

②

ソートの計算時間

ソートの計算時間を測ってみましょう。

時間の測り方

- 前回までの講義で、`%%time` を使って時間を計測しましたが、他にもいくつかの時間の計測方法があります
- ここでは、次の手順も覚えましょう
 - `%time`
 - `%%time`
 - `%timeit`
 - `%%timeit`

%%time と %time

- **%%time** : これまで学習したように、をセルの冒頭に追加すると、そのセル全体の実行時間を計測することができます
- **%time** : 行の先頭に追加すると、その行の実行時間を計測することができます
- 以下を試してみましょう

```
%%time
```

```
for i in range(10):  
    sorted([5, 2, 3, 4, 1, 9, 10, 8, 7, 6])
```

計測対象

```
for i in range(10):
```

```
    %time sorted([5, 2, 3, 4, 1, 9, 10, 8, 7, 6])
```

計測対象

%%timeit と %timeit

- `time` を `timeit` に変更すると、複数回繰り返したときの平均時間を計測することができます
 - `%%timeit` と `%timeit` の違いは、`time` の場合と同様です
- 以下を試してみましょう

```
%%timeit  
for i in range(10):  
    sorted([5, 2, 3, 4, 1, 9, 10, 8, 7, 6])
```

計測対象

```
for i in range(10):  
    %timeit sorted([5, 2, 3, 4, 1, 9, 10, 8, 7, 6])
```

計測対象

sorted の時間を計測する

- ここでは、以前と同様に **random** モジュールを使って、一定数の長いリストを作ること考えます
 - **random.sample** 関数で、引数で与えた範囲から、パラメータ **k** の数をランダムに選びます
- 以下の例では、0 以上 10000000 未満の値からなる、長さ10のリストを作っています

```
import random
```

```
target = random.sample(range(10000000), 10)  
target
```

```
[881705, 279847, 254231, 573174, 99131, 567593, 501560, 761901, 593533, 287058]
```

`sorted` の時間を計測する

- 先ほど学習した `%timeit` を用いて、長さ1000のリストのソートにかかる時間を計測してみます
- 以下のようにすることで、リストを作る時間は除外し、ソートにかかる時間のみを計測できます

```
target = random.sample(range(1000000), 1000)
%timeit sorted(target)
```

sorted の時間を計測する

- 同じ手順で、リストの長さを、1000 から 10000 まで 1000 刻みで増やして、時間を計測してみましょう

```
for i in range(1, 11):  
    target = random.sample(range(1000000), i * 1000)  
    %timeit sorted(target)
```

sorted の時間を計測する

- その結果を matplotlib を使ってプロットしてみます
 - 計測時間は、自分の計測結果を入れてください

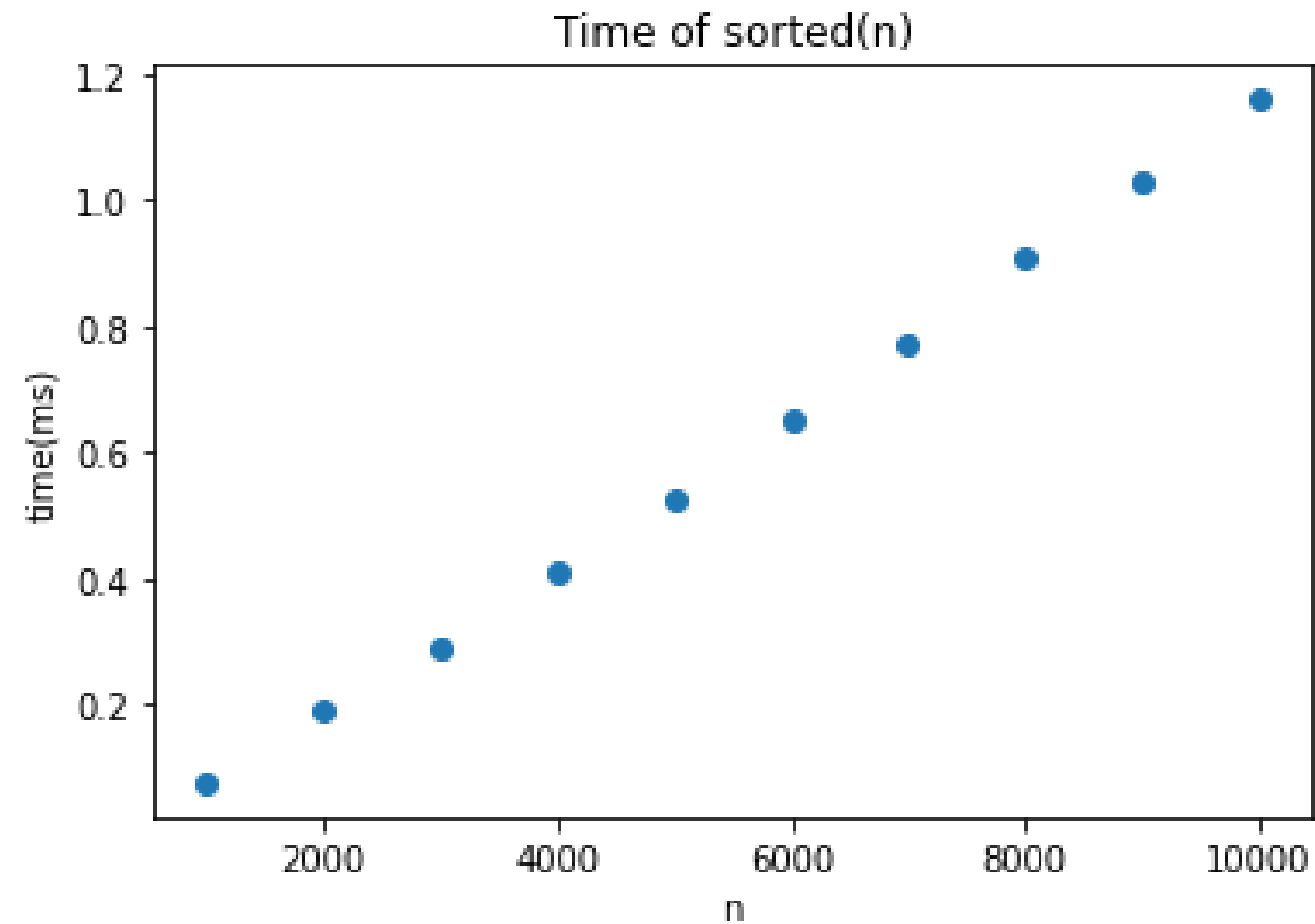
```
import matplotlib.pyplot as plt
import math
```

```
xs = [1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000]
ys = [0.0751, 0.188, 0.290, 0.408, 0.524, 0.653, 0.770, 0.907, 1.03, 1.16]
```

```
plt.plot(xs, ys, 'o')
plt.xlabel('n')
plt.ylabel('time(ms)')
plt.title('Time of sorted(n)')
plt.show()
```


以下のようにになりましたか？

- 以下のようなグラフが得られましたか？



一見直線的なグラフに見えますが、**理論的には少し違います**

ソートの計算量

- 実は、Pythonのソートの平均計算量は： $O(n \log n)$
 - Pythonのソートは「テームソート」というアルゴリズムを採用しているようです
 - このアルゴリズムに限らず、世の中の実用的なソートアルゴリズムの多くは、このオーダーになります
- 線形よりも若干上に反った増え方になります（準線形時間）

