# Assignment2Report

February 6, 2019

## 1 CS301 - Assignment 2

### 1.0.1 Lists vs Dictionaries in Python

By Joshua Swick, Evan Minor, and David Vandiver

**Assignment Prompt** Experimentally determine the running time of list and dictionary operations in Python. Write a report explaining what you found, and draw whatever conclusions you can about how lists and dictionaries might be structured ino order to give this behavior. You will probably find it useful to write a function that produces a .csv file of data about the behavior of a given function and then to use Excel to plot the data and find a curve that fits it.

We compared this by measuring the runtime of functions doing similar operations on lists and dictionary data structures of increasing size. The functions are listed below and their runtimes graphed accordingly.

### 1.1 Prepare Data (see ./Assignment2.py)

```
In [1]: from Assignment2 import *
        number_of_elements = 100000
        main()
```

```
In [2]: import matplotlib.pyplot as plt
        import pandas as pd
```

```
In [20]: !ls ./csvs/ ## Raw csv files
```

```
appendFoo_dict.csv                    getFirstElement_list.csv
appendFoo_list.csv                    getLastElement_dict.csv
concatenateFooToEachElement_dict.csv  getLastElement_list.csv
concatenateFooToEachElement_list.csv  lengthOf_dict.csv
countNumOfElements_dict.csv              lengthOf_list.csv
countNumOfElements_list.csv              sort_dict.csv
getFirstElement_dict.csv              sort_list.csv
```

## 1.2 Compare Data

### 1.2.1 Append element to data structure

#### Dictionary

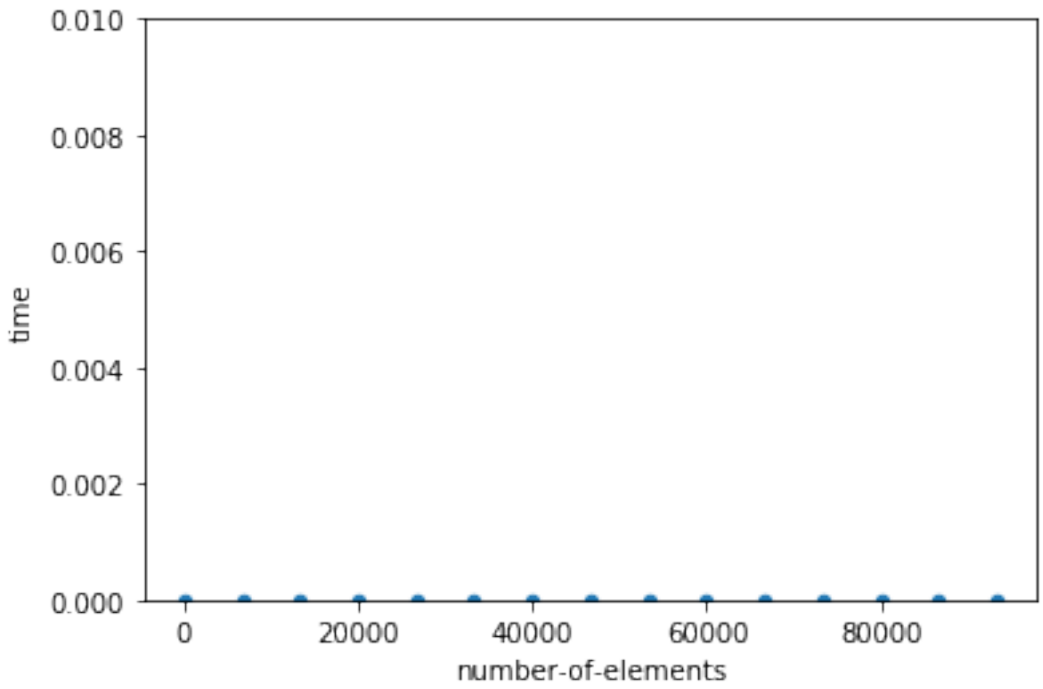
```
In [6]: file = "csvs/appendFoo_dict.csv"
        names = ['number-of-elements','time']
        dataset1 = pd.read_csv(file,names=names)
```

```
In [7]: print(dataset1)
```

```
    number-of-elements      time
0                    2  0.000004
1                 6668  0.000005
2                13334  0.000004
3                20000  0.000002
4                26666  0.000010
5                33332  0.000004
6                39998  0.000005
7                46664  0.000003
8                53330  0.000005
9                59996  0.000005
10               66662  0.000004
11               73328  0.000003
12               79994  0.000006
13               86660  0.000003
14               93326  0.000008
```

```
In [17]: ax1 = dataset1.plot(kind='scatter',x='number-of-elements',y='time')
         ax1.set_ylim(0,.01)
```

```
Out[17]: (0, 0.01)
```

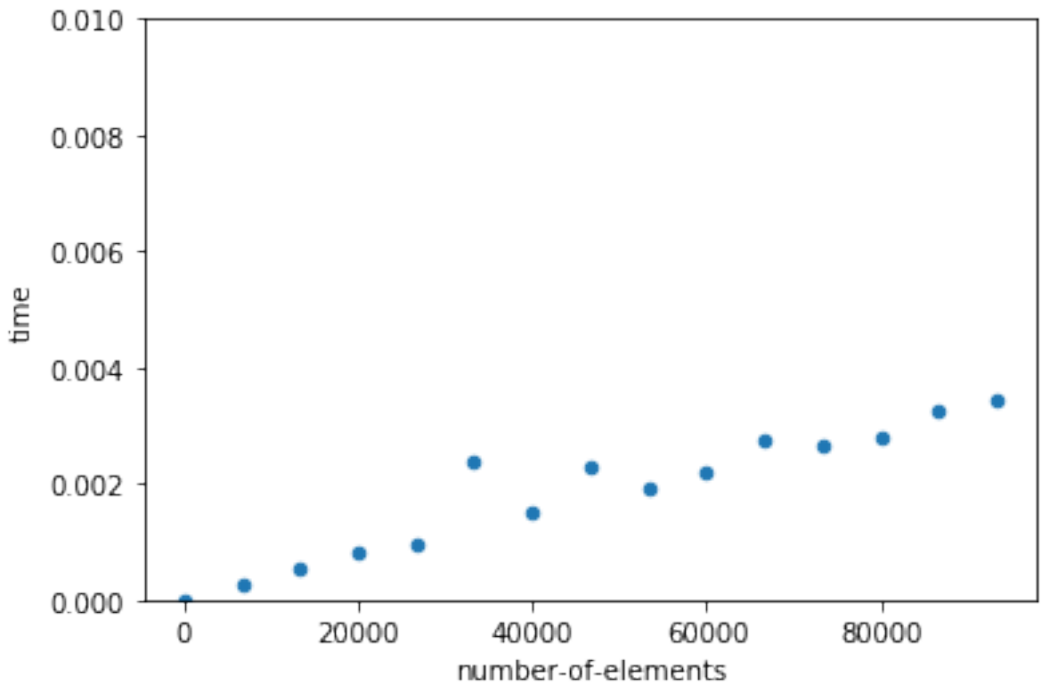



**List**

```
In [ ]: file = "csvs/appendFoo_list.csv"
        names = ['number-of-elements','time']
        dataset2 = pd.read_csv(file,names=names)
```

```
In [21]: ax2 = dataset2.plot(kind='scatter',x='number-of-elements',y='time')
         ax2.set_ylim(0,.01)
```

```
Out[21]: (0, 0.01)
```




**Explaination** As of Python 3.7 dictionaries are ordered by insertion. The equivelent of appending to list would be to insert the element by key value pair as usual. This is a constant function.

Python must check each element in the list to find the end as the operation is linear.

### 1.2.2 Concatenate 'foo' to Each Element in the Data Structure

**Dictionary**

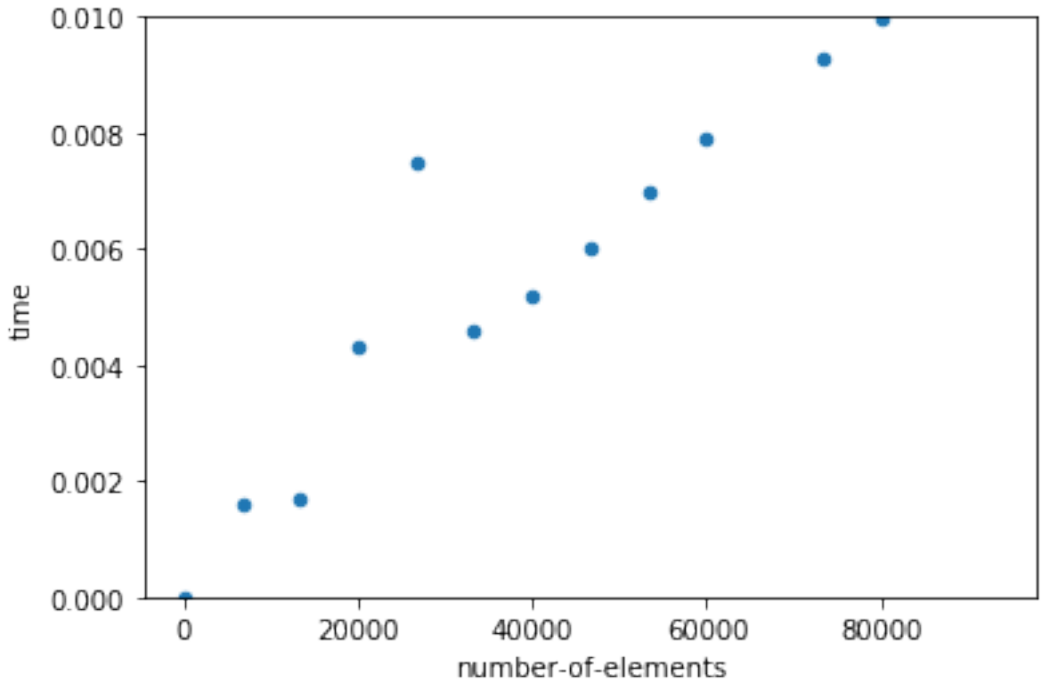
```
In [23]: file = "csvs/concatenateFooToEachElement_dict.csv"
         dataset3 = pd.read_csv(file,names=names)
         print(dataset3)
```

```
    number-of-elements      time
0                    1  0.000007
1                 6667  0.001608
2                13333  0.001697
3                19999  0.004292
4                26665  0.007481
5                33331  0.004578
6                39997  0.005165
7                46663  0.006000
8                53329  0.006974
9                59995  0.007886
10               66661  0.010344
11               73327  0.009301
```

```
12                79993  0.009957
13                86659  0.012107
14                93325  0.011796
```

```
In [24]: ax3 = dataset3.plot(kind='scatter',x='number-of-elements',y='time')
         ax3.set_ylim(0,.01)
```

```
Out[24]: (0, 0.01)
```




**List**

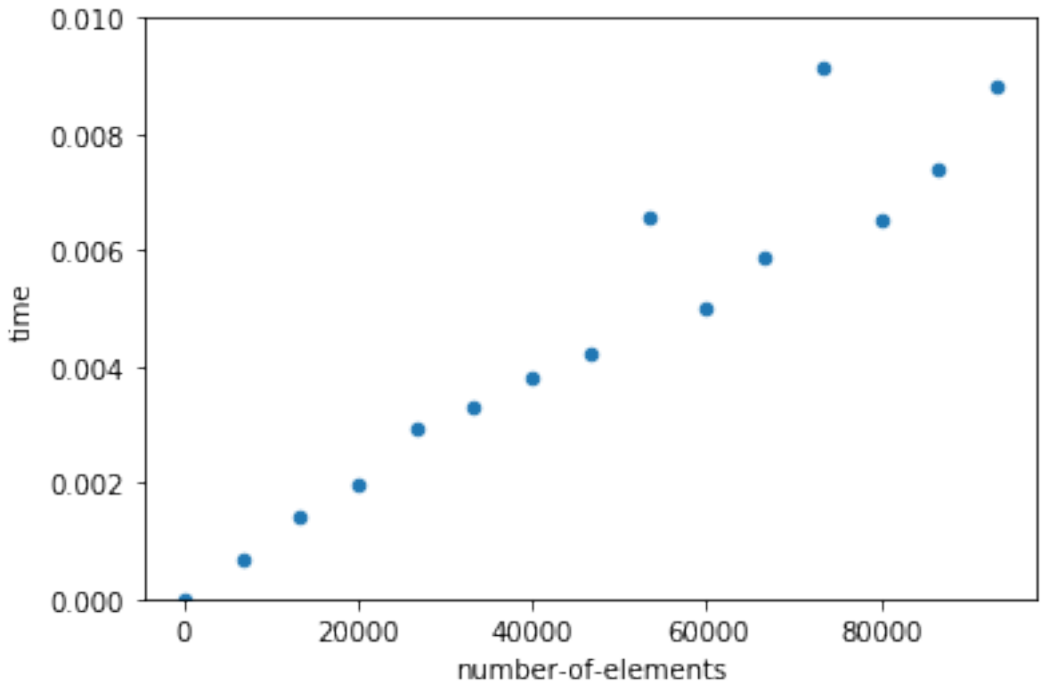
```
In [26]: file = "csvs/concatenateFooToEachElement_list.csv"
         dataset4 = pd.read_csv(file,names=names)
         print(dataset4)
```

```
    number-of-elements      time
0                    2  0.000005
1                 6668  0.000675
2                13334  0.001436
3                20000  0.001974
4                26666  0.002938
5                33332  0.003309
6                39998  0.003818
7                46664  0.004208
```

```
8                53330  0.006549
9                59996  0.005000
10               66662  0.005872
11               73328  0.009136
12               79994  0.006535
13               86660  0.007375
14               93326  0.008838
```

```
In [27]: ax4 = dataset4.plot(kind='scatter',x='number-of-elements',y='time')
         ax4.set_ylim(0,.01)
```

```
Out[27]: (0, 0.01)
```




**Explanation** Python must iterate through each element in the data structure which is a linear operation.

### 1.2.3 Count the Number of Elements in the Data Structure

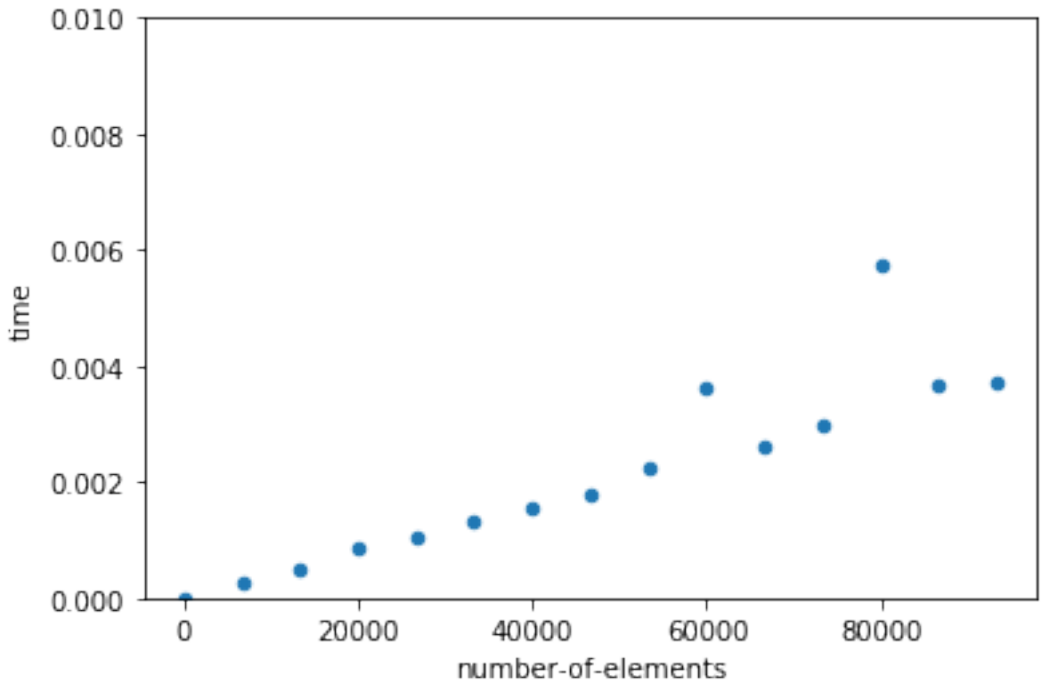
**Dictionary**

```
In [28]: file = "csvs/countNumOfElements_dict.csv"
         dataset5 = pd.read_csv(file,names=names)
         print(dataset5)
```

| | number-of-elements | time |
|---|---|---|
| 0 | 1 | 0.000009 |
| 1 | 6667 | 0.000258 |
| 2 | 13333 | 0.000508 |
| 3 | 19999 | 0.000863 |
| 4 | 26665 | 0.001042 |
| 5 | 33331 | 0.001315 |
| 6 | 39997 | 0.001568 |
| 7 | 46663 | 0.001803 |
| 8 | 53329 | 0.002257 |
| 9 | 59995 | 0.003628 |
| 10 | 66661 | 0.002611 |
| 11 | 73327 | 0.002971 |
| 12 | 79993 | 0.005750 |
| 13 | 86659 | 0.003660 |
| 14 | 93325 | 0.003728 |

In [29]:
```
ax5 = dataset5.plot(kind='scatter',x='number-of-elements',y='time')
ax5.set_ylim(0,.01)
```

Out[29]: (0, 0.01)



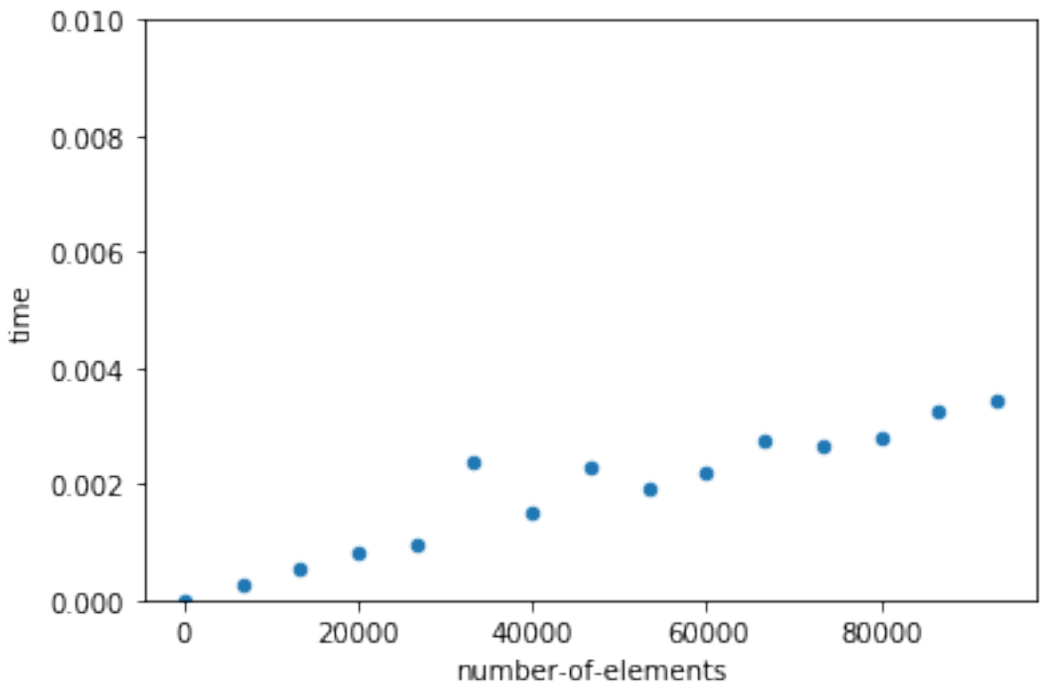

**List**

```
In [30]: file = "csvs/countNumOfElements_list.csv"
         dataset6 = pd.read_csv(file,names=names)
         print(dataset6)
```

```
    number-of-elements      time
0                    2  0.000002
1                 6668  0.000266
2                13334  0.000545
3                20000  0.000815
4                26666  0.000960
5                33332  0.002403
6                39998  0.001511
7                46664  0.002301
8                53330  0.001911
9                59996  0.002205
10               66662  0.002745
11               73328  0.002674
12               79994  0.002807
13               86660  0.003237
14               93326  0.003460
```

```
In [31]: ax6 = dataset6.plot(kind='scatter',x='number-of-elements',y='time')
         ax6.set_ylim(0,.01)
```

```
Out[31]: (0, 0.01)
```



**Explaination** Python must iterate through each element in the data structure while counting, which is a linear process.

### 1.2.4 Get the First Element in the Data Structure

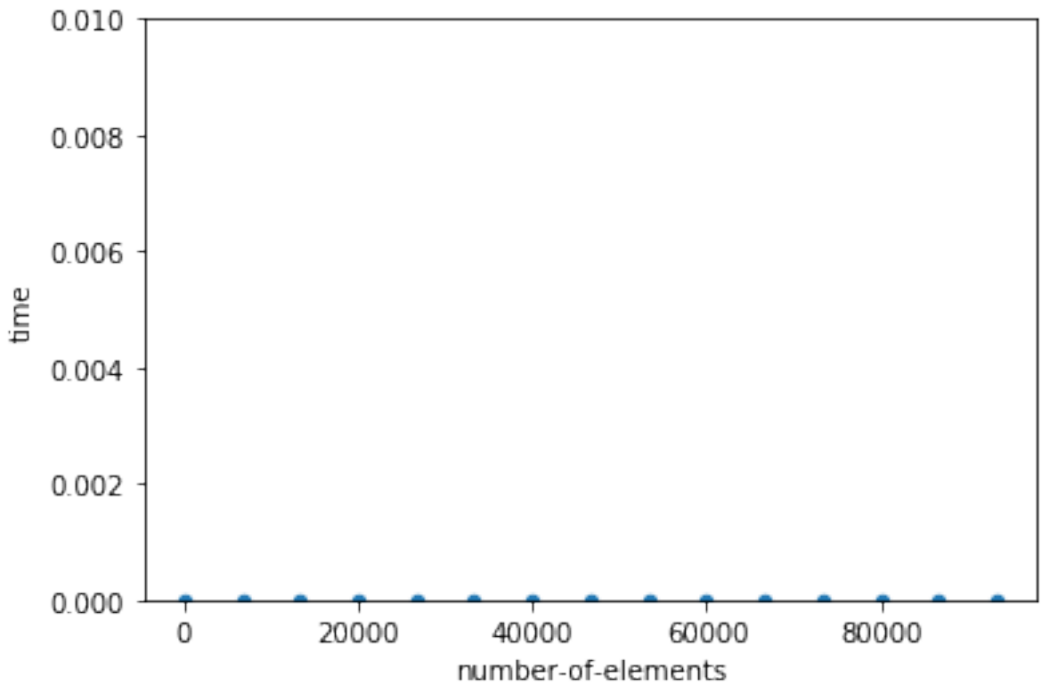
**Dictionary**

```
In [32]: file = "csvs/getFirstElement_dict.csv"
         dataset7 = pd.read_csv(file,names=names)
         print(dataset7)
```

```
    number-of-elements      time
0                    1  0.000002
1                 6667  0.000003
2                13333  0.000003
3                19999  0.000001
4                26665  0.000003
5                33331  0.000002
6                39997  0.000003
7                46663  0.000002
8                53329  0.000005
9                59995  0.000004
10               66661  0.000003
11               73327  0.000002
12               79993  0.000002
13               86659  0.000004
14               93325  0.000003
```

```
In [33]: ax7 = dataset7.plot(kind='scatter',x='number-of-elements',y='time')
         ax7.set_ylim(0,.01)
```

```
Out[33]: (0, 0.01)
```



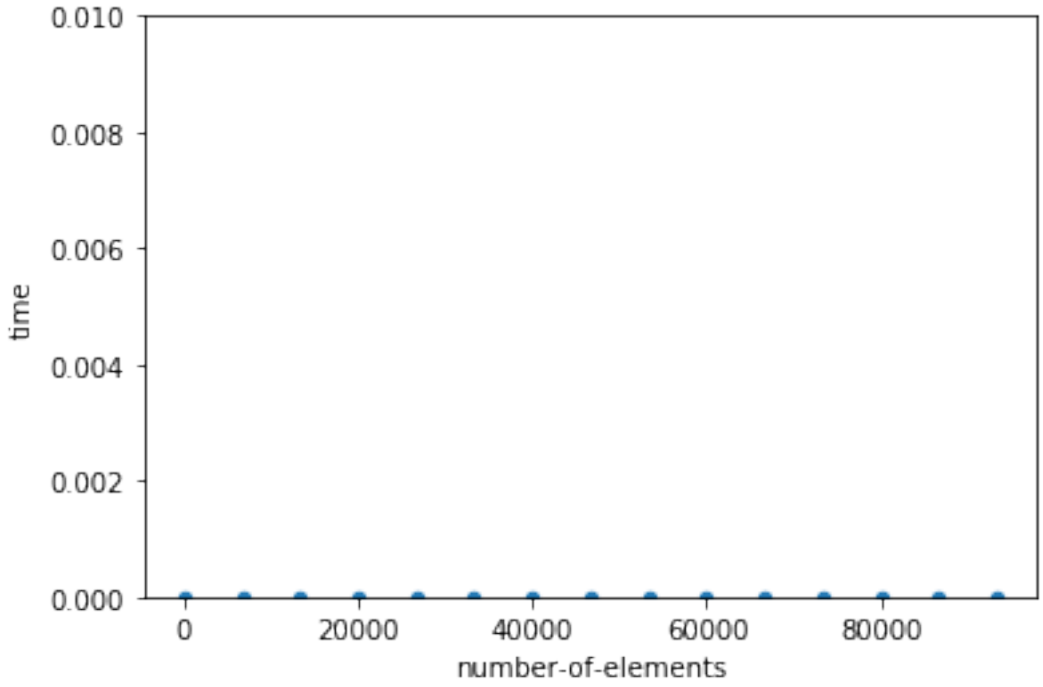

**List**

```
In [34]: file = "csvs/getFirstElement_list.csv"
         dataset8 = pd.read_csv(file,names=names)
         print(dataset8)
```

```
    number-of-elements      time
0                    2  0.000002
1                 6668  0.000001
2                13334  0.000001
3                20000  0.000002
4                26666  0.000002
5                33332  0.000001
6                39998  0.000002
7                46664  0.000002
8                53330  0.000002
9                59996  0.000005
10               66662  0.000002
11               73328  0.000002
12               79994  0.000002
13               86660  0.000002
14               93326  0.000003
```

```
In [35]: ax8 = dataset8.plot(kind='scatter',x='number-of-elements',y='time')
         ax8.set_ylim(0,.01)
```

```
Out[35]: (0, 0.01)
```




**Explaination** The first element of a data structure is always in the same index, hense a constant operation.

### 1.2.5 Get the Last Element of the Data Structure

**Dictionary**

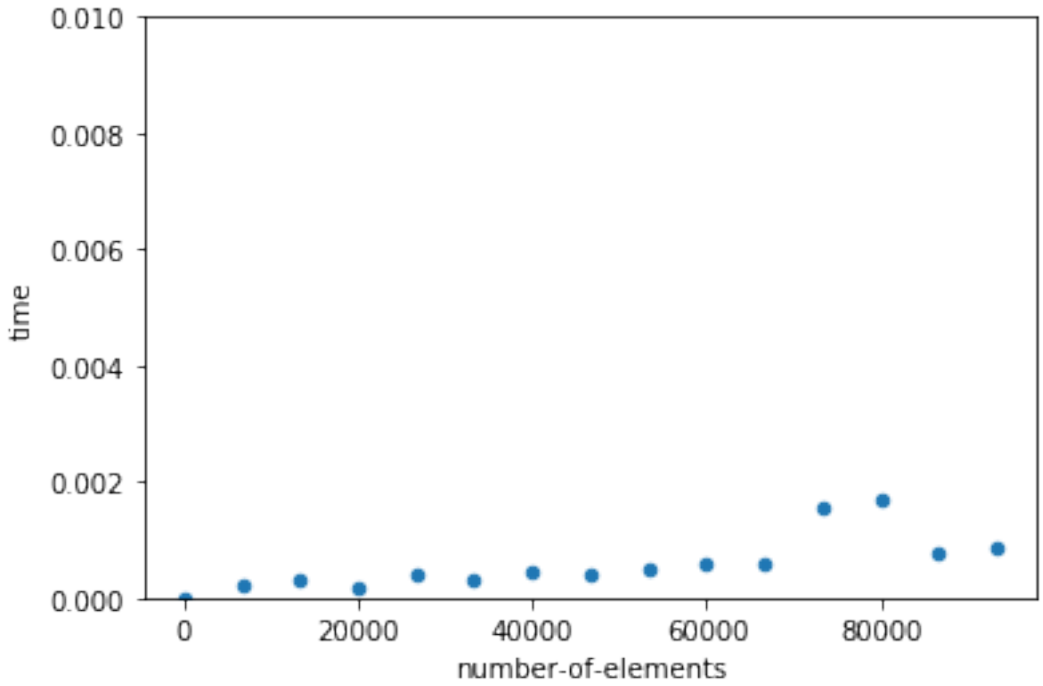
```
In [37]: file = "csvs/getLastElement_dict.csv"
         dataset9 = pd.read_csv(file,names=names)
         print(dataset9)
```

```
    number-of-elements      time
0                    1  0.000011
1                 6667  0.000214
2                13333  0.000294
3                19999  0.000165
4                26665  0.000387
5                33331  0.000310
6                39997  0.000462
7                46663  0.000419
8                53329  0.000481
9                59995  0.000587
10               66661  0.000592
```

```
11                73327  0.001571
12                79993  0.001695
13                86659  0.000772
14                93325  0.000878
```

```
In [39]: ax9 = dataset9.plot(kind='scatter',x='number-of-elements',y='time')
         ax9.set_ylim(0,.01)
```

```
Out[39]: (0, 0.01)
```




**List**

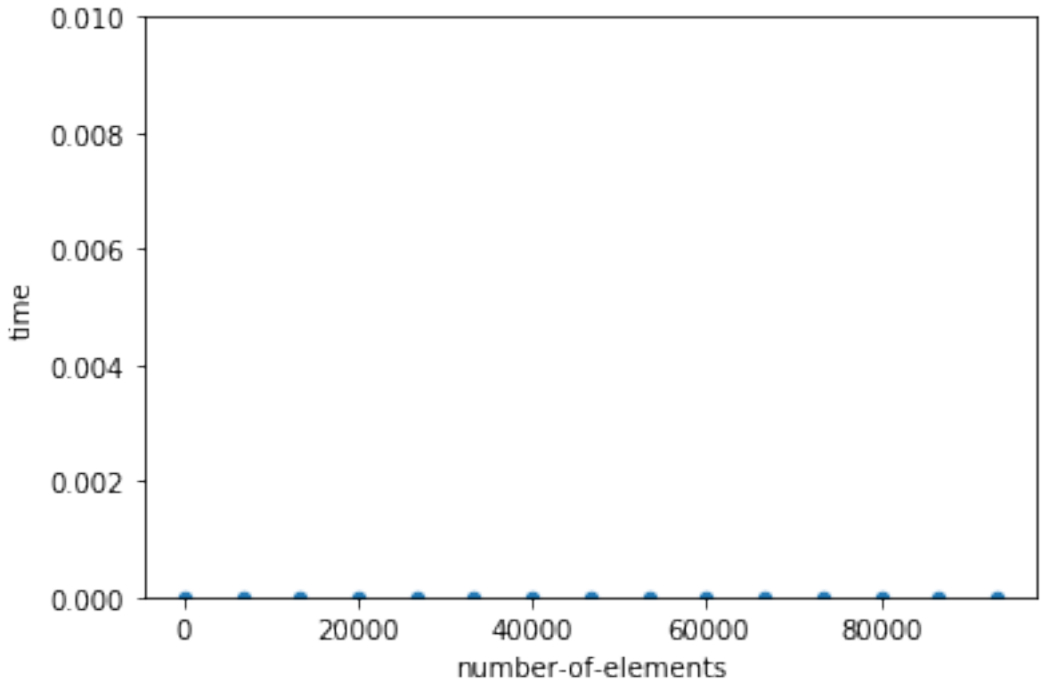
```
In [38]: file = "csvs/getLastElement_list.csv"
         dataset10 = pd.read_csv(file,names=names)
         print(dataset10)
```

```
    number-of-elements          time
0                    2  1.668930e-06
1                 6668  1.192093e-06
2                13334  1.668930e-06
3                20000  2.384186e-06
4                26666  9.536743e-07
5                33332  1.907349e-06
6                39998  1.668930e-06
```

```
7                46664  1.907349e-06
8                53330  1.668930e-06
9                59996  2.622604e-06
10               66662  2.145767e-06
11               73328  5.245209e-06
12               79994  1.430511e-06
13               86660  2.145767e-06
14               93326  2.384186e-06
```

```
In [40]: ax10 = dataset10.plot(kind='scatter',x='number-of-elements',y='time')
         ax10.set_ylim(0,.01)
```

```
Out[40]: (0, 0.01)
```




**Explaination** List looks constant, dictionary is linear?

### 1.2.6 Get the Length of the Data Structure

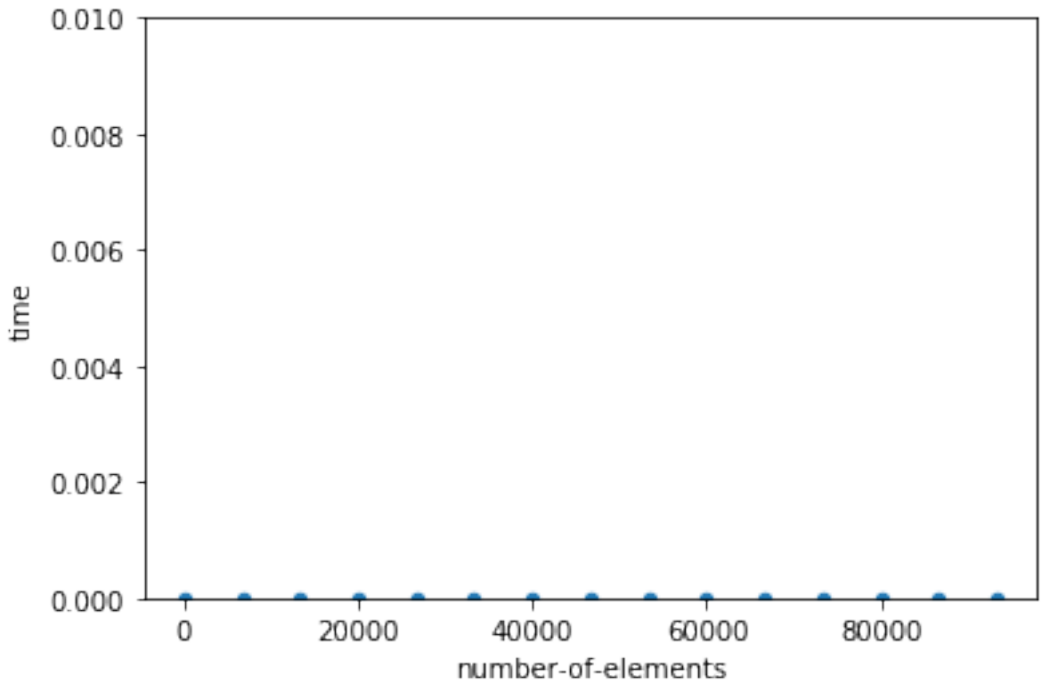
**Dictionary**

```
In [41]: file = "csvs/lengthOf_dict.csv"
         dataset11 = pd.read_csv(file,names=names)
         print(dataset11)
```

| | number-of-elements | time |
|---|---|---|
| 0 | 1 | 0.000003 |
| 1 | 6667 | 0.000006 |
| 2 | 13333 | 0.000004 |
| 3 | 19999 | 0.000003 |
| 4 | 26665 | 0.000003 |
| 5 | 33331 | 0.000002 |
| 6 | 39997 | 0.000004 |
| 7 | 46663 | 0.000004 |
| 8 | 53329 | 0.000004 |
| 9 | 59995 | 0.000004 |
| 10 | 66661 | 0.000004 |
| 11 | 73327 | 0.000003 |
| 12 | 79993 | 0.000003 |
| 13 | 86659 | 0.000003 |
| 14 | 93325 | 0.000004 |

In [43]:
```
ax11 = dataset11.plot(kind='scatter',x='number-of-elements',y='time')
ax11.set_ylim(0,.01)
```

Out[43]: (0, 0.01)



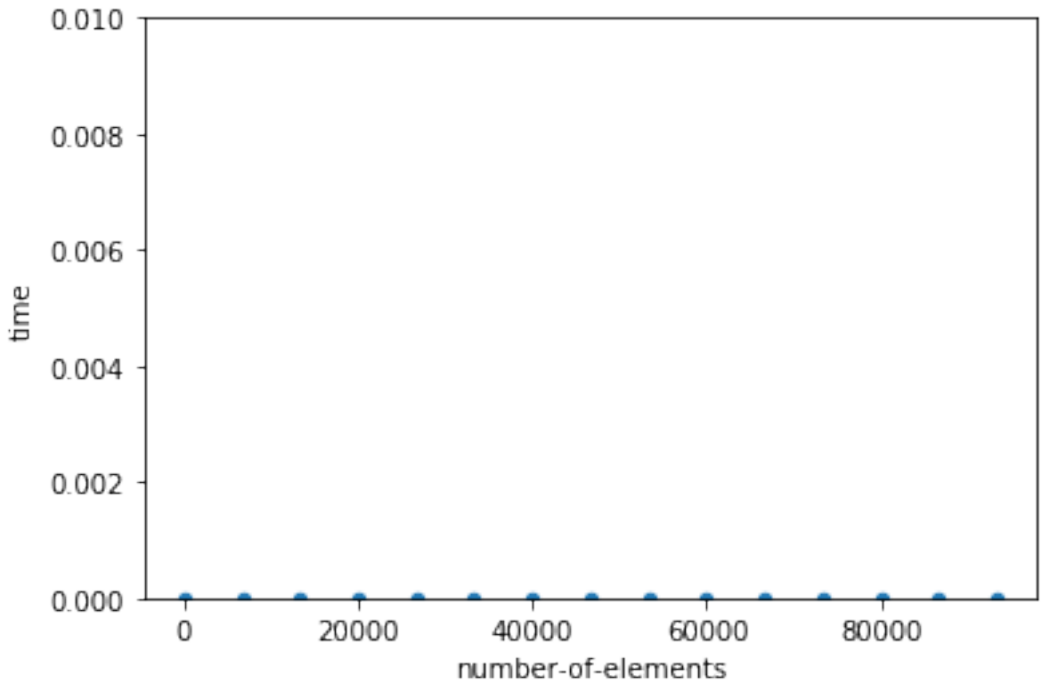

**Lists**

```
In [42]: file = "csvs/lengthOf_list.csv"
         dataset12 = pd.read_csv(file,names=names)
         print(dataset12)
```

```
    number-of-elements          time
0                    2  1.668930e-06
1                 6668  7.152557e-07
2                13334  1.192093e-06
3                20000  2.384186e-06
4                26666  3.576279e-06
5                33332  2.622604e-06
6                39998  3.814697e-06
7                46664  3.099442e-06
8                53330  2.145767e-06
9                59996  2.861023e-06
10               66662  3.099442e-06
11               73328  4.053116e-06
12               79994  2.861023e-06
13               86660  2.861023e-06
14               93326  3.337860e-06
```

```
In [44]: ax12 = dataset12.plot(kind='scatter',x='number-of-elements',y='time')
         ax12.set_ylim(0,.01)
```

```
Out[44]: (0, 0.01)
```



**Explaination** Getting the length of a list or dictionary is a constant operation, is the length saved as meta data or easily calculated?

### 1.2.7 Sort the Data Structure

**Dictionary** Dictionaries cannot be ordered by key, value elements.

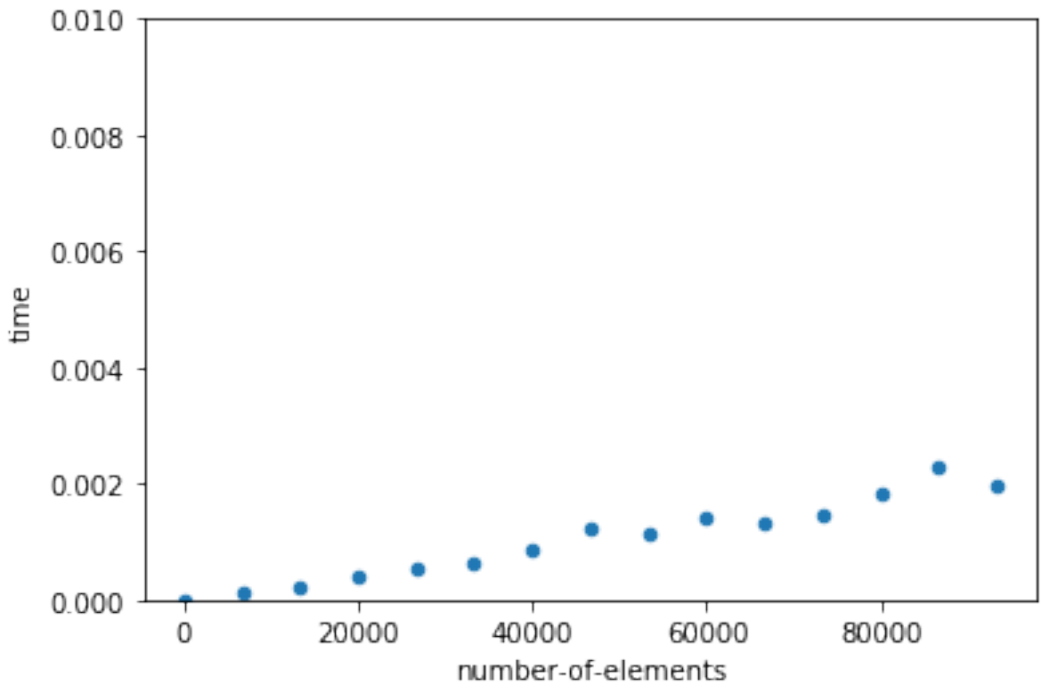
**List**

```
In [47]: file = "csvs/sort_list.csv"
         dataset14 = pd.read_csv(file,names=names)
         print(dataset14)
```

```
    number-of-elements      time
0                    2  0.000002
1                 6668  0.000126
2                13334  0.000237
3                20000  0.000422
4                26666  0.000553
5                33332  0.000641
6                39998  0.000846
7                46664  0.001221
8                53330  0.001121
9                59996  0.001438
10               66662  0.001315
11               73328  0.001483
12               79994  0.001853
13               86660  0.002292
14               93326  0.001969
```

```
In [53]: ax14 = dataset14.plot(kind='scatter',x='number-of-elements',y='time')
         ax14.set_ylim(0,.01)
```

```
Out[53]: (0, 0.01)
```




**Explaination** Sorting of a list is linear.

### 1.2.8 Summary

Operations are either constant, when independent of the number of elements in a data structure, or linear, when the operation is dependent on the number of elements in the structure.