

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%pylab inline
```

Populating the interactive namespace from numpy and matplotlib

$$M = \frac{N}{P}$$

```
In [3]: parallel = pd.read_csv('stats.txt',
                             names=['T', 'N', 'M', 'P'],
                             sep=" ")
parallel['T'] = list(map((lambda x: float(x[:-1])), parallel['T']))
parallel = parallel.loc[parallel.groupby(['P', 'N'])['T'].idxmin()]
parallel.head()
```

```
Out[3]:
```

	T	N	M	P
0	0.001247	10000	10000	1
1	0.077731	1010000	1010000	1
2	0.133625	2010000	2010000	1
3	0.207314	3010000	3010000	1
4	0.272096	4010000	4010000	1

А у стандартной сортировки по-прежнему возьмем лучший результат.

```
In [4]: qsort = pd.read_csv('qstats.txt',
                             names=['T', 'N'],
                             sep=" ")
qsort['T'] = list(map((lambda x: float(x[:-1])), qsort['T']))
qsort = qsort.loc[qsort.groupby(['N'])['T'].idxmin()]
qsort.head()
```

```
Out[4]:
```

	T	N
90	0.000613	10000
61	0.061280	1010000
92	0.127809	2010000
3	0.186800	3010000
124	0.250233	4010000

```

In [5]: plt.figure(figsize=(15, 10))

for p in [1, 2, 4, 8, 16]:
    plt.plot(parallel[parallel['P'] == p]['N'], np.array(parallel[para
#     plt.scatter(parallel[parallel['P'] == p]['N'], parallel[parallel

plt.plot(qsort['N'], np.array(qsort['T']), alpha=1, label='QSort')
# plt.scatter(qsort['N'], qsort['T'], label='QSort')

plt.title('T(N)', y=1.03, fontsize=22)
plt.xlabel('N', fontsize=16)
plt.ylabel('T', fontsize=16)

plt.legend(fontsize=12)
plt.grid()
plt.show()

```

