CS121 lab

Yang Ziheng 2023533133

1.Introduction

I implement cuckoo hash.

2.Hardware and software

OS: WSL(Ubuntu)

CPU: 13th Gen Intel(R) Core(TM) i9-13900H 2.60 GHz

GPU: NVIDIA GeForce RTX 4060

CUDA: 12.0.140

GCC: 13.3.0

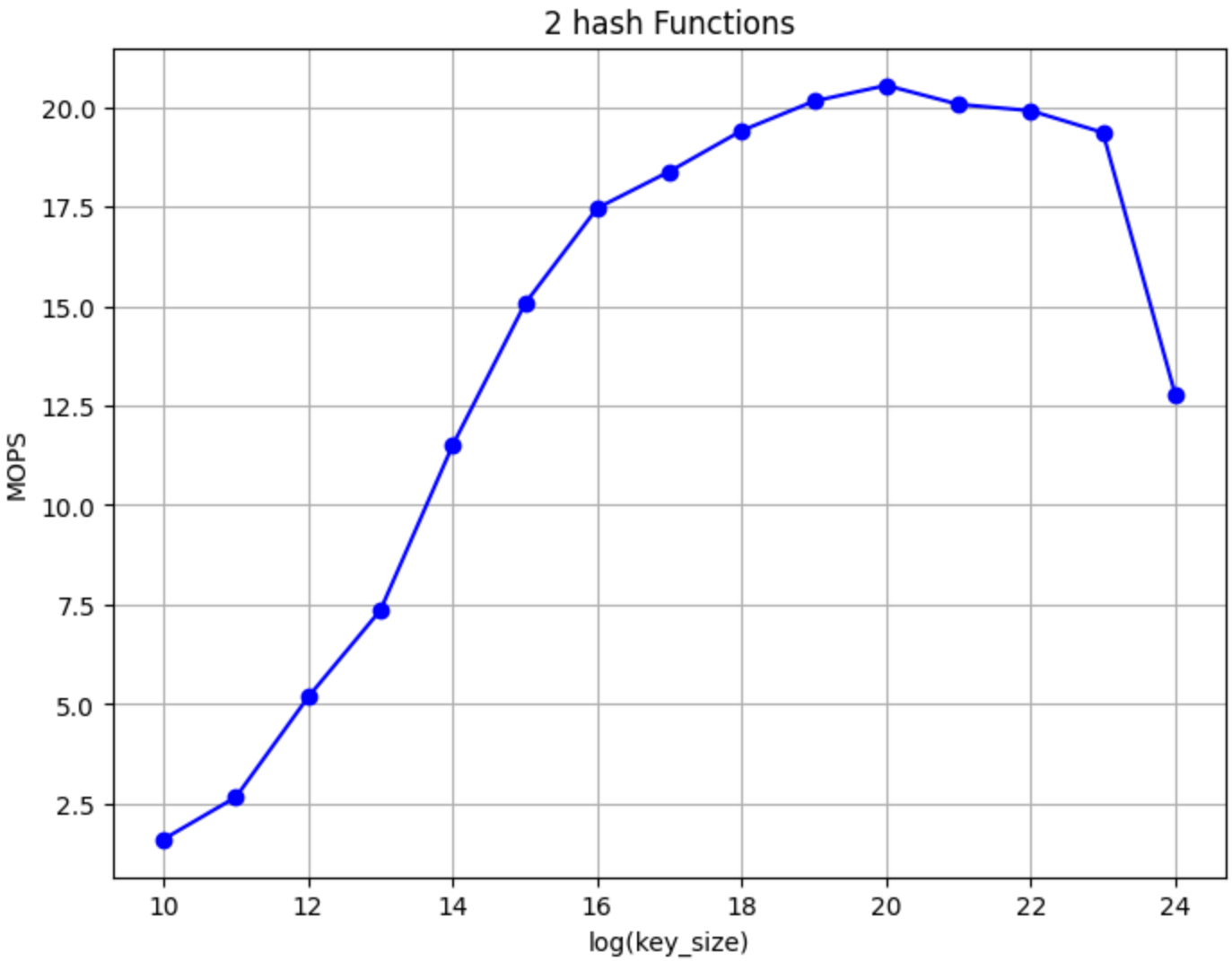
G++: 13.3.0

3.Tests

Test 1:insertion test(table_size=2^{25})

2 hash functions

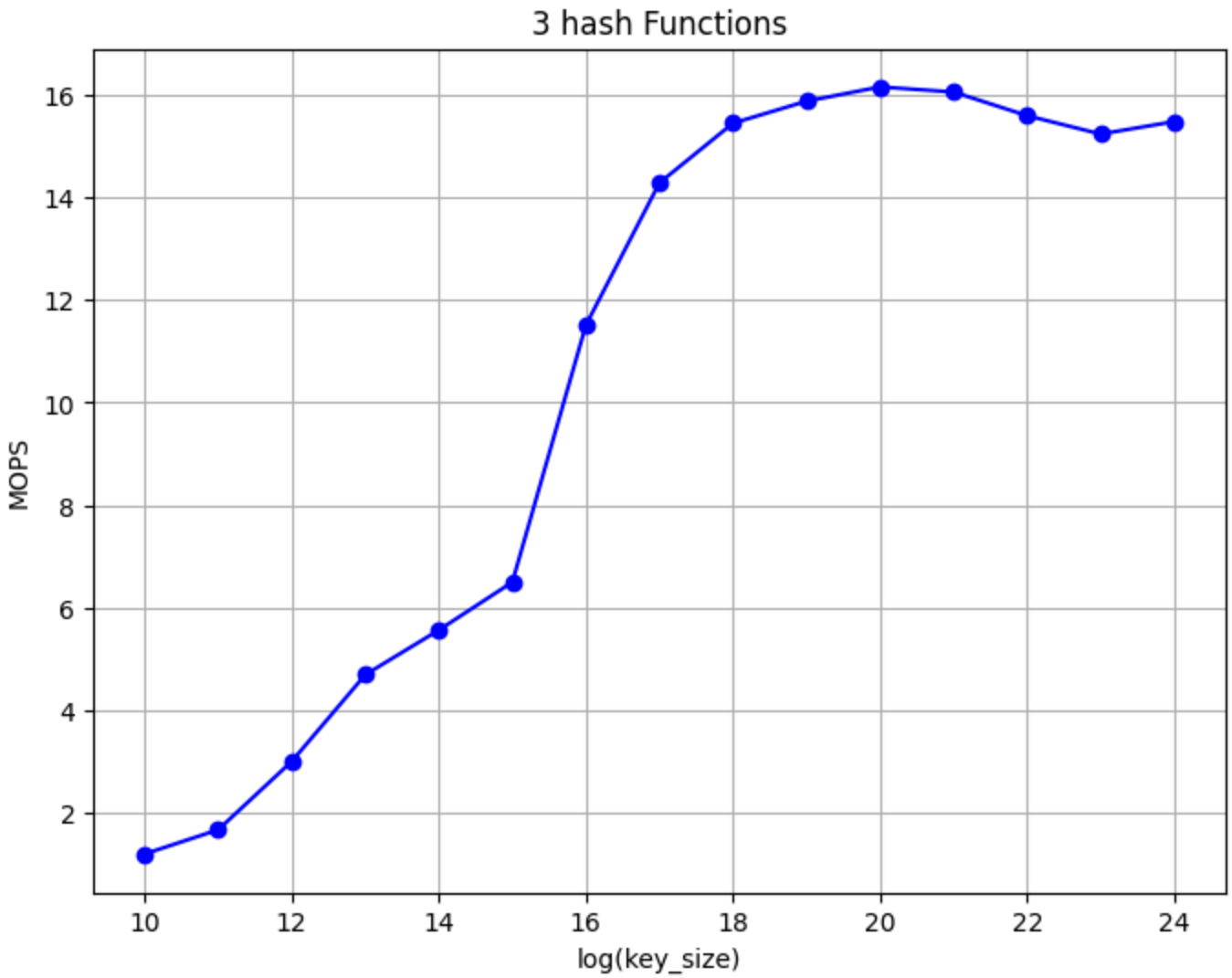
```
import matplotlib.pyplot as plt
import numpy as np
log_key_size = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
MOPS = [1.607535, 2.663199, 5.178255, 7.360288, 11.505618, 15.058824, 17.462297,
        18.38317, 19.409448, 20.149424, 20.546616, 20.070361, 19.910868, 19.361245, 12.782417]
plt.figure(figsize=(8, 6))
plt.plot(log_key_size, MOPS, marker='o', linestyle='-', color='b')
plt.title("2 hash Functions")
plt.xlabel("log(key_size)")
plt.ylabel("MOPS")
plt.grid(True)
plt.show()
```



Notice that when key_size=2^{24}, some keys are not inserted.

3 hash functions

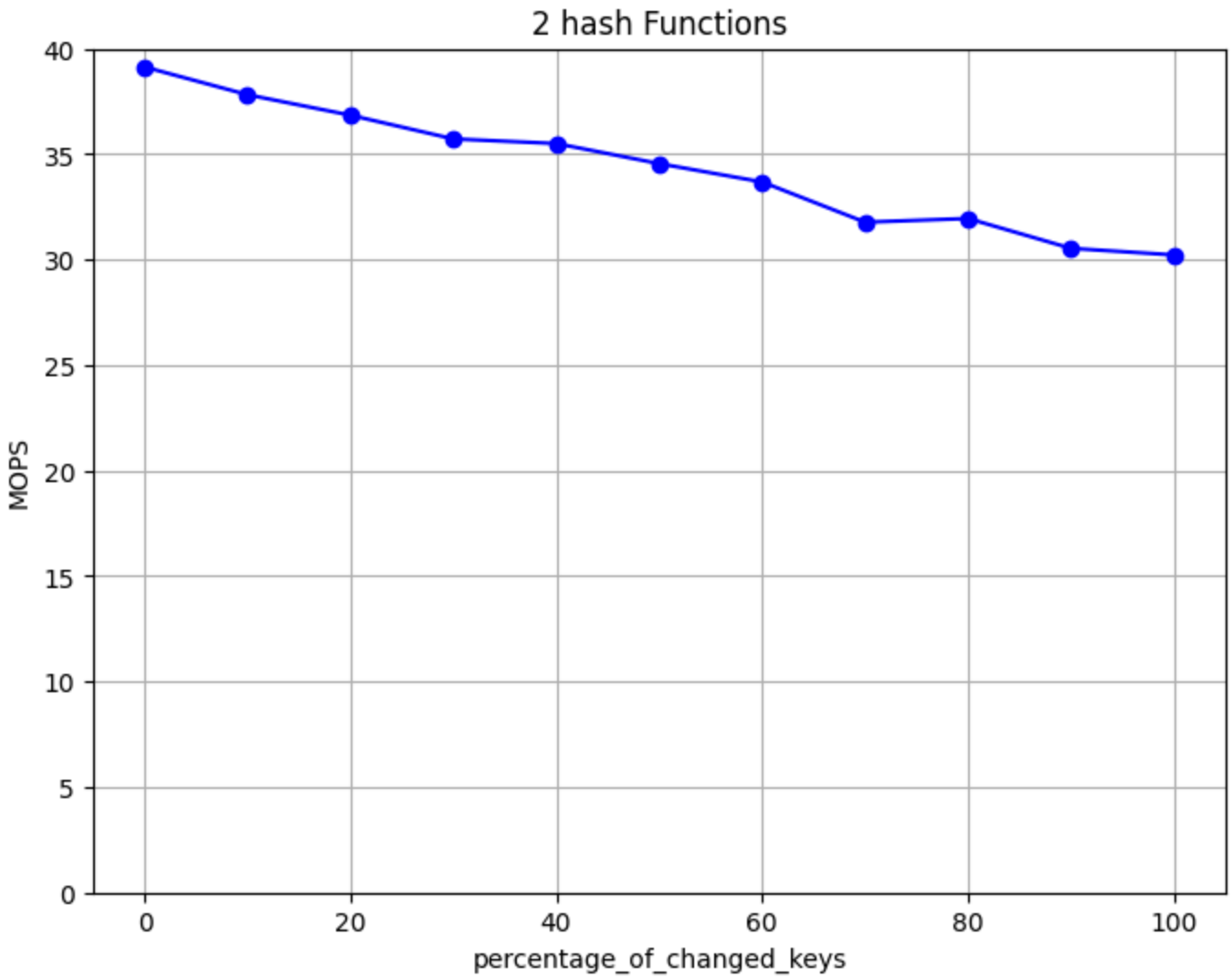
```
import matplotlib.pyplot as plt
import numpy as np
log_key_size = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
MOPS = [1.199063, 1.670473, 3.002933, 4.691867, 5.561439, 6.496431, 11.505618,
        14.276440, 15.440217, 15.869241, 16.147838, 16.048241, 15.587225, 15.228977, 15.471322]
plt.figure(figsize=(8, 6))
plt.plot(log_key_size, MOPS, marker='o', linestyle='-', color='b')
plt.title("3 hash Functions")
plt.xlabel("log(key_size)")
plt.ylabel("MOPS")
plt.grid(True)
plt.show()
```



Test 2:Lookup test(key_size = 2^{24} table_size=2^{25})

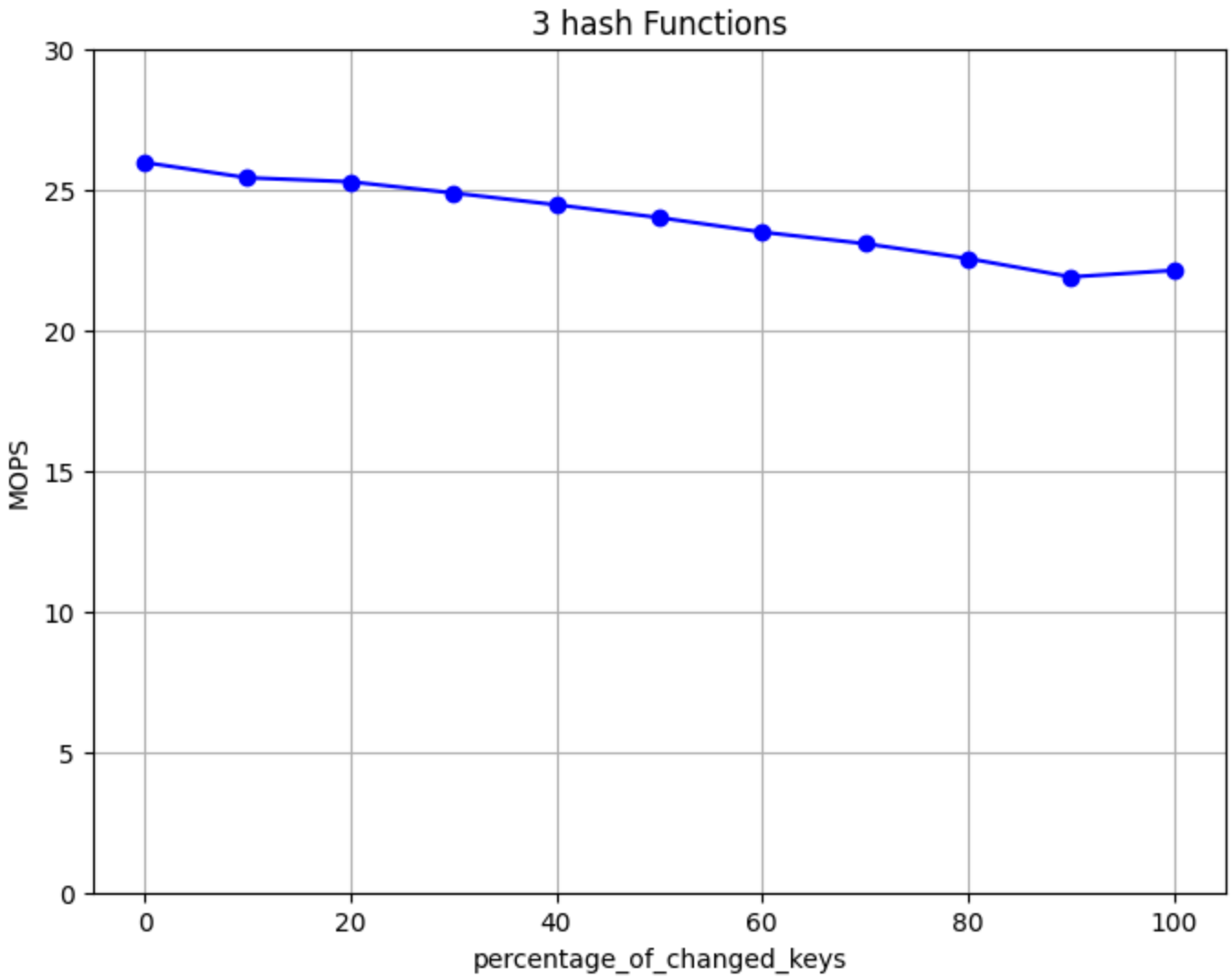
2 hash functions(some keys are not inserted)

```
import matplotlib.pyplot as plt
import numpy as np
percentage_of_changed_keys = [0,10,20,30,40,50,60,70,80,90,100]
MOPS = [39.126700,37.811475,36.840858,35.715734,35.496818,34.544548,33.674853,31.768051,31.9427
plt.figure(figsize=(8, 6))
plt.plot(percentage_of_changed_keys, MOPS, marker='o', linestyle='-', color='b')
plt.title("2 hash Functions")
plt.xlabel("percentage_of_changed_keys")
plt.ylabel("MOPS")
plt.ylim(0, 40)
plt.grid(True)
plt.show()
```



3 hash functions

```
import matplotlib.pyplot as plt
import numpy as np
percentage_of_changed_keys = [0,10,20,30,40,50,60,70,80,90,100]
MOPS = [25.949071,25.406590,25.263962,24.863938,24.445964,23.988458,23.473854,23.062353,22.5333
plt.figure(figsize=(8, 6))
plt.plot(percentage_of_changed_keys, MOPS, marker='o', linestyle='-', color='b')
plt.title("3 hash Functions")
plt.xlabel("percentage_of_changed_keys")
plt.ylabel("MOPS")
plt.ylim(0, 30)
plt.grid(True)
plt.show()
```



Test 3:Insertion with different table_size(key_size = 2^{24})

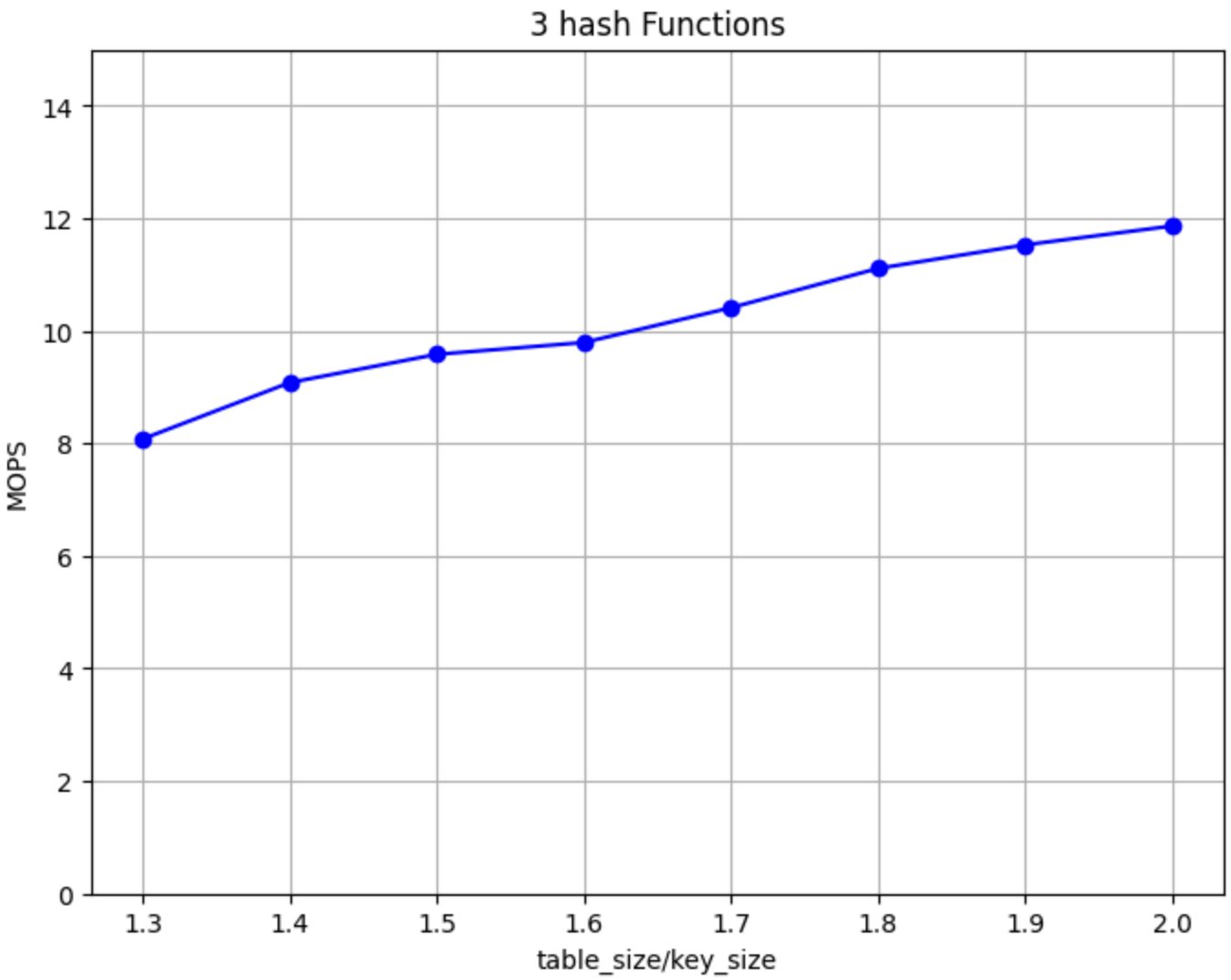
2 hash functions

meaningless,since no table_size accept all keys

3 hash functions

starting to accept all keys when table_size = 1.3key_size

```
import matplotlib.pyplot as plt
import numpy as np
factor = [1.3,1.4,1.5,1.6,1.7,1.8,1.9,2.0]
MOPS = [8.076636,9.078629, 9.581079,9.791214,10.409317, 11.107650,11.526890,11.862114]
plt.figure(figsize=(8, 6))
plt.plot(factor, MOPS, marker='o', linestyle='-', color='b')
plt.title("3 hash Functions")
plt.xlabel("table_size/key_size")
plt.ylabel("MOPS")
plt.ylim(0, 15)
plt.grid(True)
plt.show()
```



Test 4:Insertion with different length of an eviction chain(key_size = 2^{24} table_size=2^{25})

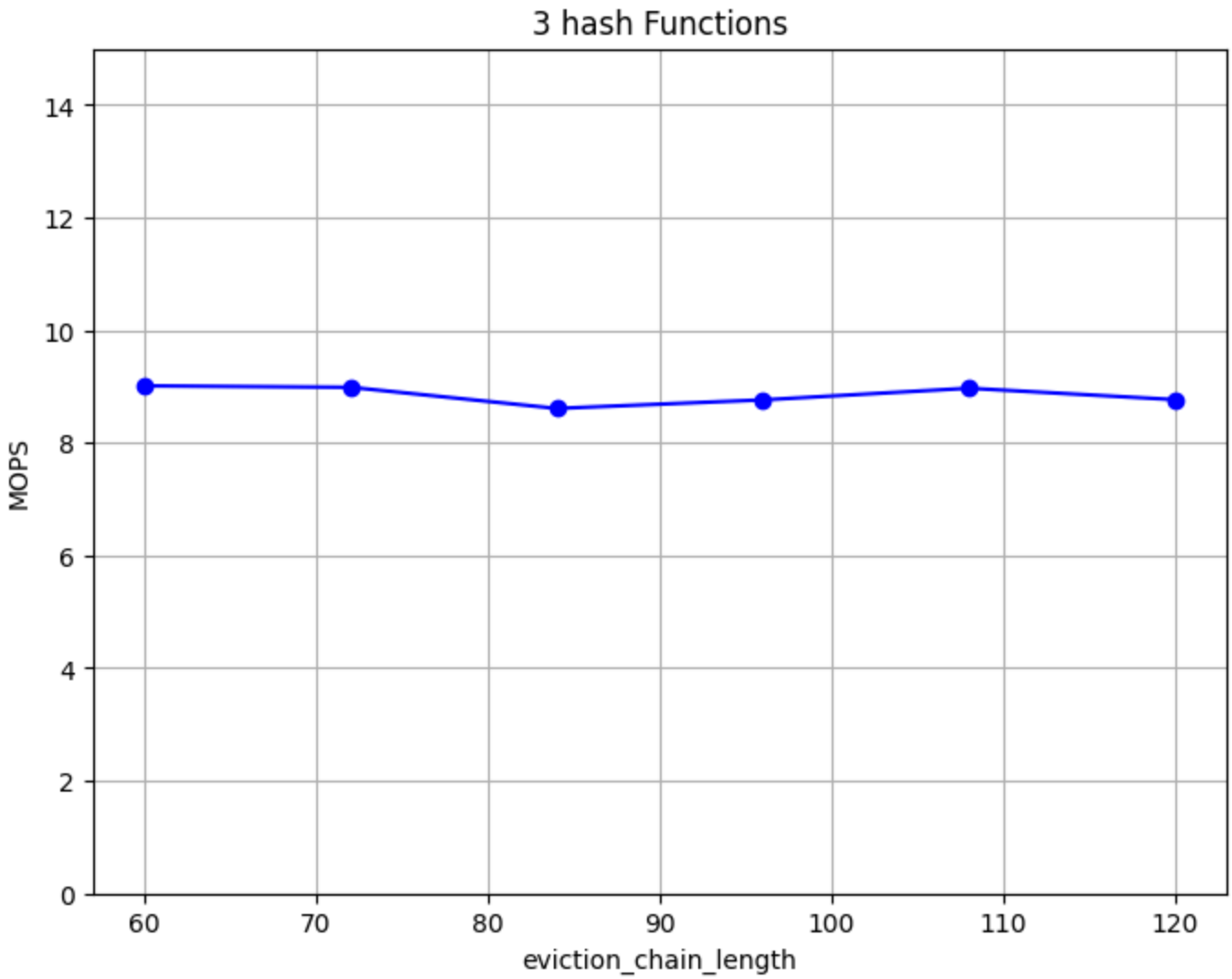
2 hash functions

meaningless,since no table_size accept all keys

3 hash functions

starting to accept all keys when eviction_chain_length = 60

```
import matplotlib.pyplot as plt
import numpy as np
eviction_chain_length = [60,72,84,96,108,120]
MOPS = [ 9.010794,8.984712, 8.609286,8.760967,8.968157, 8.765970]
plt.figure(figsize=(8, 6))
plt.plot(eviction_chain_length, MOPS, marker='o', linestyle='-', color='b')
plt.title("3 hash Functions")
plt.xlabel("eviction_chain_length")
plt.ylabel("MOPS")
plt.ylim(0, 15)
plt.grid(True)
plt.show()
```


3 hash Functions
MOPS
eviction_chain_length
0
2
4
6
8
10
12
14
60
70
80
90
100
110
120