In [1]:
```
import random
import time
import tracemalloc
import matplotlib.pyplot as plt
from random import sample
```

In [2]:
```
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

def cost(seq,number):
    start = time.time()
    tracemalloc.start()
    for i in range(number):
        llist.delete_node_at_pos(i)
    end = time.time()
    runtimeIn = end - start
    current, peak = tracemalloc.get_traced_memory()
    usageIn = current / 10**6
    print(f"The current memory usage is: {usageIn} MB; Peak was {peak / 10**6} MB")
    tracemalloc.stop()
    print(f"the runtime is: {runtimeIn} sec")
    return runtimeIn,usageIn
```

In [5]:
```
random.seed(1)
sequence = [i for i in range(100000)]
random.shuffle(sequence)
llist = LinkedList()
for i in range(len(sequence)):
    llist.append(sequence[i])
```

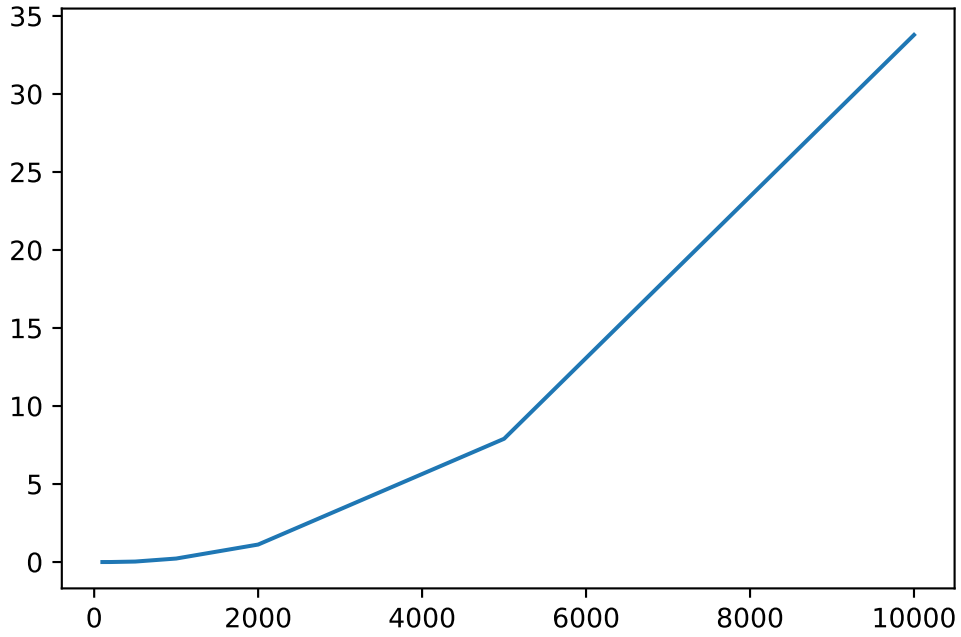
In [14]:
```
lots = []
for i in range(7):
    print(f"{i+1}: For {seqRange[i]} items\n ---------------------")
    lots.append(cost(llist,seqRange[i]))
```

```
1: For 100 items
 --------------------
The current memory usage is: 0.0 MB; Peak was 8e-05 MB
the runtime is: 0.0009930133819580078 sec
2: For 200 items
 --------------------
The current memory usage is: 0.0 MB; Peak was 8e-05 MB
the runtime is: 0.0019884109497070312 sec
3: For 500 items
 --------------------
The current memory usage is: 0.001465 MB; Peak was 0.001553 MB
the runtime is: 0.0317234992980957 sec
4: For 1000 items
 --------------------
The current memory usage is: 0.006054 MB; Peak was 0.00727 MB
the runtime is: 0.22507524490356445 sec
5: For 2000 items
 --------------------
The current memory usage is: 0.006816 MB; Peak was 0.010495 MB
the runtime is: 1.121764898300171 sec
6: For 5000 items
 --------------------
The current memory usage is: 0.006 MB; Peak was 0.00967 MB
the runtime is: 7.9018049240112305 sec
7: For 10000 items
 --------------------
The current memory usage is: 0.00666 MB; Peak was 0.009161 MB
the runtime is: 33.78710744488525 sec
```

In [15]:
```
runtime, usage = map(list, zip(*lots))
print(runtime)
```

```
[0.0009930133819580078, 0.0019884109497070312, 0.0317234992980957, 0.22507524490356445, 1.121764898
300171, 7.9018049240112305, 33.787107944488525]
```

Out[15]:
```
[<matplotlib.lines.Line2D at 0x18d5c9ce100>]
```



In [16]:

Out[16]: [<matplotlib.lines.Line2D at 0x18d5c9a1700>]