

## Отчет по 5ой лабе ассемблера

Код ко всему и графики: <https://github.com/leeiozh/ass3sem/tree/master/5lab>

1. см код

2. +inf = 01111111100000000000000000000000

-inf = 11111111100000000000000000000000

0 = 00000000000000000000000000000000

-0 = 10000000000000000000000000000000

NaN = 11111111100000000000000000000000

3. положили в переменную 0.2

4. к очень большому прибавили очень маленькое и наоборот

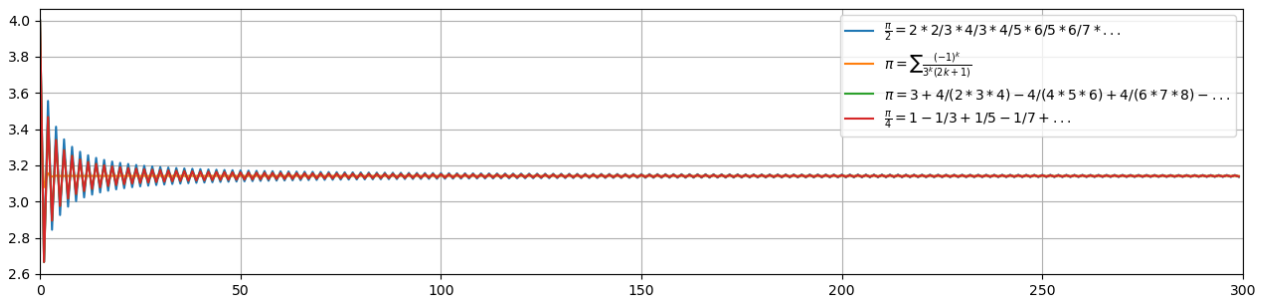
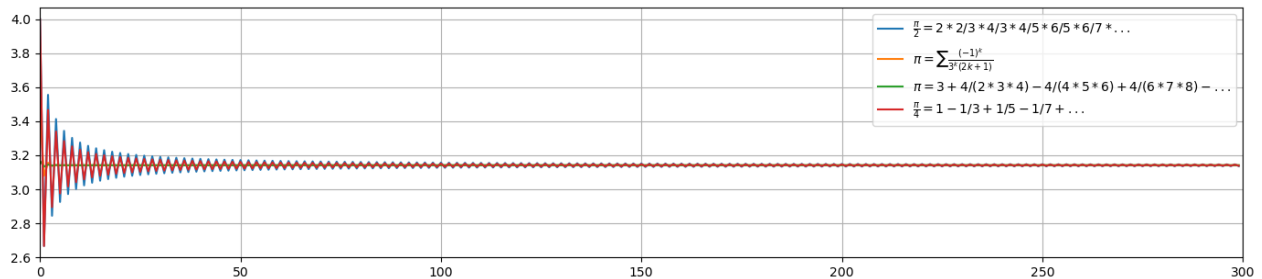
5. за операции отвечают add, sub, mul, div соответственно

префиксы разные в зависимости от типа (у float - s, у double – d)

работает в регистрах xmm0 и xmm1

6. сверху double, снизу float, разница минимальна даже на 1e5 итераций, но дабл точнее

расходящуюся формулу не нашла(



## Отчет по бой лабе ассемблера

Код ко всему и графики: <https://github.com/leeiozh/ass3sem/tree/master/6lab>

1. по операциям см 5 пункт 5 лабы, по инициализации см картинку

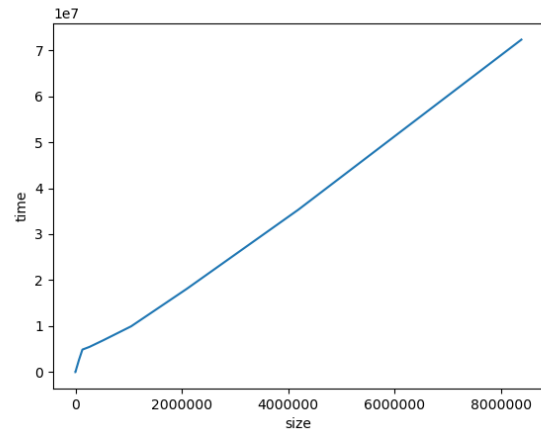
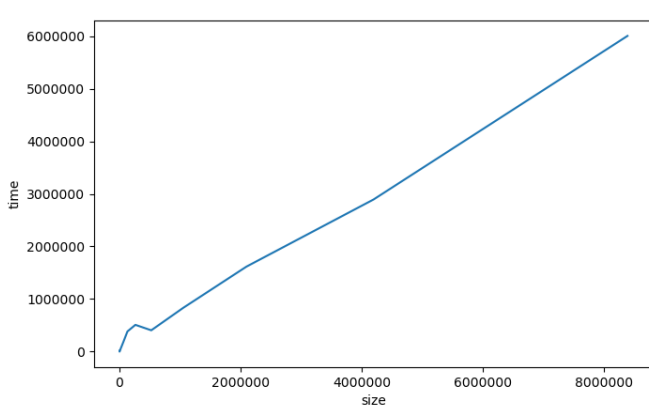
2. я сравнила своё суммирование четверками с numeric accumulate слева мой график, справа std::accumulate по неведомым мне причинам на 16 элементах он работает в 32 раза дольше, чем должен, но в остальном побеждает сильно

масштаб линейный, у моего наклон 0.7, у штатного 8  
дополнительно масштаб логарифмический лежит на гите

```
.long 1092616192
.globl b
.align 4
.type b, @object
.size b, 4

b:
.long 1085276160
.globl d
.align 8
.type c, @object
.size c, 8

c:
```



3. я ошибалась в п.6 – расходится четвертая (по условия пятая) формула  
сверху график в линейном масштабе (здесь обрезаны синий и красный графики), снизу в логарифмическом

