

Проблемы исчисления и один из способов их решения на языке Python

Доброго времени суток. Как говорят, краткость — сестра таланта, поэтому статья будет ясной и максимально информативной.

Введение

Когда начал изучать Python, я наткнулся на очень любопытную, раннюю неизвестную мне тему 'Как устроены вещественные числа'. В ней говорилось про мантиссу, экспоненту и прочие детали вроде отведения битов на хранение числа.

Мантисса

↓

$$1.111 = 1111 * 10^{-3}$$

↑

Экспоненциальная запись

Оказывается, компьютеру не составляет никаких проблем посчитать целые числа, но, когда мы доходим до вещественных, начинается интересная часть.

Суть проблемы

Из-за способа представления чисел с плавающей точкой операции с ними имеют не абсолютную точность. В этот момент меня, как новичка, просто поразило, как самая точная, совершенная машина не в состоянии посчитать точно даже банальные вещественные числа. Тем более мы все знаем, на что способны компьютеры в наше время, неужели такое может быть? Да, действительно, программа не в состоянии сложить два вещественных числа, которые без труда посчитает школьник в пятом классе:

```
x = 0.1
y = 0.2
print(x + y)  # 0.30000000000000004
```

Ну, почти точно, скажете вы. Кому важен 17 знак после запятой (иногда важен). Сейчас представьте, что вам нужно посчитать не два числа, а допустим двадцать, и подумайте, какая погрешность будет из-за того, что она будет возрастать каждый раз в геометрической прогрессии. При изучении языка встречались упоминание о библиотеках вроде NumPy, которые, как я думал, решают этот вопрос. Решают ли? Спойлер: нет.

Решение

Как решить эту проблему мне пришло в голову следующим образом: надо всего лишь заставить программу считать такие числа как человек. То есть оперировать строго целыми числами. Ведь мы делаем так же.

На бумаге идея представлялась довольно легко, осталось только реализовать практически.

При написании кода возникали ошибки различного типа, но в итоге получилось создать именно то, что и хотелось — функцию, которая будет складывать числа как человек, то есть, опять повторяю, оперирую только целыми числами.

[Ссылка на код репозиторий GitHub](#)

Общий итог

Любой учитель математики скажет вам, что точность — самое главное в исчислении. По моему мнению, в программировании она не менее важна. Инженерное дело, фотонная, атомная физика, биология — всем этим областям важная самая максимальная точность, где даже незначительная погрешность приведет к неверным результатам.

Хочу представить результаты работы функции на примере с обычным сложением в Питоне

```
plus_sum = plus(52.92, 20.99, 0.10)
print(float('{:.21f}'.format(plus_sum))) # 74.01

python_sum = 52.92 + 20.99 + 0.10
print(float('{:.21f}'.format(python_sum))) # 74.009999999999999

python_sum = 0.1 + 0.2
print(python_sum == 0.3) # False

plus_sum = plus(0.1, 0.2)
print(plus_sum == 0.3) # True
```

Да, конечно, этот алгоритм не решает всю проблему целиком и работает только с положительными числами на сложение, плюс медленнее обычных операций.

Зато он решает хоть только одну, но такую важную задачу как наиточнейшее сложение вещественных чисел. Мне хотелось придумать одно из возможных решений проблемы и внести вклад в общее дело

Надеюсь, в будущем, этот пример поспособствует развитию устранения погрешностей.

С радостью приму к сведению все отзывы и замечания.

Спасибо за отведенное время.