Computational Physics
Homework 2

Shalma Wegsman

October 2020

Problem 1: Floating Point Binary

(A)

We want to convert our floating point number into scientific notation in base 2:

$$A = 3.4657 \times 10^{25}$$

$$\log_2 A = \log_2(3.4657 \times 10^{25})$$

$$\log_2 A = \log_2(3.4657) + \frac{\log_{10}(10^{25})}{\log_{10}(2)}$$

Where

$$\frac{\log_{10}(10^{25})}{\log_{10}(2)} \approx 83$$

Using this as a guideline, we can divide A by 2^{83} to get ≈ 3.583450, so we can write:

$$A = 3.583450 \times 2^{83} = 1.791725 \times 2^{84}$$

Looking at this, we see that $s = 0$ (since A is positive), $f = 0.791725$, and $84 = e - 127 \Rightarrow e = 211$.

(B)

Now we want to convert s, f and e to binary. We trivially have that $\boxed{s = 0}$.

$$e = 211 = 128 + 64 + 16 + 2 + 1 = 2^7 + 2^6 + 2^4 + 2^1 + 2^0$$

$$\Rightarrow \boxed{e = 11010011}$$

$$2f = \mathbf{1}.58345$$

$$2(0.58345) = \mathbf{1}.1669$$

$$2(0.1669) = \mathbf{0}.3338$$

and so on gets us to:

$$\Rightarrow \boxed{f = 0.110010101010111101}$$

Problem 2: User-Defined Float

(A)

Consider a 32-bit float where we have 12 bits for the exponent but only 19 bits for the mantissa. We want to find the largest normal number and the smallest (positive) number that can be stored. We can see that:

$$x_{\text{max}} = 0|1111\ 1111\ 1110|1111\ 1111\ 1111\ 1111\ 111$$

where the vertical line demarcates $s|e|f$. Note that the zero at position 20 is there to avoid overflow. We can convert e to decimal:

$$e = \sum_{i=1}^{11} 2^i = \frac{2^{12} - 1}{2 - 1} - 1 = 4094$$

and we have

$$\sum_{i=0}^{18} f_i \times 2^{-(19-i)} = 2^{-19} \sum_{i=0}^{18} 2^i = \frac{2^{19} - 1}{2^{19}} \approx 1$$

So we can now rewrite x_{max} as:

$$x_{\text{max}} = \big[1 + 1\big] \times 2^{4094-2047} = \boxed{2^{2048}}$$

Similarly, we can find the minimum number you can store:

$$x_{\text{min}} = 0|0000\ 0000\ 0000|0000\ 0000\ 0000\ 0000\ 001$$

Here, the one in position zero is to avoid subnormal numbers. Then $s = e = 0$, and:

$$\sum_{i=0}^{18} f_i \times 2^{-(19-i)} = 2^{-19}$$

So we have:

$$x_{\text{min}} = \big[0 + 2^{-19}\big] \times 2^{-2046} = \boxed{2^{-2065}}$$

(B)

Now we want to find the machine precision for part (A). This is determined by the place value of the lowest bit in the mantissa, which in this case is 19. And so:

$$\sigma_x = 2^{-19}$$

(C)

This part is done in the jupyter notebook.

Problem 3: Quadratic Equations

(A)

This part is done in the jupyter notebook.

(B)

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \times \left(\frac{-b \mp \sqrt{b^2 - 4ac}}{-b \mp \sqrt{b^2 - 4ac}}\right)$$

Case 1:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \times \left(\frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}}\right) = \frac{b^2 - (b^2 - 4ac)}{2a(-b - \sqrt{b^2 - 4ac})} = \frac{2c}{-b - \sqrt{b^2 - 4ac}}$$

Case 2:

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \times \left(\frac{-b + \sqrt{b^2 - 4ac}}{-b + \sqrt{b^2 - 4ac}}\right) = \frac{b^2 - (b^2 - 4ac)}{2a(-b + \sqrt{b^2 - 4ac})} = \frac{2c}{-b + \sqrt{b^2 - 4ac}}$$

So we can write solutions to the equation $ax^2 + bx + c = 0$ as

$$\boxed{x = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}}}$$

(C)

This part is done in the jupyter notebook.

Problem 4: Object-oriented Programming

This problem is done in the jupyter notebook.