HashLab

December 18, 2020

[1]:
```
from IPython.display import IFrame
IFrame("task.pdf", width=900, height=800)
```

[1]:
```
<IPython.lib.display.IFrame at 0x7fa6a4bd3c10>
```

1

1.1 1.1

[10]:
```
%%latex

a)             $h_a : \mathbb{Z} \rightarrow \mathbb{Z}_m$,
  $a \in \mathbb{Z}_m$   $h_a(x) = (x + a) \; mod \; m$.      ,
↪       , . .
                                                          .
       ,   ,      ,        $x$   $x+1$                              .
```

a) $h_a : \mathbb{Z} \rightarrow \mathbb{Z}_m, \quad a \in \mathbb{Z}_m \quad h_a(x) = (x + a) \; mod \; m.$, ,
. . . , , ,
x $x + 1$.

b) ...

1.2 1.2

: , , . , ‘ ’
- , .

2 2 -

2.1 2.1

IHashSet inc/hashset. - inc/hash(
openssl).

- : - - - - -

- : - md5 - sha256 - murmur3

(std::hash . . gcc murmur).

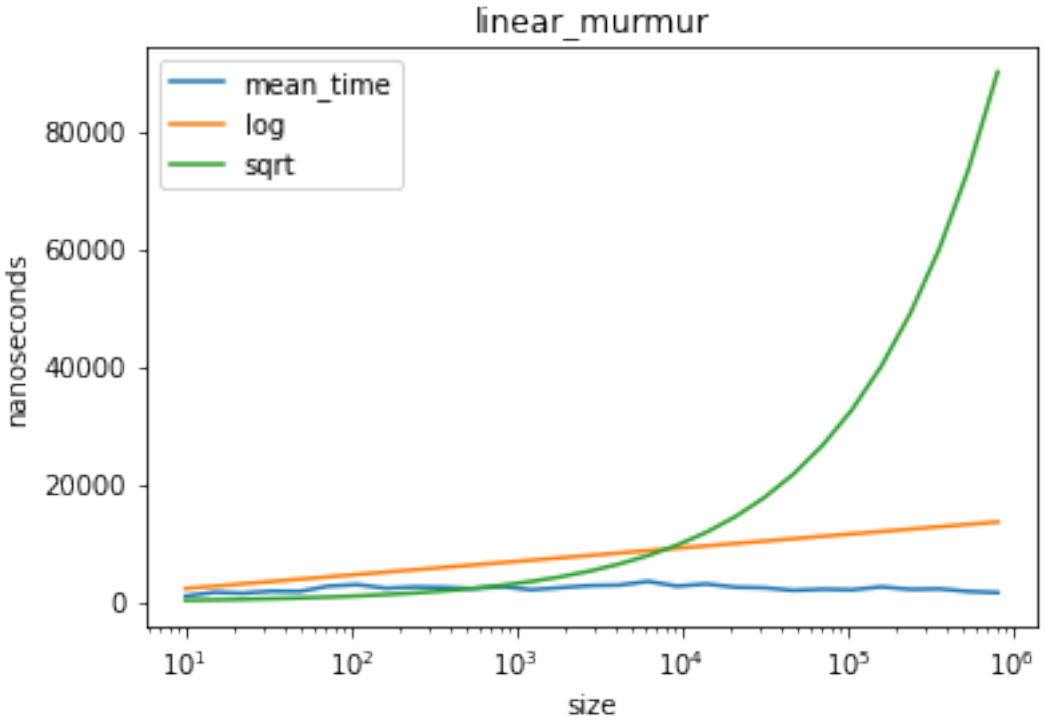
2.2 2.2

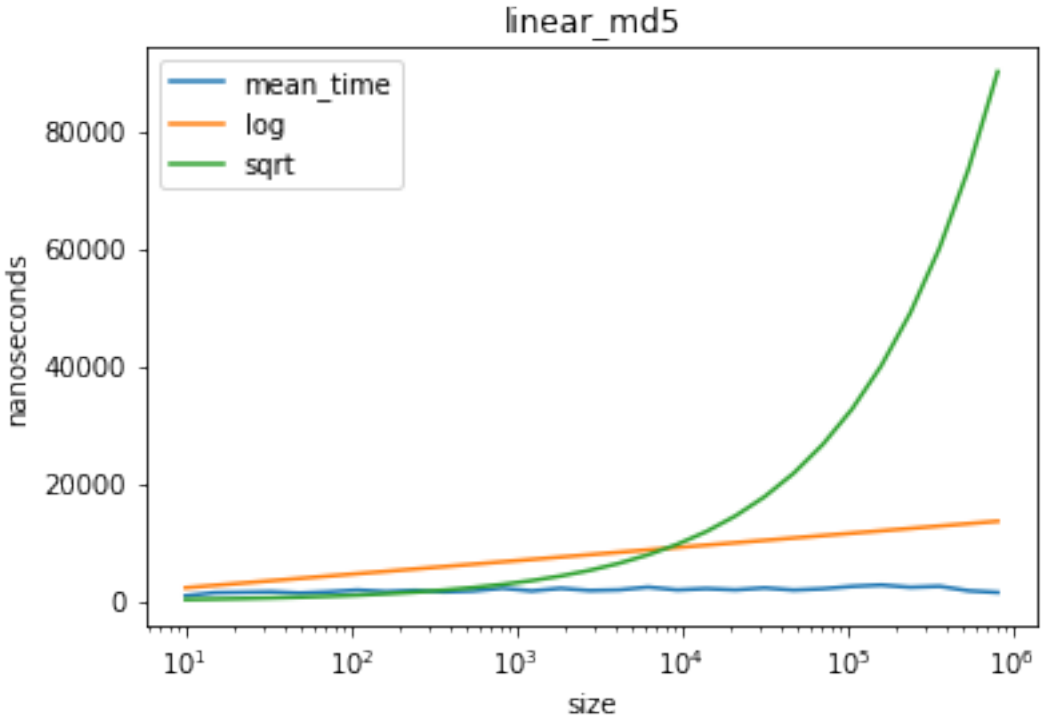
- 10^6. - 370 ,
. hash-lab.cpp. csv
data.

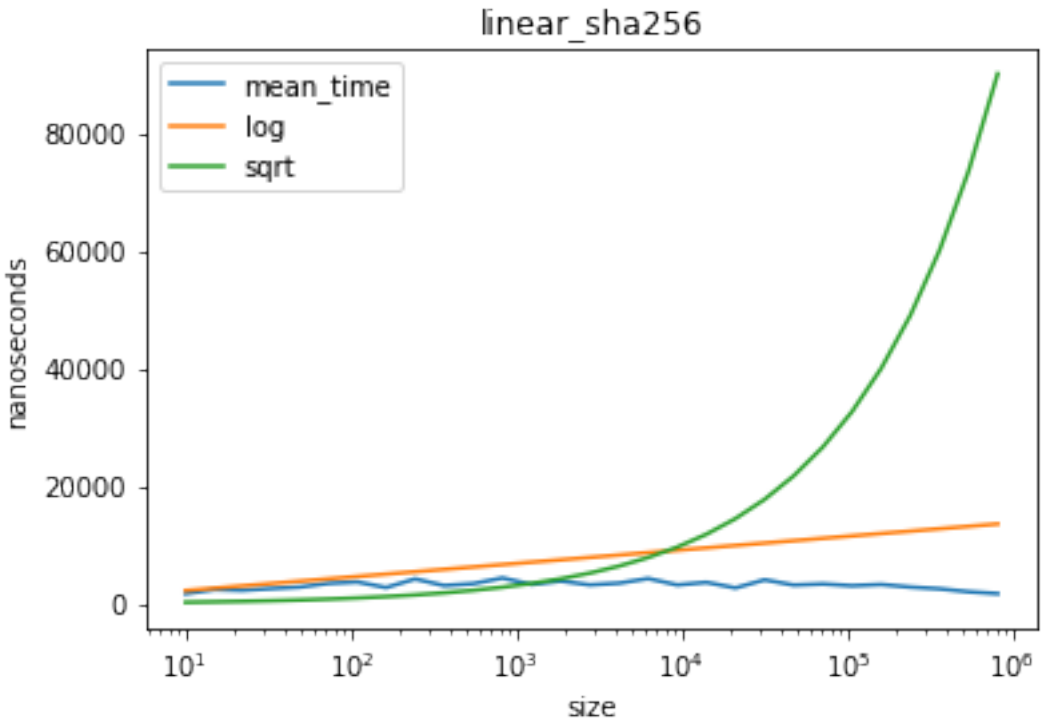
[3]:
```
from math import log, sqrt

def visualize(data, name):
    data["log"] = data["size"].map(log) * 1000 # some scaling
    data["sqrt"] = data["size"].map(sqrt) * 100 # to fit

    plt = data.plot(x="size", y=["mean_time", "log", "sqrt"])
    plt.set_ylabel("nanoseconds")
    plt.set_xscale("log")
    plt.set_title(name)
```

[4]:
```
import pandas as pd
from glob import glob
from pathlib import Path

def process(table):
    for fname in glob(f"data/{table}*.csv"):
        data = pd.read_csv(fname, sep='\t')
        name = Path(fname).stem
        visualize(data, name)
```

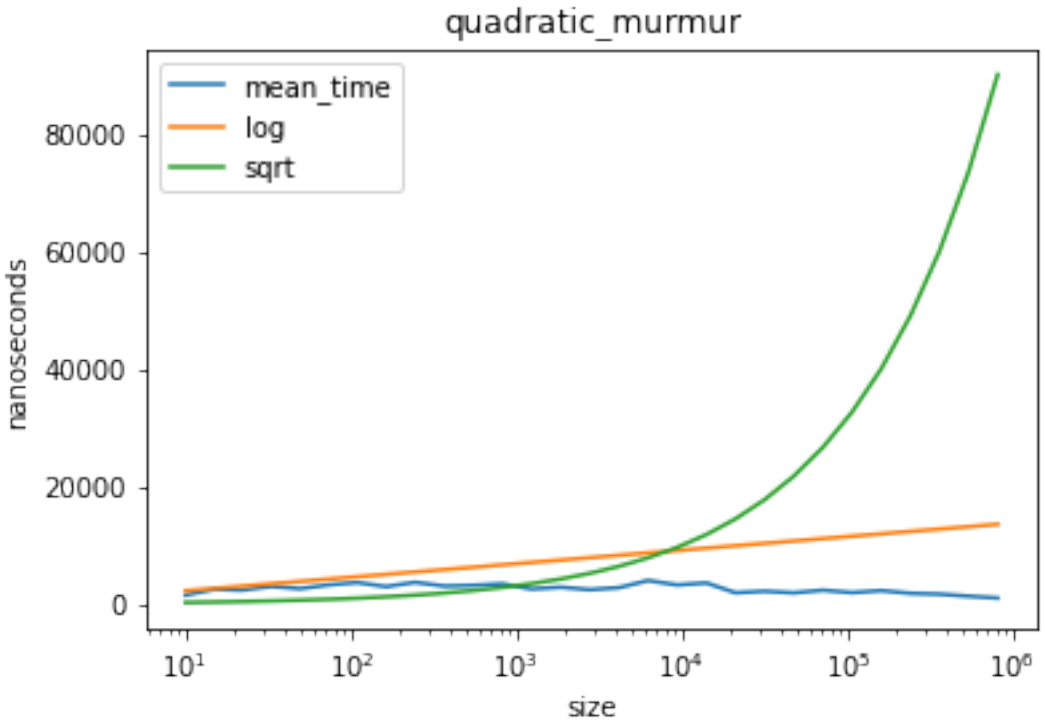
[5]:
```
process("linear")
```

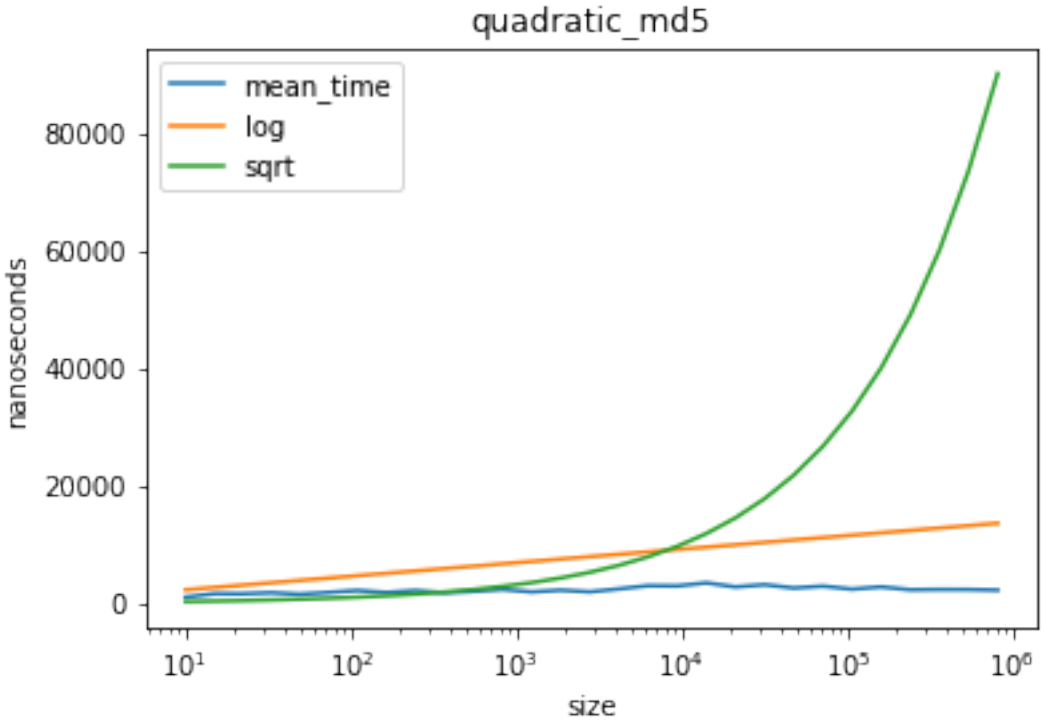

linear_murmur
mean_time
log
sqrt
80000
60000
40000
20000
0
nanoseconds
10^1
10^2
10^3
10^4
10^5
10^6
size

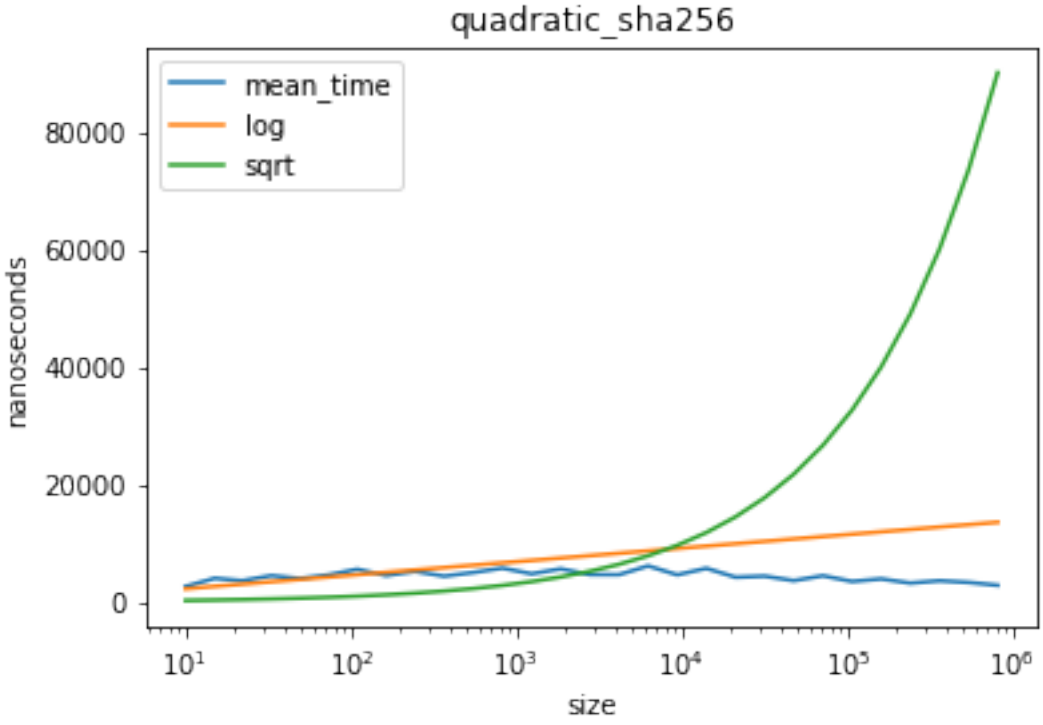

linear_md5
mean_time
log
sqrt
80000
60000
40000
20000
0
nanoseconds
10^1
10^2
10^3
10^4
10^5
10^6
size



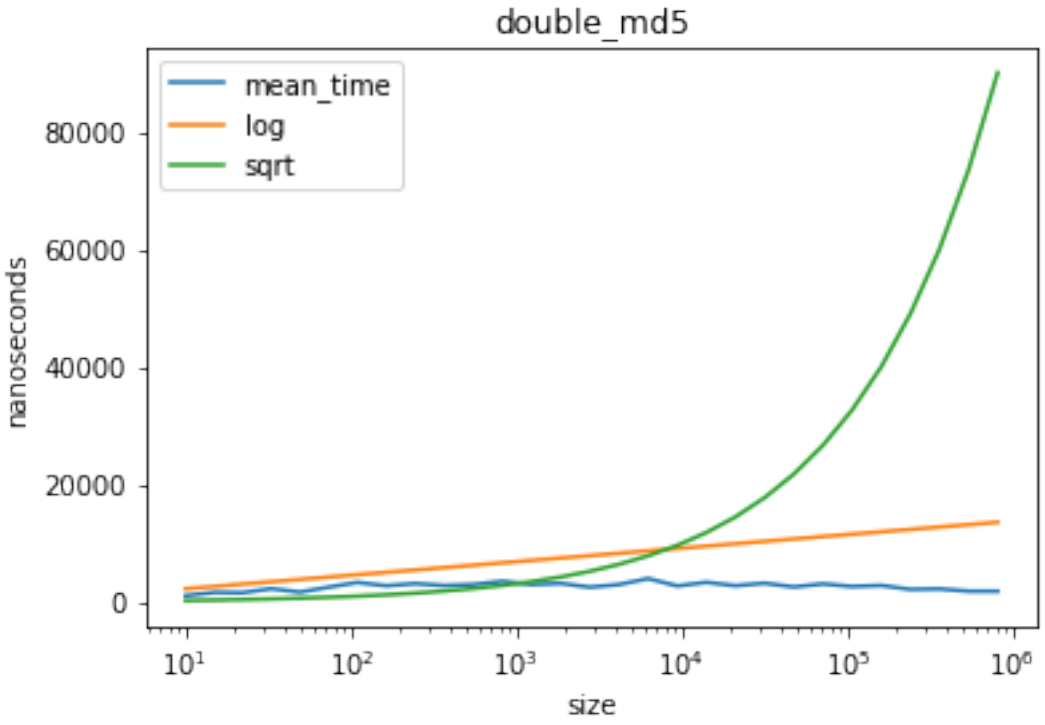
[6]:
```
process("quadratic")
```

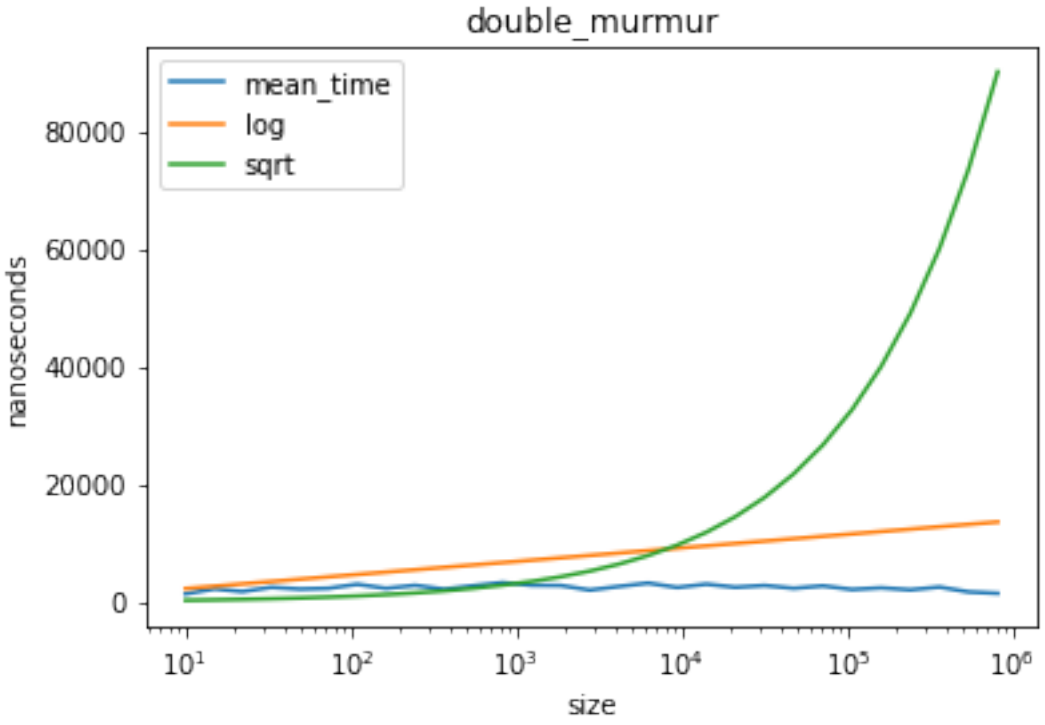


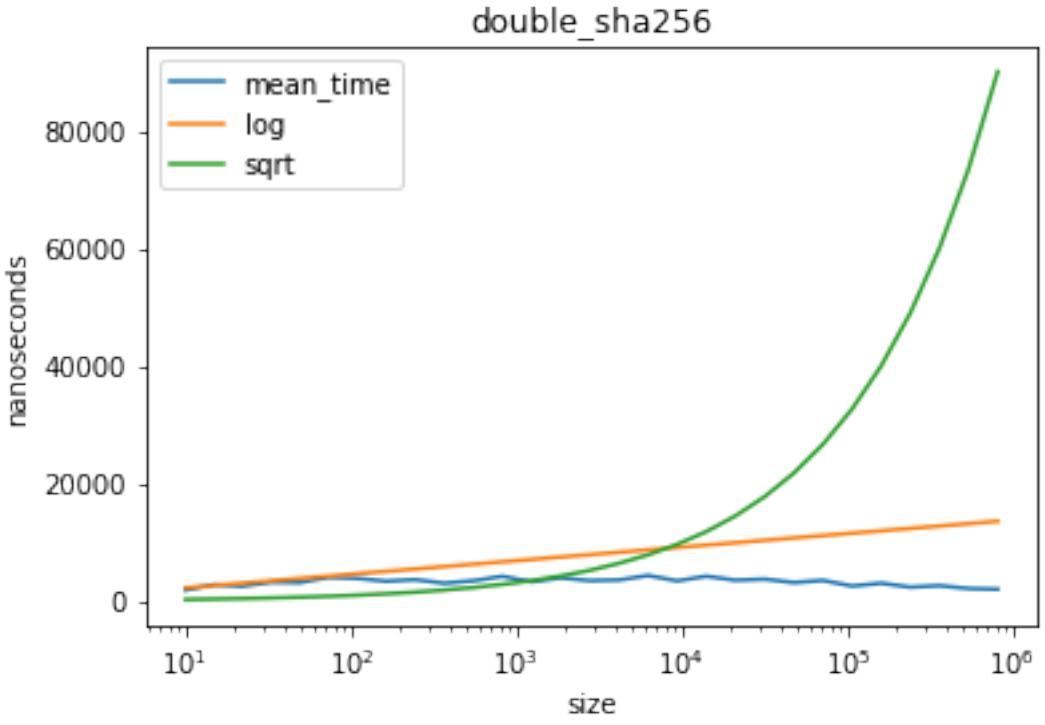

quadratic_md5
mean_time
log
sqrt
80000
60000
40000
20000
0
nanoseconds
10^1
10^2
10^3
10^4
10^5
10^6
size


quadratic_sha256
mean_time
log
sqrt
80000
60000
40000
20000
0
nanoseconds
10^1
10^2
10^3
10^4
10^5
10^6
size

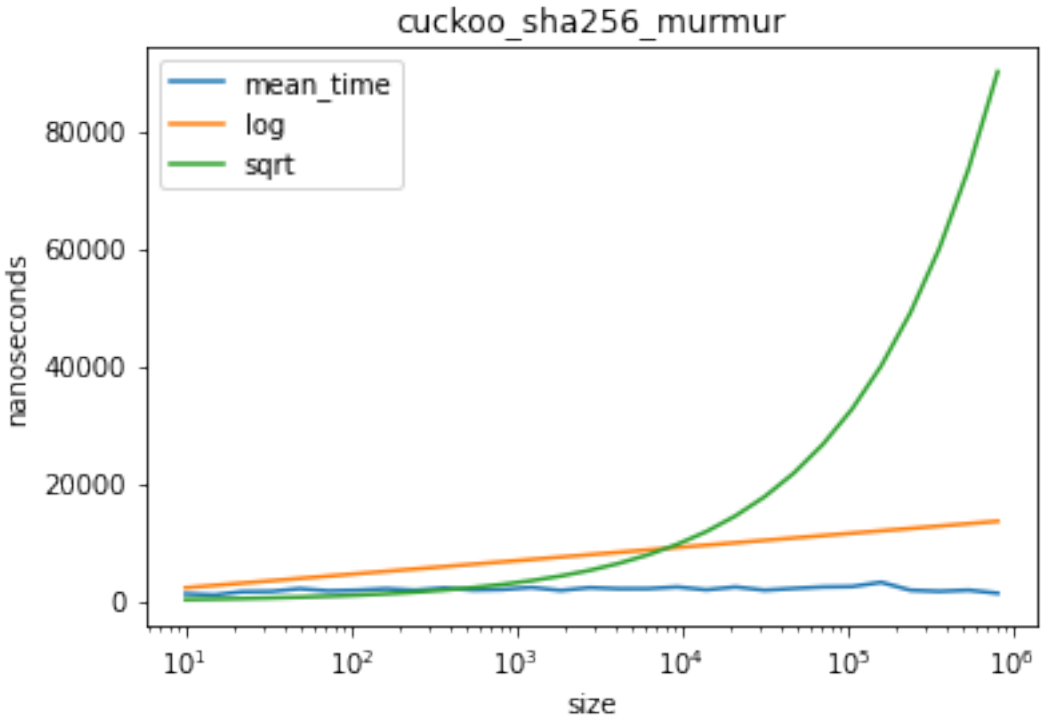
[7]:
```
process("double")
```

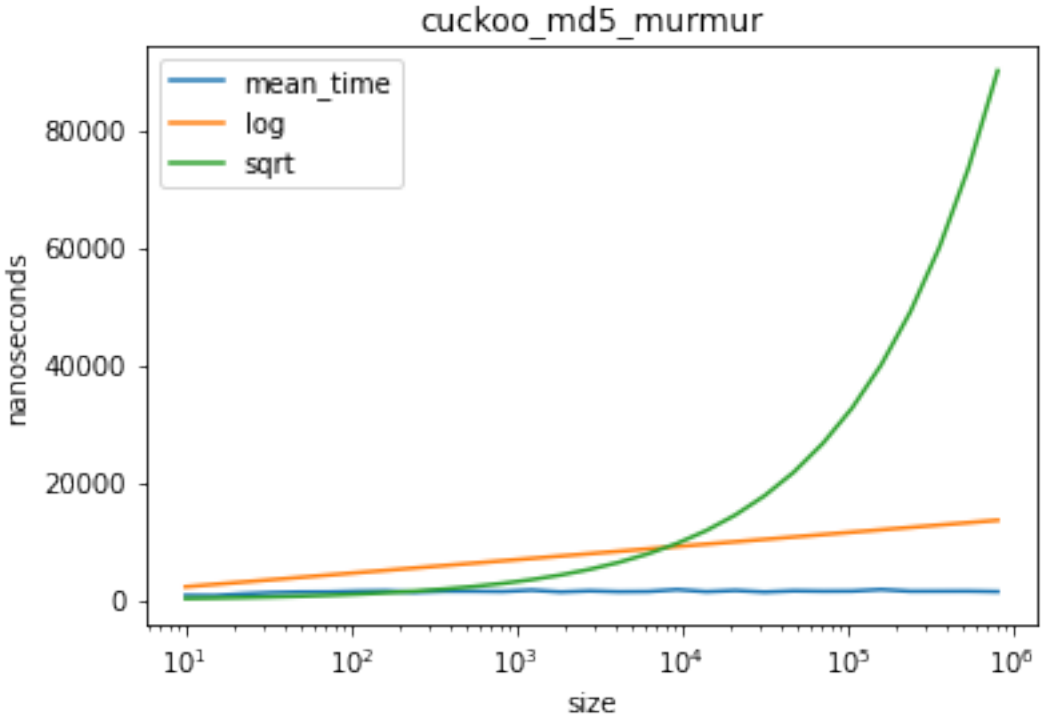


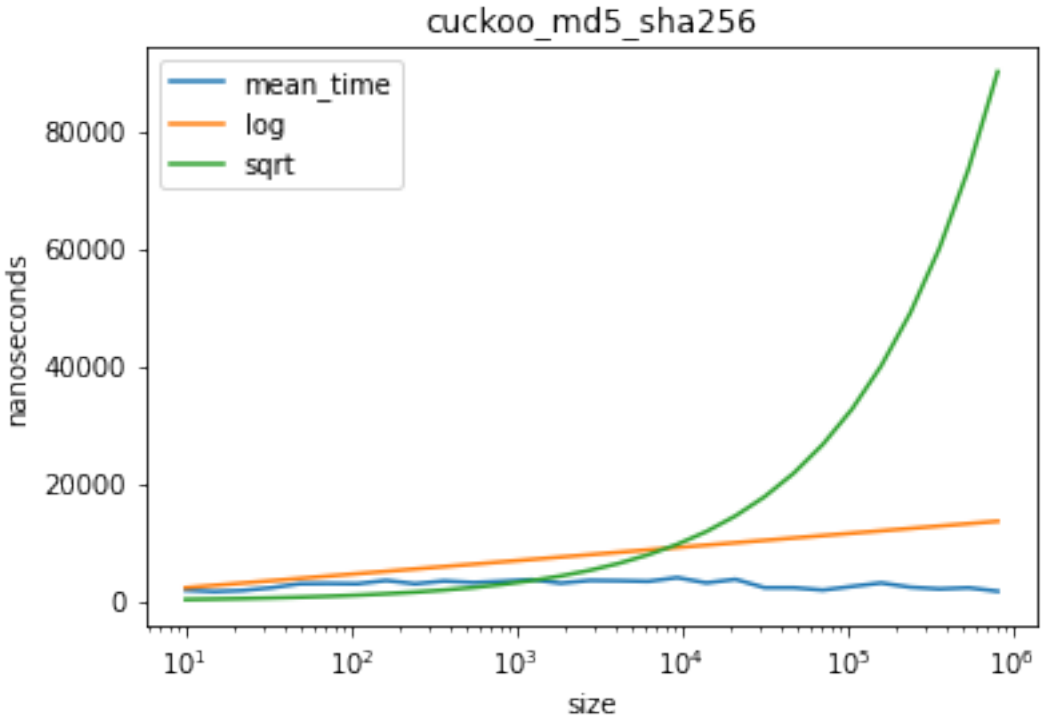

double_murmur
mean_time
log
sqrt
80000
60000
40000
20000
0
nanoseconds
10^1
10^2
10^3
10^4
10^5
10^6
size


double_sha256
mean_time
log
sqrt
80000
60000
40000
20000
0
nanoseconds
10^1
10^2
10^3
10^4
10^5
10^6
size

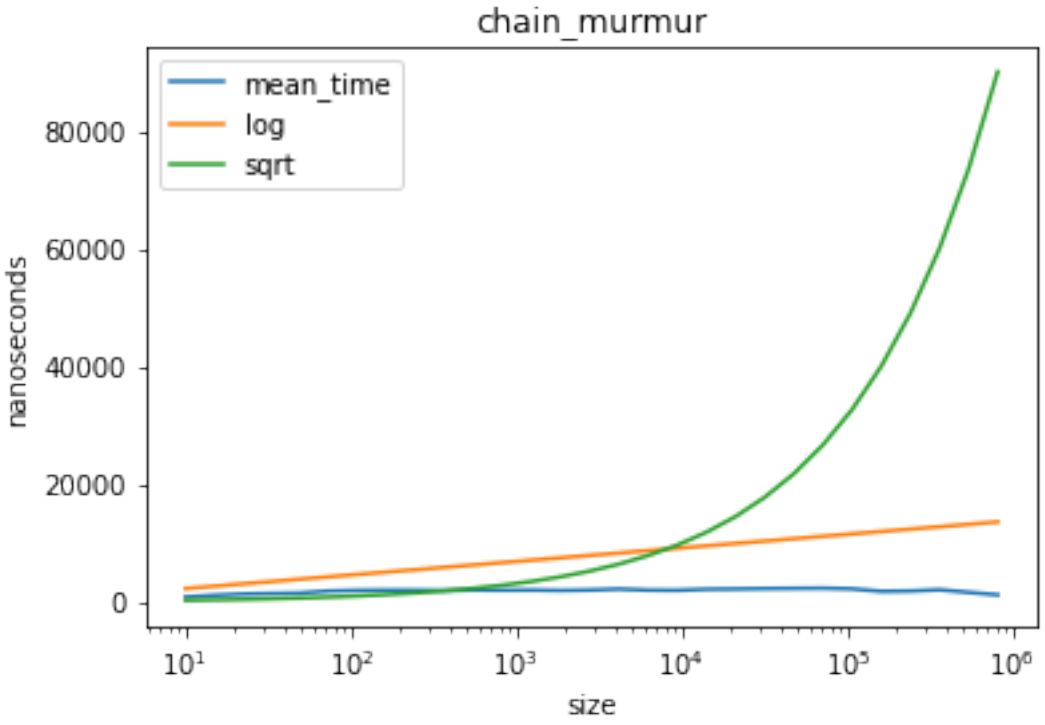
[8]:
```
process("cuckoo")
```

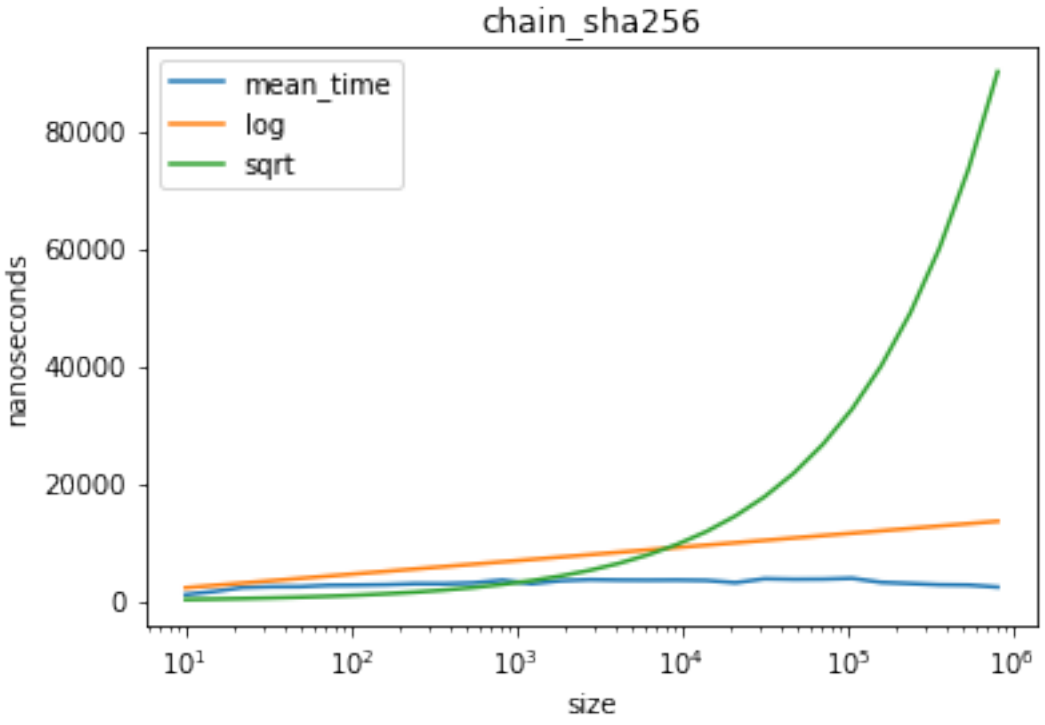


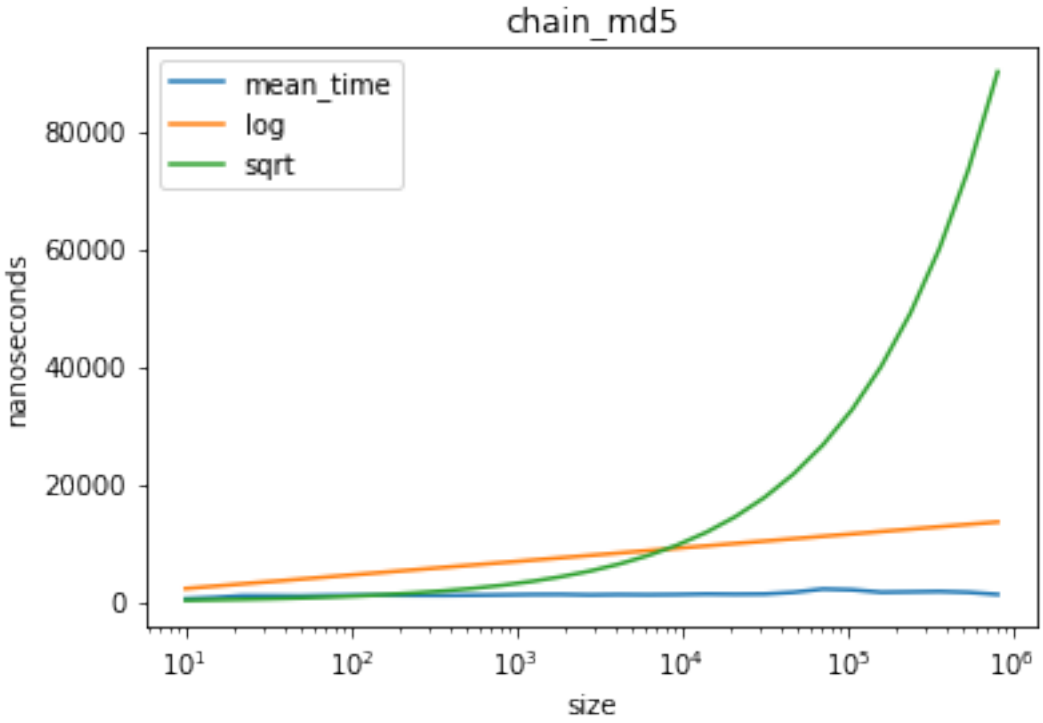




[9]:
```
process("chain")
```


chain_murmur
mean_time
log
sqrt
80000
60000
40000
20000
0
nanoseconds
10^1
10^2
10^3
10^4
10^5
10^6
size


chain_sha256
mean_time
log
sqrt
80000
60000
40000
20000
0
nanoseconds
10^1
10^2
10^3
10^4
10^5
10^6
size


chain_md5
mean_time
log
sqrt
80000
60000
40000
20000
0
nanoseconds
10^1
10^2
10^3
10^4
10^5
10^6
size

, .