 48$ possible normalized numbers.

Normalized values are represented as $v = (-)^s(1+f)2^{e-bias}$

Sign	Exponent	Mantissa	Value
0	001	00	$1*(1+0)*2^{1-3} = 0.25$
0	010	00	$1*(1+0)*2^{2-3} = 0.5$
0	011	00	$1*(1+0)*2^{3-3} = 1$
0	100	00	$1*(1+0)*2^{4-3} = 2$
0	101	00	$1*(1+0)*2^{5-3} = 4$
0	110	00	$1*(1+0)*2^{6-3} = 8$
0	001	01	$1*(1+0.25)*2^{1-3} = 0.3125$
0	010	01	$1*(1+0.25)*2^{2-3} = 0.625$
0	011	01	$1*(1+0.25)*2^{3-3} = 1.25$
0	100	01	$1*(1+0.25)*2^{4-3} = 2.5$
0	101	01	$1*(1+0.25)*2^{5-3} = 5$
0	110	01	$1*(1+0.25)*2^{6-3} = 10$
0	001	10	$1*(1+0.5)*2^{1-3} = 0.375$
0	010	10	$1*(1+0.5)*2^{2-3} = 0.75$
0	011	10	$1*(1+0.5)*2^{3-3} = 1.5$
0	100	10	$1*(1+0.5)*2^{4-3} = 3$
0	101	10	$1*(1+0.5)*2^{5-3} = 6$
0	110	10	$1*(1+0.5)*2^{6-3} = 9$
0	001	11	$1*(1+0.75)*2^{1-3} = 0.4375$
0	010	11	$1*(1+0.75)*2^{2-3} = 0.875$
0	011	11	$1*(1+0.75)*2^{3-3} = 1.75$
0	100	11	$1*(1+0.75)*2^{4-3} = 3.5$
0	101	11	$1*(1+0.75)*2^{5-3} = 7$
0	110	11	$1*(1+0.75)*2^{6-3} = 14$
1	001	00	$-1*(1+0)*2^{1-3} = -0.25$
1	010	00	$-1*(1+0)*2^{2-3} = -0.5$
1	011	00	$-1*(1+0)*2^{3-3} = -1$
1	100	00	$-1*(1+0)*2^{4-3} = -2$
1	101	00	$-1*(1+0)*2^{5-3} = -4$
1	110	00	$-1*(1+0)*2^{6-3} = -8$
1	001	01	$-1*(1+0.25)*2^{1-3} = -0.3125$
1	010	01	$-1*(1+0.25)*2^{2-3} = -0.625$
1	011	01	$-1*(1+0.25)*2^{3-3} = -1.25$
1	100	01	$-1*(1+0.25)*2^{4-3} = -2.5$
1	101	01	$-1*(1+0.25)*2^{5-3} = -5$
1	110	01	$-1*(1+0.25)*2^{6-3} = -10$
1	001	10	$-1*(1+0.5)*2^{1-3} = -0.375$

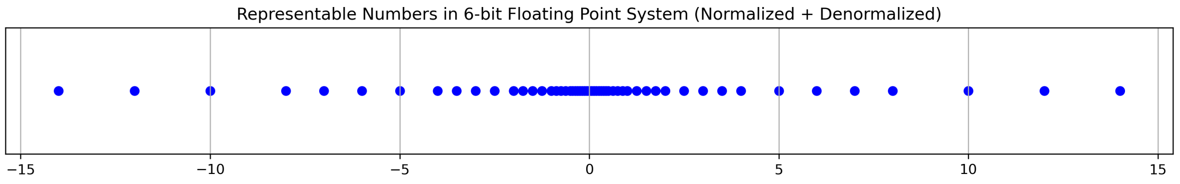
Sign	Exponent	Mantissa	Value
1	010	10	$-1 * (1 + 0.5) * 2^{2-3} = -0.75$
1	011	10	$-1 * (1 + 0.5) * 2^{3-3} = -1.5$
1	100	10	$-1 * (1 + 0.5) * 2^{4-3} = -3$
1	101	10	$-1 * (1 + 0.5) * 2^{5-3} = -6$
1	110	10	$-1 * (1 + 0.5) * 2^{6-3} = -9$
1	001	11	$-1 * (1 + 0.75) * 2^{1-3} = -0.4375$
1	010	11	$-1 * (1 + 0.75) * 2^{2-3} = -0.875$
1	011	11	$-1 * (1 + 0.75) * 2^{3-3} = -1.75$
1	100	11	$-1 * (1 + 0.75) * 2^{4-3} = -3.5$
1	101	11	$-1 * (1 + 0.75) * 2^{5-3} = -7$
1	110	11	$-1 * (1 + 0.75) * 2^{6-3} = -14$

b) *Calculate by hand all the representable denormalized numbers. Show work.* $2 * 2^2 - 2 = 6$ possible denormalized numbers

Since denormalized numbers have exponents of all 0, we have many fewer numbers we can represent. And the value is represented as $v = (-)^s * f * 2^{1-bias}$

Sign	Exponent	Mantissa	Value
0	000	01	$1 * \frac{1}{4} * 2^{-2} = \frac{1}{16} = 0.0625$
1	000	01	$-1 * \frac{1}{4} * 2^{-2} = -\frac{1}{16} = -0.0625$
0	000	10	$1 * \frac{1}{2} * 2^{-2} = \frac{1}{8} = 0.125$
1	000	10	$-1 * \frac{1}{2} * 2^{-2} = -\frac{1}{8} = -0.125$
0	000	11	$1 * \frac{1}{2} + \frac{1}{4} * 2^{-2} = \frac{3}{4} = 0.1875$
1	000	11	$-1 * \frac{1}{2} + \frac{1}{4} * 2^{-2} = -\frac{3}{4} = -0.1875$

c) *Plot both sets of numbers (ignoring NANs and infinities) as a number line to see the gaps of (un)representable numbers.*



Problem 7

Conversions. Show work.

a) *Write* $(D3B701)_{16}$ *as an integer in base-10.* $(13 \times 16^5) + (3 \times 16^4) + (11 \times 16^3) + (7 \times 16^2) + (0 \times 16^1) + (1 \times 16^0) = (13874945)_{10}$

b) *Write* $(1010000100111111)_2$ *as an integer in base-16, i.e. as a hexadecimal* Break into 4 bit intervals and convert each interval: $1010000100111111 \rightarrow 10\ 1\ 3\ 15 \rightarrow (A13F)_{16}$

Problem 8

Are there $a, b, c \in \mathbb{Z}$ *s.t.*$6a + 9b + 15c = 107$ *? Show work.*

To do this, we need to find the greatest common divisor of the coefficients.

$$gcd(6, 9, 15) = 3$$

If we factor both sides by the gcd:

$$3(2a + 3b + 5c) = \frac{107}{3}$$

Since $\frac{107}{3}$ is not an integer, there is no way for these coefficients to be combined to make 107.

Problem 9

Equivalence classes modulo n. $\forall\ a, b \in Z$ *then* $a \equiv b(mod\ n)$ *means* $n|(a - b)$ *or* $a = b + k \cdot n$ *and* $k \in \boldsymbol{Z}$. $\boldsymbol{Z}_n$ *is the set of equivalence classes* $[0], [1], ..., [n - 1]$. *Is* $(\boldsymbol{Z}_n, +, \cdot)$ *a ring? Hint: If* $s \in [i]$, *then* $n|(s - i)$. *Show work (use ring properties).*

To be a ring, the set must satisfy all properties of addition and multiplication:

1. Addition:
   - associative
   - commutative
   - have the additive identity
   - have the additive inverse
2. Multiplication
   - distributive

This is all I remember about rings...