# In this part we convert boolean variables to int and will do someUnsufuleOperations() to make the proccess slower

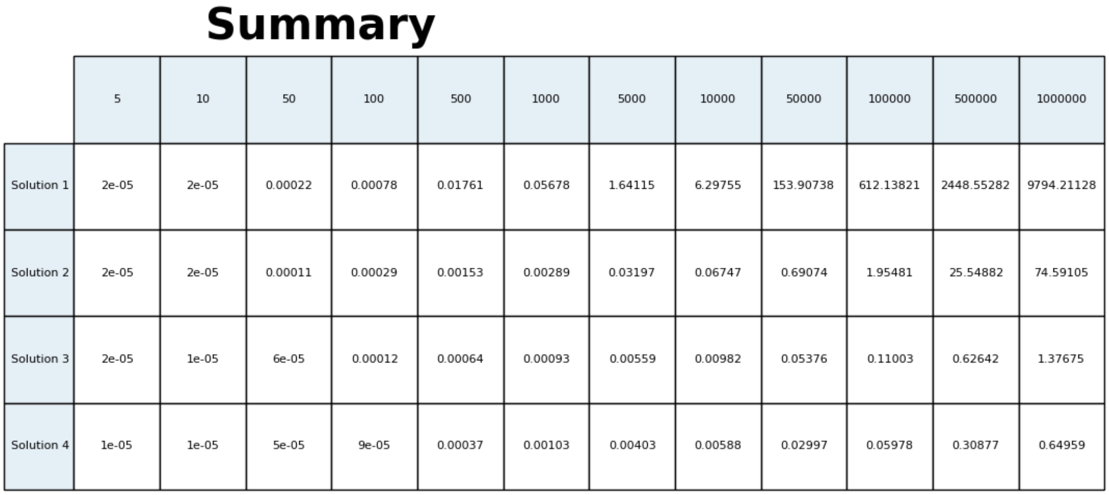
## Code with auto data, table and graph generate:

```
import math
import random
import time
import matplotlib.pyplot as plt
import numpy as np



def someUnsufuleOperations():
    p = 1
    p -= 1
    p += 1
    p = 0

def solution1(n):
    count = 0
    for i in range(2, n + 1):
        someUnsufuleOperations()
        isPrime = 1
        for j in range(2, i):
            if(i%j == 0):
                isPrime = 0
        if(isPrime == 1):
            count += 1
    return count

def solution2(n):
    count = 0
    for i in range(2, n + 1):
        someUnsufuleOperations()
        isPrime = 1
        for j in range(2, math.floor(math.sqrt(i))+1):
            if(i%j == 0):
                isPrime = 0
        if(isPrime == 1):
            count += 1
    return count

def solution3(n):
    # hasDivisor means divisors except 1 and the number itself
    hasDivisor = [0 for i in range(n+1)] # from 0 to n
    for i in range(2, n + 1):
        someUnsufuleOperations()
        for j in range(2 * i, n + 1, i):
            hasDivisor[j] = 1
    return n - hasDivisor.count(1) - 1 # -1 is for 1

def solution4(n):
    # hasDivisor means divisors except 1 and the number itself
    hasDivisor = [0 for i in range(n+1)] # from 0 to n
    for i in range(2, n + 1):
        someUnsufuleOperations()
        if(hasDivisor[i] == 0):
            for j in range(i * i, n + 1, i):
                hasDivisor[j] = 1
    return n - hasDivisor.count(1) - 1  #-1 is for 1

def drawGraph(x, y):
    plt.plot(x,y)
    plt.show()

def drawSummaryGraphs():
    for k in range(0, 3):
        for i in range(k, 4):
            plt.plot(inputs, times[i], label = "solution
{}".format(i+1))
        plt.legend()
        plt.show()

def drawTable(rows, columns, data, title):
    fig, ax = plt.subplots()
    ax.set_axis_off()
    rcolors = plt.cm.BuPu(np.full(len(rows), 0.1))
    ccolors = plt.cm.BuPu(np.full(len(columns), 0.1))
    table = ax.table(
        cellText = data,
        rowLabels = rows,
        colLabels = columns,
        rowColours = rcolors,
        colColours = ccolors,
        cellLoc ='center',
        loc ='upper left')

#     table.auto_set_font_size(False)
    table.set_fontsize(30)
    table.scale(2, 5)
    ax.set_title(title,
                 fontweight ="bold", fontdict={'fontsize': 30})

    plt.show()

def drawAllSingleTG():
    for i in range(4):
        drawTable(["Time"], inputs, [times[i]], 'Solution
{}'.format(i+1))
        drawGraph(inputs, times[i])

def calculateTimes():
    for n in inputs:
        start_time = time.time()
        solution1(n)
        times[0].append(time.time() - start_time)

        start_time = time.time()
        solution2(n)
        times[1].append(time.time() - start_time)

        start_time = time.time()
        solution3(n)
        times[2].append(time.time() - start_time)

        start_time = time.time()
        solution4(n)
        times[3].append(time.time() - start_time)

def drawStuff():
    drawTable(["Solution 1", "Solution 2", "Solution 3", "Solution
4"], inputs, times, 'Summary')
    drawSummaryGraphs()
    drawAllSingleTG()
    print("="*10 + " Times List " + "="*10)
    print(times)


inputs = [5, 10, 50, 100, 500, 10**3, 5 * 10**3, 10**4, 5 * 10**4,
10**5, 5 * 10**5, 10**6]
times = [[], [], [], []]
outputs = [[], [], [], []]
calculateTimes()
drawStuff()
```

### Code with saved datas:

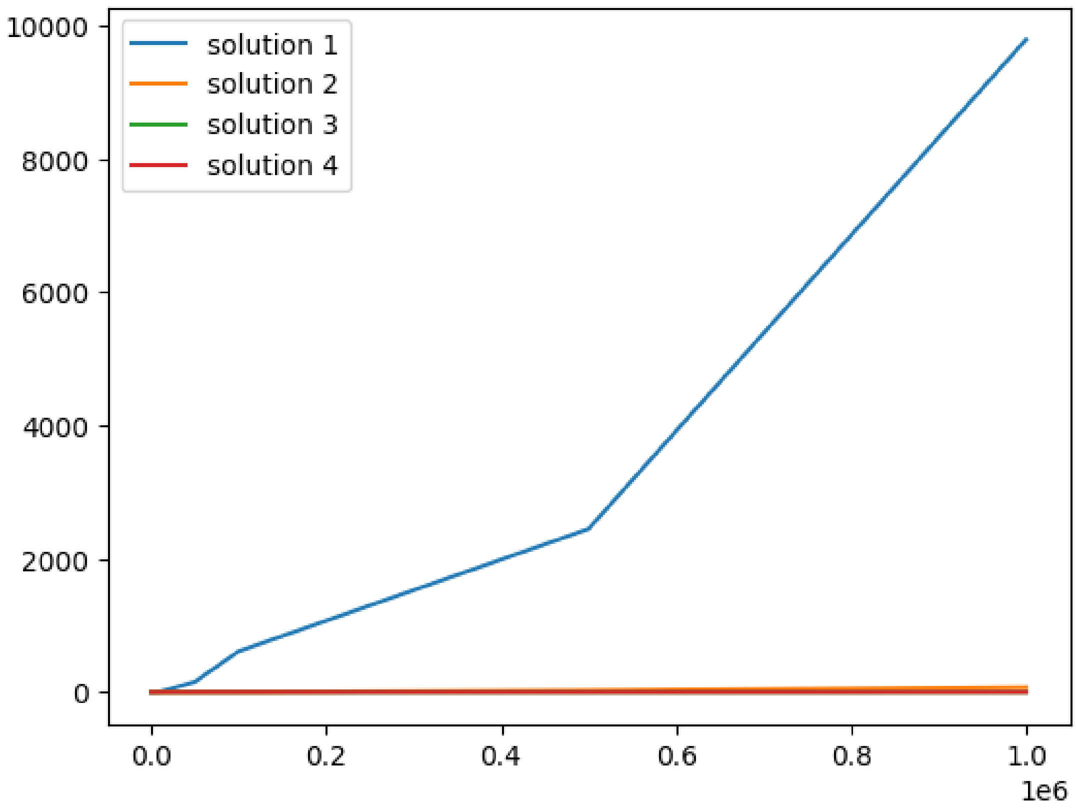
```
import matplotlib.pyplot as plt
import numpy as np

inputs = [5, 10, 50, 100, 500, 10**3, 5 * 10**3, 10**4, 5 * 10**4,
10**5, 5 * 10**5, 10**6]

times = [[1.71661376953125e-05, 1.7642974853515625e-05,
0.00021934509277734375, 0.0007827281951904297, 0.017605304718017578,
0.05678439140319824, 1.64115309715271, 6.2975544929504395,
153.90738463401794, 612.1382052898407, 4*612.1382052898407,
```

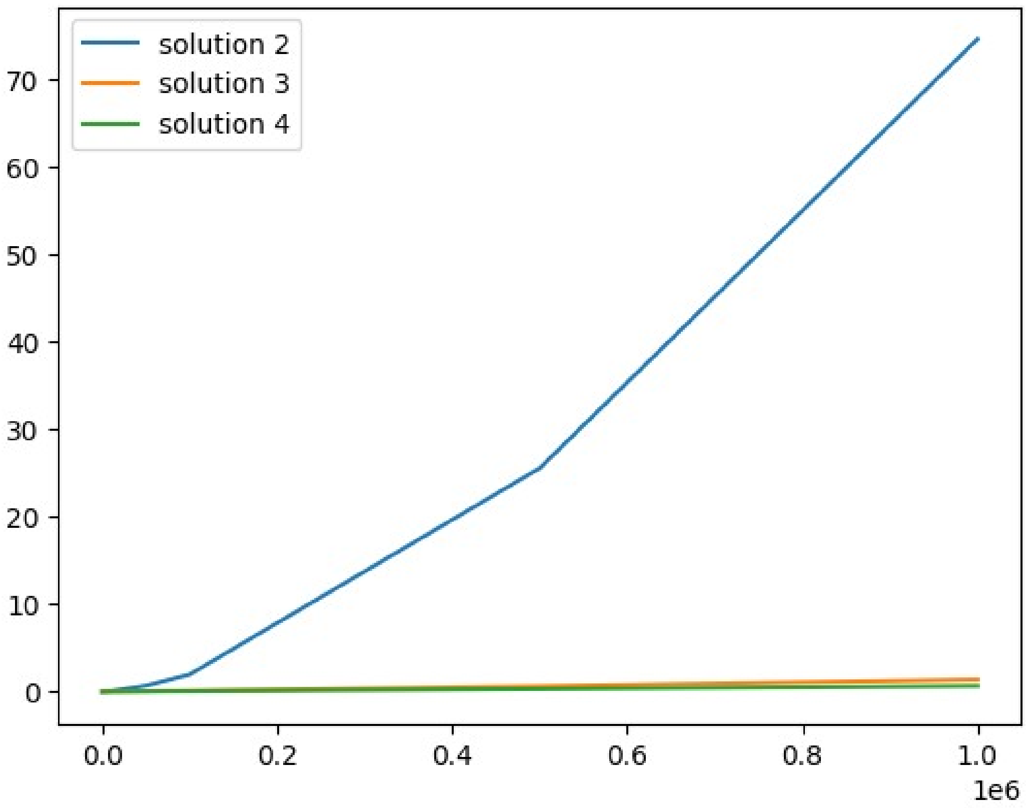
```python
16*612.1382052898407], [1.6927719116210938e-05, 1.7881393432617188e-
05, 0.00010609626770019531, 0.0002925395965576172,
0.00153350830078125, 0.002889394760131836, 0.031970977783203125,
0.06746912002563477, 0.6907429695129395, 1.9548075199127197,
25.54882311820984, 74.59105324745178], [1.5974044799804688e-05,
1.4543533325195312e-05, 6.103515625e-05, 0.0001220703125,
0.0006418228149414062, 0.0009315013885498047, 0.005592823028564453,
0.009818553924560547, 0.05376315116882324, 0.1100320816040039,
0.6264150142669678, 1.376746654510498], [1.0728836059570312e-05,
1.2636184692382812e-05, 4.601478576660156e-05, 8.606910705566406e-05,
0.0003707408905029297, 0.0010349750518798828, 0.004034519195556641,
0.00587916374206543, 0.02996516227722168, 0.059784889221191406,
0.30876660346984863, 0.6495921611785889]]

for i in range(len(times)):
    for j in range(len(times[i])):
        times[i][j] = float("{:.5f}".format(times[i][j]))

def drawGraph(x, y):
    plt.plot(x,y)
    plt.show()

def drawSummaryGraphs():
    for k in range(0, 3):
        for i in range(k, 4):
            plt.plot(inputs, times[i], label = "solution
{}".format(i+1))
        plt.legend()
        plt.show()

def drawTable(rows, columns, data, title):
    fig, ax = plt.subplots()
    ax.set_axis_off()
    rcolors = plt.cm.BuPu(np.full(len(rows), 0.1))
    ccolors = plt.cm.BuPu(np.full(len(columns), 0.1))
    table = ax.table(
        cellText = data,
        rowLabels = rows,
        colLabels = columns,
        rowColours = rcolors,
        colColours = ccolors,
        cellLoc ='center',
        loc ='upper left')

#     table.auto_set_font_size(False)
    table.set_fontsize(30)
    table.scale(2, 5)
    ax.set_title(title,
                 fontweight ="bold", fontdict={'fontsize': 30})
    plt.show()

    for i in range(len(data)):
        for j in range(len(data[i])):
            if(data[i][j] == "Too much"):
                data[i][j] = -1

def drawAllSingleGraphs():
    for i in range(4):
        drawTable(["Time"], inputs, [times[i]], 'Solution
{}'.format(i+1))
        drawGraph(inputs, times[i])

def drawStuff():
    drawTable(["Solution 1", "Solution 2", "Solution 3", "Solution
4"], inputs, times, 'Summary')
    drawSummaryGraphs()
    drawAllSingleGraphs()



drawStuff()
```

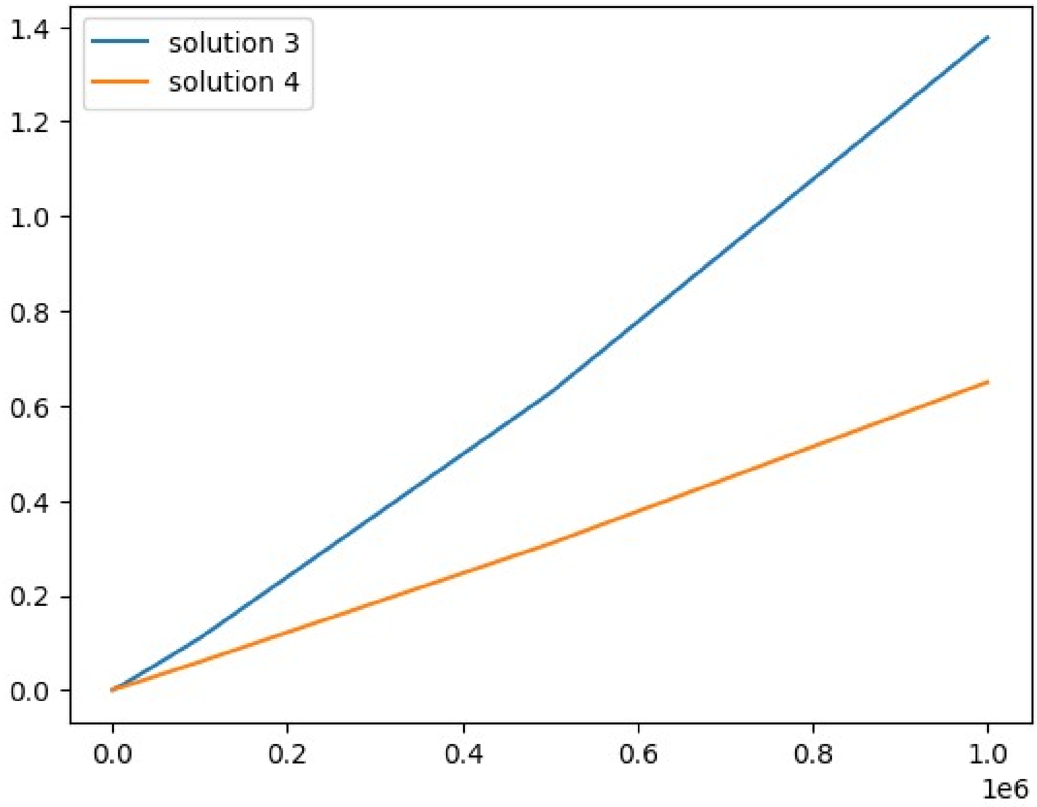


Summary

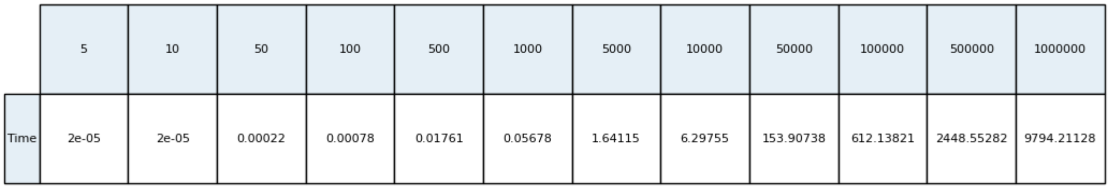
| | 5 | 10 | 50 | 100 | 500 | 1000 | 5000 | 10000 | 50000 | 100000 | 500000 | 1000000 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Solution 1 | 2e-05 | 2e-05 | 0.00022 | 0.00078 | 0.01761 | 0.05678 | 1.64115 | 6.29755 | 153.90738 | 612.13821 | 2448.55282 | 9794.21128 |
| Solution 2 | 2e-05 | 2e-05 | 0.00011 | 0.00029 | 0.00153 | 0.00289 | 0.03197 | 0.06747 | 0.69074 | 1.95481 | 25.54882 | 74.59105 |
| Solution 3 | 2e-05 | 1e-05 | 6e-05 | 0.00012 | 0.00064 | 0.00093 | 0.00559 | 0.00982 | 0.05376 | 0.11003 | 0.62642 | 1.37675 |
| Solution 4 | 1e-05 | 1e-05 | 5e-05 | 9e-05 | 0.00037 | 0.00103 | 0.00403 | 0.00588 | 0.02997 | 0.05978 | 0.30877 | 0.64959 |


solution 1
solution 2
solution 3
solution 4
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
1e6

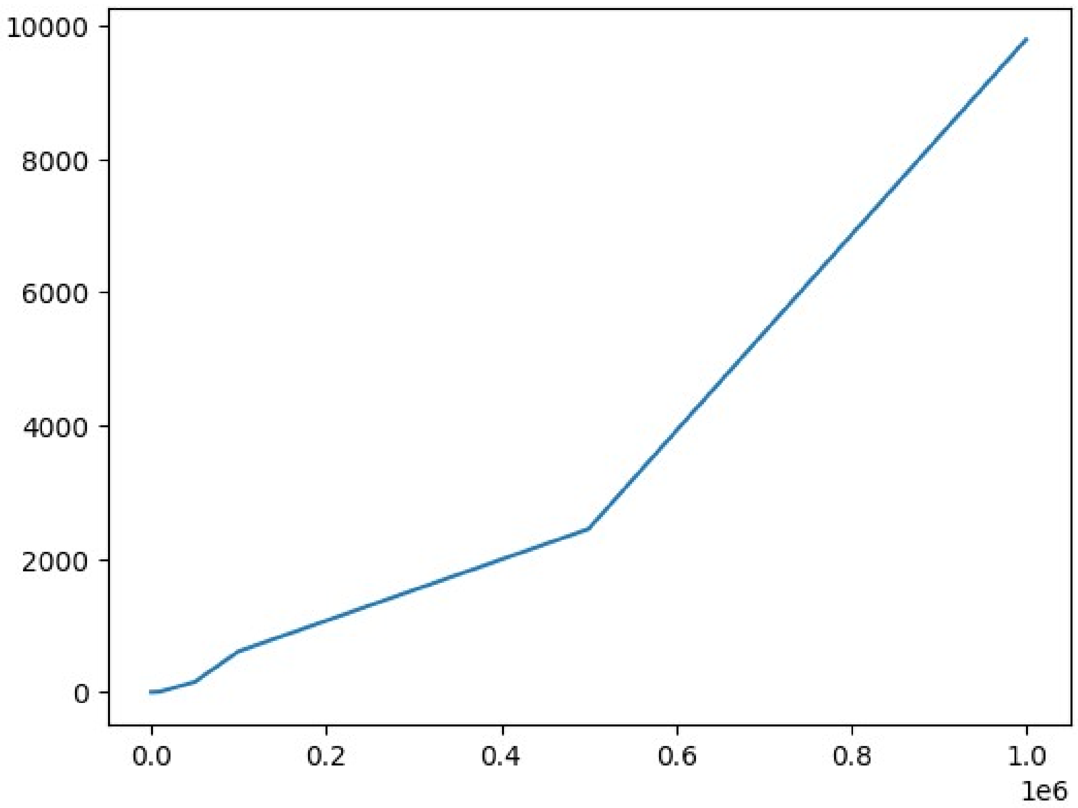

solution 2
solution 3
solution 4
70
60
50
40
30
20
10
0
0.0
0.2
0.4
0.6
0.8
1.0
1e6



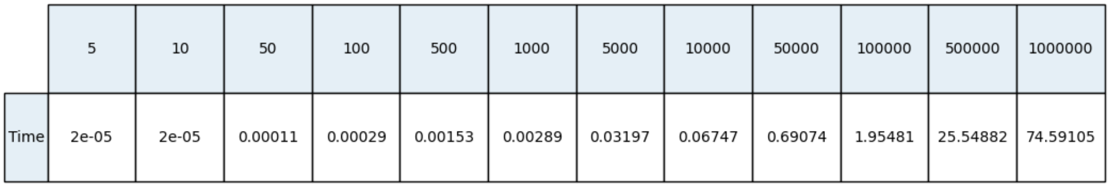
## Solution 1



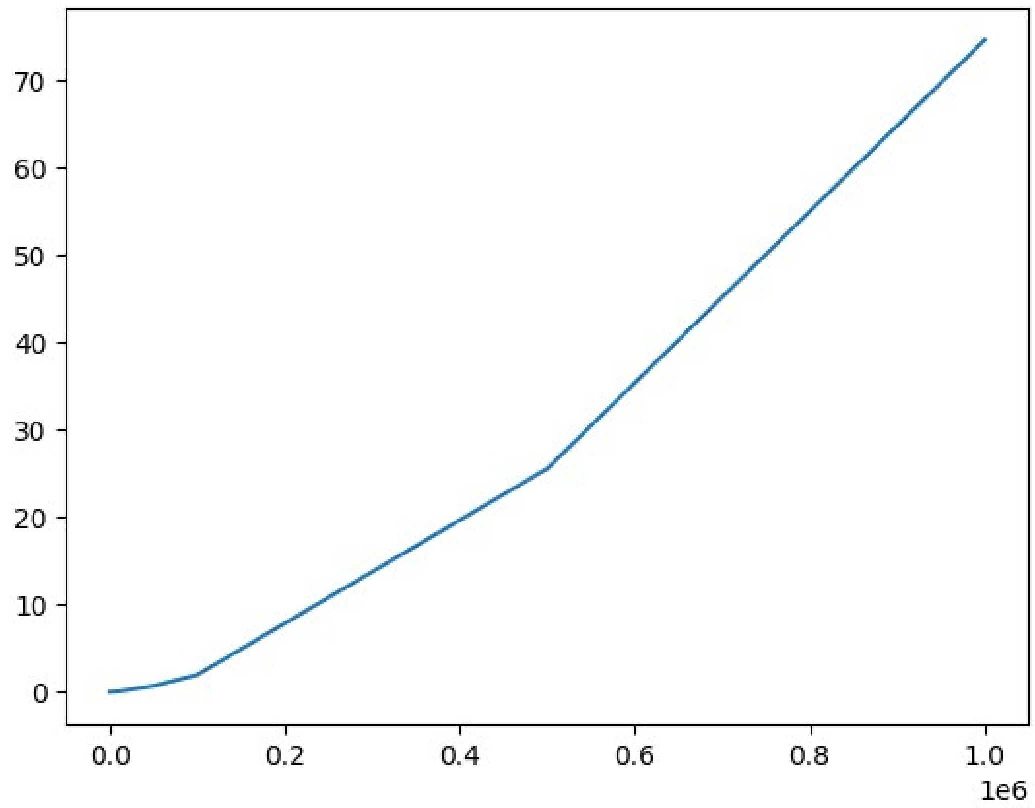
| | 5 | 10 | 50 | 100 | 500 | 1000 | 5000 | 10000 | 50000 | 100000 | 500000 | 1000000 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time | 2e-05 | 2e-05 | 0.00022 | 0.00078 | 0.01761 | 0.05678 | 1.64115 | 6.29755 | 153.90738 | 612.13821 | 2448.55282 | 9794.21128 |



## Solution 2

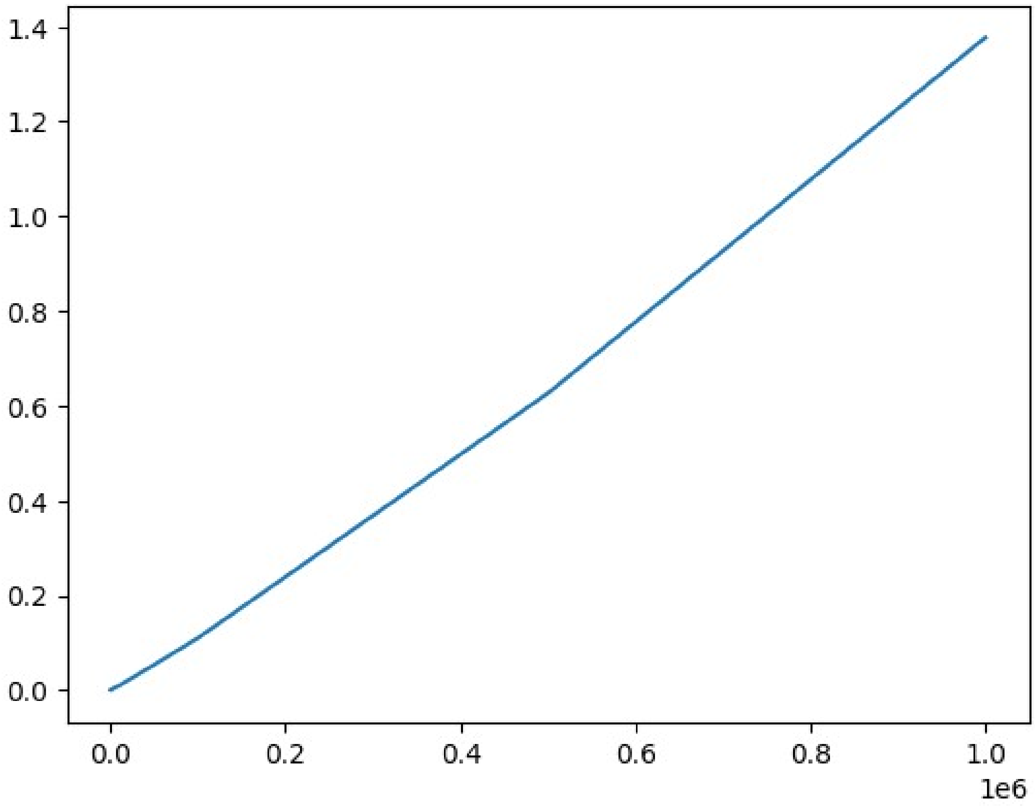


| | 5 | 10 | 50 | 100 | 500 | 1000 | 5000 | 10000 | 50000 | 100000 | 500000 | 1000000 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time | 2e-05 | 2e-05 | 0.00011 | 0.00029 | 0.00153 | 0.00289 | 0.03197 | 0.06747 | 0.69074 | 1.95481 | 25.54882 | 74.59105 |



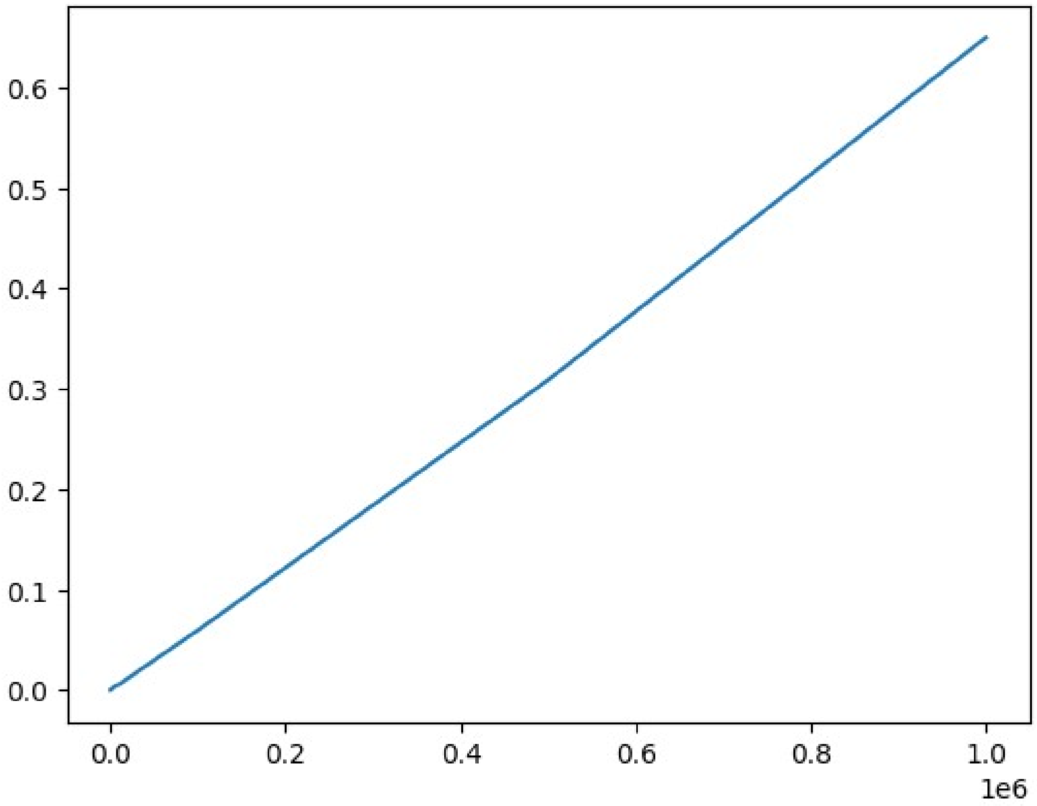
## Solution 3

| | 5 | 10 | 50 | 100 | 500 | 1000 | 5000 | 10000 | 50000 | 100000 | 500000 | 1000000 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time | 2e-05 | 1e-05 | 6e-05 | 0.00012 | 0.00064 | 0.00093 | 0.00559 | 0.00982 | 0.05376 | 0.11003 | 0.62642 | 1.37675 |



## Solution 4

| | 5 | 10 | 50 | 100 | 500 | 1000 | 5000 | 10000 | 50000 | 100000 | 500000 | 1000000 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time | 1e-05 | 1e-05 | 5e-05 | 9e-05 | 0.00037 | 0.00103 | 0.00403 | 0.00588 | 0.02997 | 0.05978 | 0.30877 | 0.64959 |


0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1e6