In [10]:

```
import random
import time
import tracemalloc
import matplotlib.pyplot as plt
from random import sample
```

In [ ]:

```
random.seed(1)
sequence = [i for i in range(100000)]
random.shuffle(sequence)
#print(sequence)
```

In [11]:

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None


class LinkedList:
    def __init__(self):
        self.head = None

    def print_list(self):
        cur_node = self.head
        while cur_node:
            print(cur_node.data)
            cur_node = cur_node.next

    def append(self, data):
        new_node = Node(data)

        if self.head is None:
            self.head = new_node
            return

        last_node = self.head
        while last_node.next:
            last_node = last_node.next
        last_node.next = new_node

    def delete_node_at_pos(self, pos):

        cur_node = self.head
```

```
        if pos == 0:
            self.head = cur_node.next
            cur_node = None
            return

        prev = None
        count = 0
        while cur_node and count != pos:
            prev = cur_node
            cur_node = cur_node.next
            count += 1

        if cur_node is None:
            return

        prev.next = cur_node.next
        cur_node = None
```

In [12]:

```
llist = LinkedList()
for i in range(len(sequence)):
    llist.append(sequence[i])
#llist.print_list()
```

In [35]:

```
start = time.time()
tracemalloc.start()
for i in range(50):
    llist.delete_node_at_pos(i)
end = time.time()
runtime1 = end - start
current, peak = tracemalloc.get_traced_memory()
usage1 = current / 10**6
print(f"Current memory usage is {usage1}MB; Peak was {peak / 10**6}MB")
tracemalloc.stop()
print(f"Runtime of the snippet is {runtime1}")
```

```
Current memory usage is 0.000933MB; Peak was 0.011929MB
Runtime of the snippet is 0.0
```

In [36]:

```
start = time.time()
tracemalloc.start()
for i in range(100):
    llist.delete_node_at_pos(i)
end = time.time()
```

```
runtime2 = end - start
current, peak = tracemalloc.get_traced_memory()
usage2 = current / 10**6
print(f"Current memory usage is {usage2}MB; Peak was {peak / 10**6}MB")
tracemalloc.stop()
print(f"Runtime of the snippet is {runtime2}")
```

```
Current memory usage is 0.000933MB; Peak was 0.011929MB
Runtime of the snippet is 0.0009936471557617188
```

In [37]:

```
start = time.time()
tracemalloc.start()
for i in range(500):
    llist.delete_node_at_pos(i)
end = time.time()
runtime3 = end - start
current, peak = tracemalloc.get_traced_memory()
usage3 = current / 10**6
print(f"Current memory usage is {usage3}MB; Peak was {peak / 10**6}MB")
tracemalloc.stop()
print(f"Runtime of the snippet is {runtime3}")
```

```
Current memory usage is 0.001284MB; Peak was 0.012561MB
Runtime of the snippet is 0.036338090896606445
```

In [38]:

```
start = time.time()
tracemalloc.start()
for i in range(1000):
    llist.delete_node_at_pos(i)
end = time.time()
runtime4 = end - start
current, peak = tracemalloc.get_traced_memory()
usage4 = current / 10**6
print(f"Current memory usage is {usage4}MB; Peak was {peak / 10**6}MB")
tracemalloc.stop()
print(f"Runtime of the snippet is {runtime4}")
```

```
Current memory usage is 0.006499MB; Peak was 0.017376MB
Runtime of the snippet is 0.2176377773284912
```

In [39]:

```
start = time.time()
tracemalloc.start()
for i in range(2000):
    llist.delete_node_at_pos(i)
end = time.time()
runtime5 = end - start
```

```
current, peak = tracemalloc.get_traced_memory()
usage5 = current / 10**6
print(f"Current memory usage is {usage5}MB; Peak was {peak / 10**6}MB")
tracemalloc.stop()
print(f"Runtime of the snippet is {runtime5}")
```

```
Current memory usage is 0.006499MB; Peak was 0.017376MB
Runtime of the snippet is 1.1903250217437744
```

In [40]:

```
start = time.time()
tracemalloc.start()
for i in range(5000):
    llist.delete_node_at_pos(i)
end = time.time()
runtime6 = end - start
current, peak = tracemalloc.get_traced_memory()
usage6 = current / 10**6
print(f"Current memory usage is {usage6}MB; Peak was {peak / 10**6}MB")
tracemalloc.stop()
print(f"Runtime of the snippet is {runtime6}")
```

```
Current memory usage is 0.006979MB; Peak was 0.017856MB
Runtime of the snippet is 8.6834397315979
```

In [41]:

```
start = time.time()
tracemalloc.start()
for i in range(10000):
    llist.delete_node_at_pos(i)
end = time.time()
runtime7 = end - start
current, peak = tracemalloc.get_traced_memory()
usage7 = current / 10**6
print(f"Current memory usage is {usage7}MB; Peak was {peak / 10**6}MB")
tracemalloc.stop()
print(f"Runtime of the snippet is {runtime7}")
```

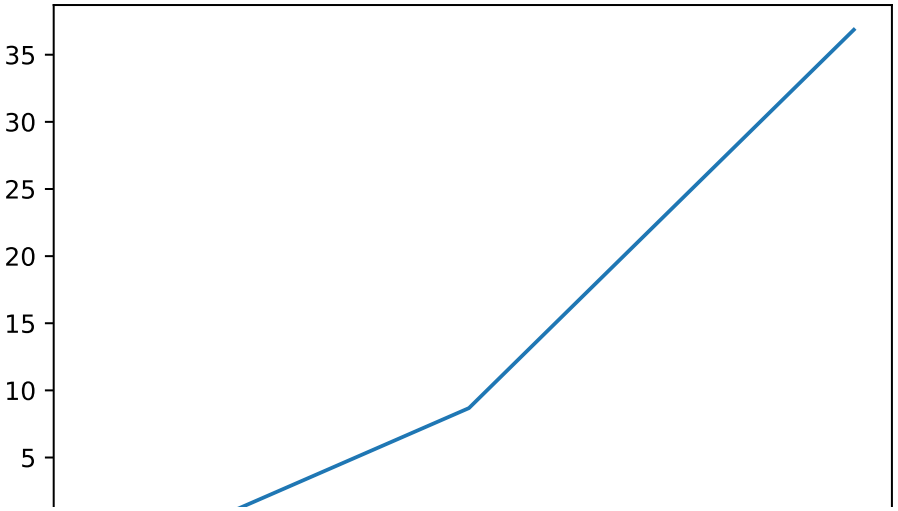
```
Current memory usage is 0.007759MB; Peak was 0.018636MB
Runtime of the snippet is 36.86092448234558
```

In [46]:

```
plt.plot([100,200,500, 1000, 2000, 5000, 10000],[ runtime1,runtime2,runtime3,runtime4,runtime5,runtime6,runtime7])
```

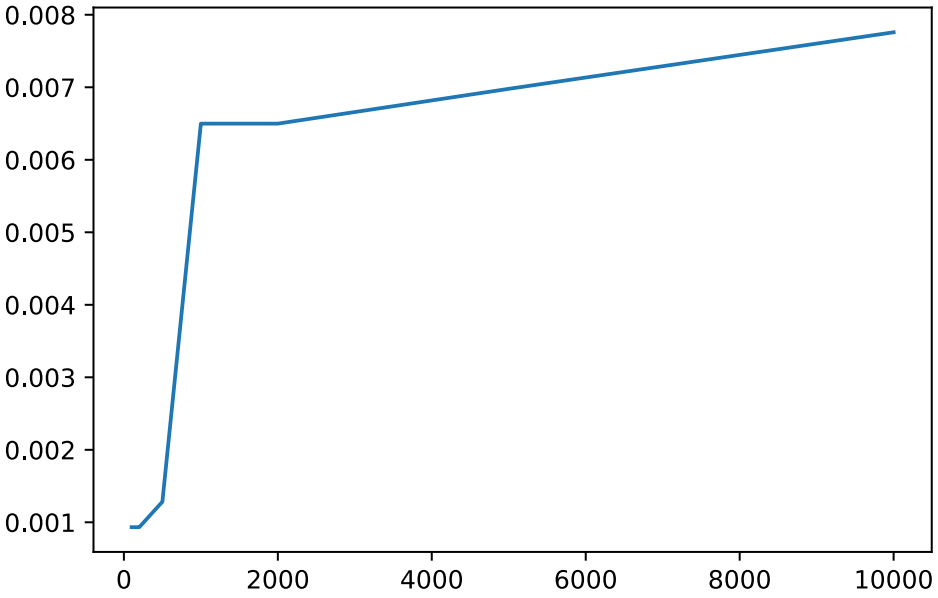
Out[46]:

```
[<matplotlib.lines.Line2D at 0x264651d5c70>]
```



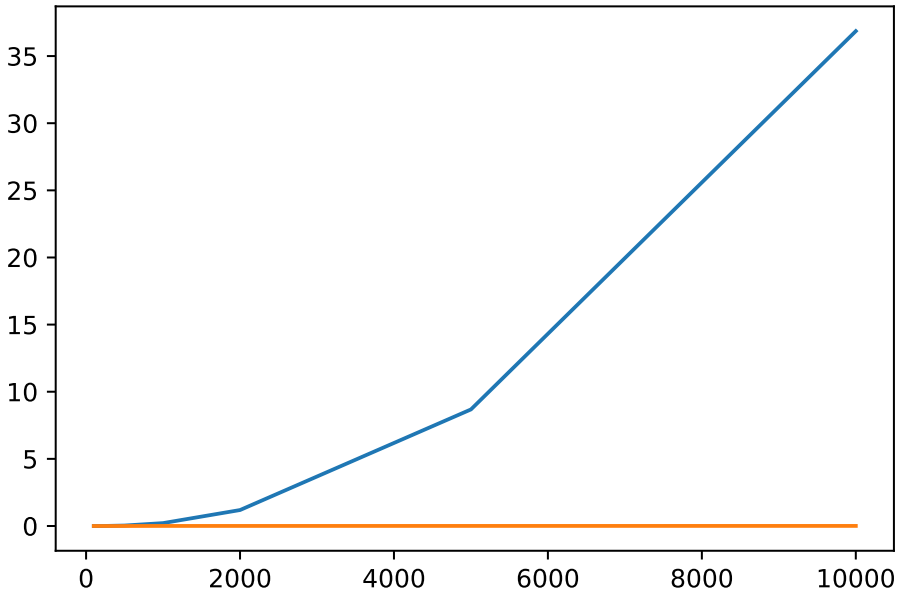
In [45]:
```
plt.plot([100,200,500, 1000, 2000, 5000, 10000],[ usage1,usage2,usage3,usage4,usage5,usage6,usage7])
```

Out[45]: [<matplotlib.lines.Line2D at 0x2646532cc70>]



In [47]:
```
#plt.plot([100,200,500, 1000, 2000, 5000, 10000],[ runtime1,runtime2,runtime3,runtime4,runtime5,runtime6,runtime7])
#plt.plot([100,200,500, 1000, 2000, 5000, 10000],[ usage1,usage2,usage3,usage4,usage5,usage6,usage7])
```

Out[47]: [<matplotlib.lines.Line2D at 0x2646503dee0>]



In [ ]: