homework2

baibizhe

May 25, 2020

1(f)

```
a = smallest_positive  ## smallest_postive  is too long to type in
exmaple1f = ([0, 0, 1, 0, 0], 1, 1)  # the original number
print(add_float(add_float(add_float(add_float(exmaple1f, a), a), a), a))
# use (((1+a)+a)+a)+a) method
suma = ([0, 0, 0, 0, 0], 0, 1)
for i in range(4):
    suma = add_float(suma, a)
print(add_float(exmaple1f, suma))  # use (1+(a+(a+(a+(+a)))) method
>>> ([0, 0, 1, 0, 0], 1, 1)
>>> ([0, 0, 1, 0, 1], 1, 1)
```

From the last two line , we can see they are not the same . Then , add_float is not associative

2:

let x be -0.01 and n = 15 . then there is a catastrophic cancellation
for list h1(-0.01,15):
[1.0, -0.01, 0.0001, -1.0000000000000002e-06, 1e-08, -1.0000000000000002e-10, 1.0000000000000002e-12, -1.0000000000000002e-14, 1.0000000000000002e-16, -1.0000000000000003e-18, 1.0000000000000002e-20, -1.0000000000000003e-22, 1.0000000000000003e-24, -1.0000000000000003e-26, 1.0000000000000003e-28]
When we add them each time one more items , we get :
0, 1.0, 0.99, 0.9901, 0.990099, 0.99009901, 0.9900990099, 0.990099009901, 0.99009900990099, 0.9900990099009901, 0.9900990099009901, 0.9900990099009901 .....
we can see the summation are not changing anymore , so there is catastrophic cancellation

for list h2(-0.01,15):
[1.0, -0.01, 5e-05, -1.666666666666667e-07, 4.166666666666667e-10, -8.333333333333335e-13, 1.3888888888888892e-15, -1.9841269841269846e-18, 2.4801587301587306e-21, -2.7557319223985896e-24, 2.7557319223985897e-27, -2.5052108385441727e-30, 2.0876756987868105e-33, -1.605904383682162e-36, 1.1470745597729728e-39]
When we add them each time one more items , we get :
0, 1.0, 0.99, 0.99005, 0.9900498333333333, 0.99004983375, 0.9900498337491667, 0.9900498337491681, 0.9900498337491681, 0.9900498337491681, 0.9900498337491681, ...
we can see the summation are not changing anymore , so there is catastrophic cancellation
So for x = -0.01 both h1() and h2() suffer from catastrophic cancellation

3.

S is sign E is exponent

a = 2^{53} = S(0) E(1000 0110 100) M(00000000000000000000000000000000000000000000000000000000)

b = 5 = S(0) E(1000 0000 001) M(01000000000000000000000000000000000000000000000000000000)

c = 10 = S(0) E(1000 0000 010) M(0100 00000000 00000000 00000000 00000000 00000000 00000000)

$E(10000110100)_2 = 1076_{10}$. minus 1023 is 53

$E(10000000001)_2 = 1025_{10}$. minus 1023 is 2

$E(10000000010)_2 = 1026_{10}$. minus1023 is 3

a = 1.0 x $2^5 3$

$b = (1.01)_2 * 2^2 = (1.25)_{10} * 2^2 = (1.01e-51)_2 * 2^{53}$

c = $1.01_2 * 2^3 = (1.01e-50)_2 * 2^{53}$

proof of rounding error in 2^{53}+5

a +b 2^{53}+5 : mantissa

1.00000000000000000000000000000000000000000000000000|52index 0

0.00000000000000000000000000000000000000000000000010|52index 1

=1.00000000000000000000000000000000000000000000000010|52index 1

round to even . the last digit in mantissa(index 52) is 0 , then round down

=1.00000000000000000000000000000000000000000000000010

there is rounding error with 2^{53}+5

when add 5 again , then there is a rounding error again

proof of no rounding in 2^{53}+10

a+c : 2^{53}+10 mantissa

1.00000000000000000000000000000000000000000000000000|52index 0

0.00000000000000000000000000000000000000000000000101|52index 0

=1.00000000000000000000000000000000000000000000000101

there is no rounding error with 2^{55}+10

The same problem is also happened in 2^{55} and 2^{56}