



CPE 212 ALGORITHM DESIGN
SORTING PERFORMANCE PROJECT

SUMMITTED TO

Assoc. Prof. PEERAPON SIRIPONGWUTIKORN

BY

- | | |
|----------------------------|-------------|
| 1. SEIYA TAJIMA | 63070503409 |
| 2. NATTAPONG SAENGARUNVONG | 63070503415 |
| 3. PREUTTA WORATOL | 63070503433 |
| 4. MAHUDPOJ PONGPANICH | 63070503443 |
| 5. MAYSON ROSS KIENG | 63070503445 |
| 6. PORNSAWALEE KANJANATITI | 63070503467 |

SEMESTER 1/2021

DEPARTMENT OF COMPUTER ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI

Selection sort

```
PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort
Enter the number of element: 8
Element of the array before sorting
88000 304897 484971 115933 25864 383859 416036 834136

Element of the array after sorting
25864 88000 115933 304897 383859 416036 484971 834136
```

For $n = 8$

```
PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort
Enter the number of element: 15
Element of the array before sorting
873133 356501 428299 646941 302419 923761 960313 650561 460301 503329
662651 221482 263749 7465 99011
Element of the array after sorting
7465 99011 221482 263749 302419 356501 428299 460301 503329 646941
650561 662651 873133 923761 960313
```

For $n = 15$

Comparison of selection sort algorithms

```
Enter the number of element: 50
Time taken for selection sort : 0 seconds
Enter the number of element: 50
Time taken for selection sort : 0 seconds
Enter the number of element: 50
Time taken for selection sort : 0 seconds
Enter the number of element: 50
Time taken for selection sort : 0 seconds
Enter the number of element: 50
Time taken for selection sort : 0 seconds
```

N = 50

[illegible]

```
Enter the number of element: 100
Time taken for selection sort : 0 seconds
Enter the number of element: 100
Time taken for selection sort : 0 seconds
Enter the number of element: 100
Time taken for selection sort : 0 seconds
Enter the number of element: 100
Time taken for selection sort : 0 seconds
Enter the number of element: 100
Time taken for selection sort : 0 seconds
```

N = 100

[illegible]

```
Enter the number of element: 500
Time taken for selection sort : 0.000965 seconds
Enter the number of element: 500
Time taken for selection sort : 0 seconds
Enter the number of element: 500
Time taken for selection sort : 0 seconds
Enter the number of element: 500
Time taken for selection sort : 0.001031 seconds
Enter the number of element: 500
Time taken for selection sort : 0 seconds
```

N = 500

[illegible]

```

PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 1000
Time taken for selection sort : 0.00103 seconds
Enter the number of element: 1000
Time taken for selection sort : 0.000994 seconds
Enter the number of element: 1000
Time taken for selection sort : 0.000997 seconds
Enter the number of element: 1000
Time taken for selection sort : 0.001996 seconds
Enter the number of element: 1000
Time taken for selection sort : 0.001982 seconds

```

N = 1000

Number	Time (seconds)
1000	0.00103
1000	0.000994
1000	0.000997
1000	0.001996
1000	0.001982
Average	0.0013998

```

PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 5000
Time taken for selection sort : 0.03192 seconds
Enter the number of element: 5000
Time taken for selection sort : 0.031957 seconds
Enter the number of element: 5000
Time taken for selection sort : 0.032936 seconds
Enter the number of element: 5000
Time taken for selection sort : 0.032913 seconds
Enter the number of element: 5000
Time taken for selection sort : 0.031917 seconds

```

N = 5000

Number	Time (seconds)
5000	0.03192
5000	0.031957
5000	0.032936
5000	0.032913
5000	0.031917
Average	0.0323286

```

PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 10000
Time taken for selection sort : 0.134181 seconds
Enter the number of element: 10000
Time taken for selection sort : 0.135185 seconds
Enter the number of element: 10000
Time taken for selection sort : 0.136667 seconds
Enter the number of element: 10000
Time taken for selection sort : 0.143615 seconds
Enter the number of element: 10000
Time taken for selection sort : 0.126643 seconds

```

N = 10000

Number	Time (seconds)
10000	0.134181
10000	0.135185
10000	0.136667
10000	0.143615
10000	0.126643
Average	0.1352582

```

PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 50000
Time taken for selection sort : 3.08826 seconds
Enter the number of element: 50000
Time taken for selection sort : 3.03246 seconds
Enter the number of element: 50000
Time taken for selection sort : 3.06398 seconds
Enter the number of element: 50000
Time taken for selection sort : 3.12607 seconds
Enter the number of element: 50000
Time taken for selection sort : 3.18301 seconds

```

N = 50000

Number	Time (seconds)
50000	3.08826
50000	3.03246
50000	3.06398
50000	3.12607
50000	3.18301
Average	3.098756

```

PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 100000
Time taken for selection sort : 12.1497 seconds
Enter the number of element: 100000
Time taken for selection sort : 12.7409 seconds
Enter the number of element: 100000
Time taken for selection sort : 12.4713 seconds
Enter the number of element: 100000
Time taken for selection sort : 12.3194 seconds
Enter the number of element: 100000
Time taken for selection sort : 12.5124 seconds

```

N = 100000

Number	Time (seconds)
100000	12.1497
100000	12.7409
100000	12.4713
100000	12.3194
100000	12.5124
Average	12.43874

```

PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 500000
Time taken for selection sort : 319.956 seconds
Enter the number of element: 500000
Time taken for selection sort : 314.915 seconds
Enter the number of element: 500000
Time taken for selection sort : 314.972 seconds
Enter the number of element: 500000
Time taken for selection sort : 315.22 seconds
Enter the number of element: 500000
Time taken for selection sort : 327.931 seconds

```

N = 500000

Number	Time (seconds)
500000	319.956
500000	314.915
500000	314.972
500000	315.22
500000	327.931
Average	318.5988

```

PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 1000000
Time taken for selection sort : 1220.06 seconds
Enter the number of element: 1000000
Time taken for selection sort : 1207.39 seconds
Enter the number of element: 1000000
Time taken for selection sort : 1190.94 seconds
Enter the number of element: 1000000
Time taken for selection sort : 1189.83 seconds
Enter the number of element: 1000000
Time taken for selection sort : 1184.7 seconds

```

N = 1000000

Number	Time (seconds)
1000000	1220.06
1000000	1207.39
1000000	1190.94
1000000	1189.83
1000000	1184.7
Average	1198.584

```
PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 5000000

Time taken for selection sort : 33769.5 seconds
```

```
PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 5000000

Time taken for selection sort : 37375.2 seconds
```

```
PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 5000000

Time taken for selection sort : 32566.8 seconds
```

```
PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 5000000

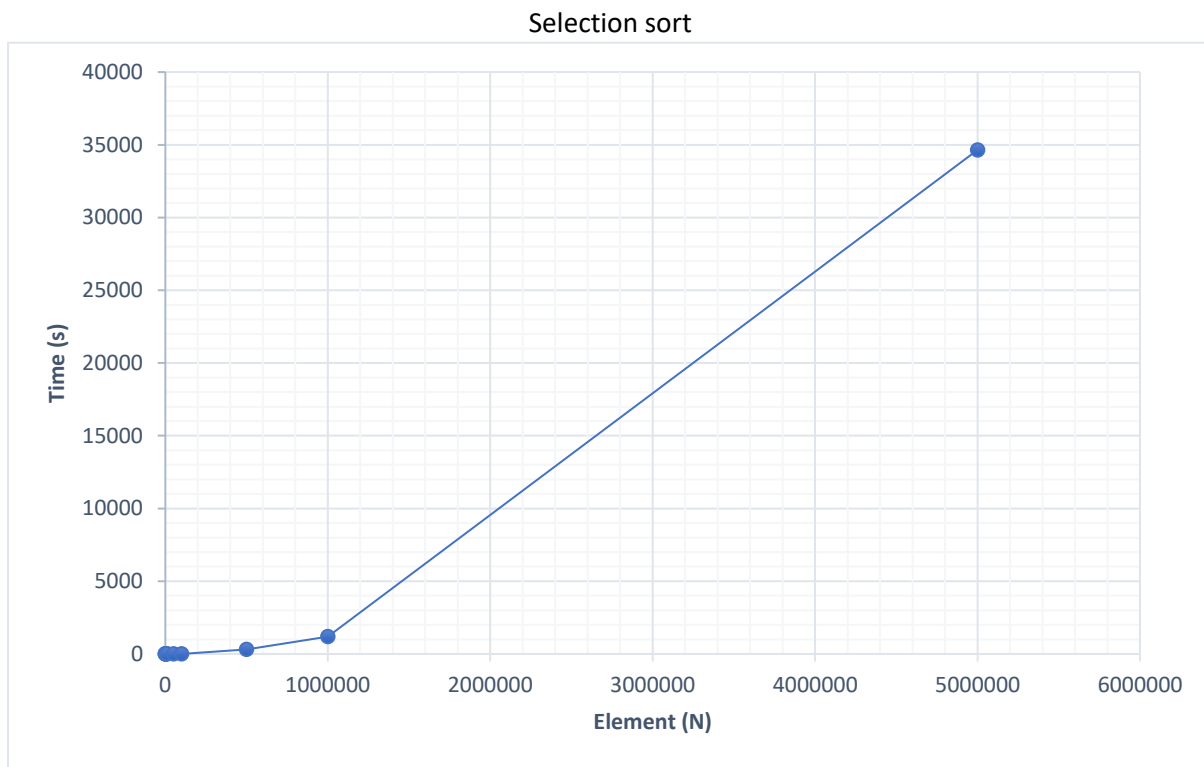
Time taken for selection sort : 33828.3 seconds
```

```
PS C:\Users\HP\Desktop\Work\Visual_Studio> ./selection_sort_time
Enter the number of element: 5000000

Time taken for selection sort : 35705.6 seconds
```

N = 5000000

Number	Time (seconds)
5000000	33769.5
5000000	37375.2
5000000	32566.8
5000000	33828.3
5000000	35705.6
Average	34649.08



Result : We see that the selection sort makes the same number of comparisons as the bubble sort and is therefore also $O(n^2)$. However, due to the increment in the number of exchanges, the selection sort typically executes slower in benchmark studies. In fact, for our list, the selection sort takes more than 30000 seconds to sort 5000000 numbers.

Insertion Sort

1)

N = 8

```
Enter the number of elemnt size 8
Elements of the array before sorting
2949 10858 12291 1551 26191 6544 21051 1443
Elements of the array after sorting
1443 1551 2949 6544 10858 12291 21051 26191
Running time is 0 milliseconds
```

N= 15

```
Enter the number of elemnt size 15
Elements of the array before sorting
3070 15333 17905 7158 14615 20761 5139 24206 26515 27413 15631 18706 6370 8411 27112
Elements of the array after sorting
3070 5139 6370 7158 8411 14615 15333 15631 17905 18706 20761 24206 26515 27112 27413
Running time is 0 milliseconds
```

2a) N = 50

50	0 milli
50	0 milli
50	0 milli
50	0 milli
50	0 milli
Average time	0 milli (0 Second)

N = 100

100	0 milli
100	0 milli
100	0 milli
100	0 milli
100	0 milli
Average time	0 milli (0 Second)

N = 500

500	0 milli
500	0 milli
500	0 milli
500	0 milli
500	0 milli
Average time	0 milli (0 Second)

N = 1000

1000	1 milli
1000	0 milli
1000	0 milli
1000	0 milli
1000	1 milli
Average time	0.4 milli (0.004 Second)

N = 5000

5000	22 milli
5000	25 milli
5000	22 milli
5000	20 milli
5000	21 milli
Average time	22 milli (0.022 Second)

N = 10000

10,000	86 milli
10,000	86 milli
10,000	108 milli
10,000	83 milli
10,000	89 milli
Average time	90.4 milli (0.0904 Second)

N = 50000

50,000	2139 milli
50,000	2213 milli
50,000	2199 milli
50,000	2190 milli
50,000	2128 milli
Average time	2173.8 milli (2.1738 Second)

N = 100000

100,000	9093 milli
100,000	9881 milli
100,000	9230 milli
100,000	8427 milli
100,000	8592 milli
Average time	9044.6 milli (9.0446 Second)

N = 500000

500,000	252,157 milli
500,000	240,395 milli
500,000	222,417 milli
500,000	234,119 milli
500,000	262,398 milli
Average time	242297 milli (242.3 Second)

N = 1000000

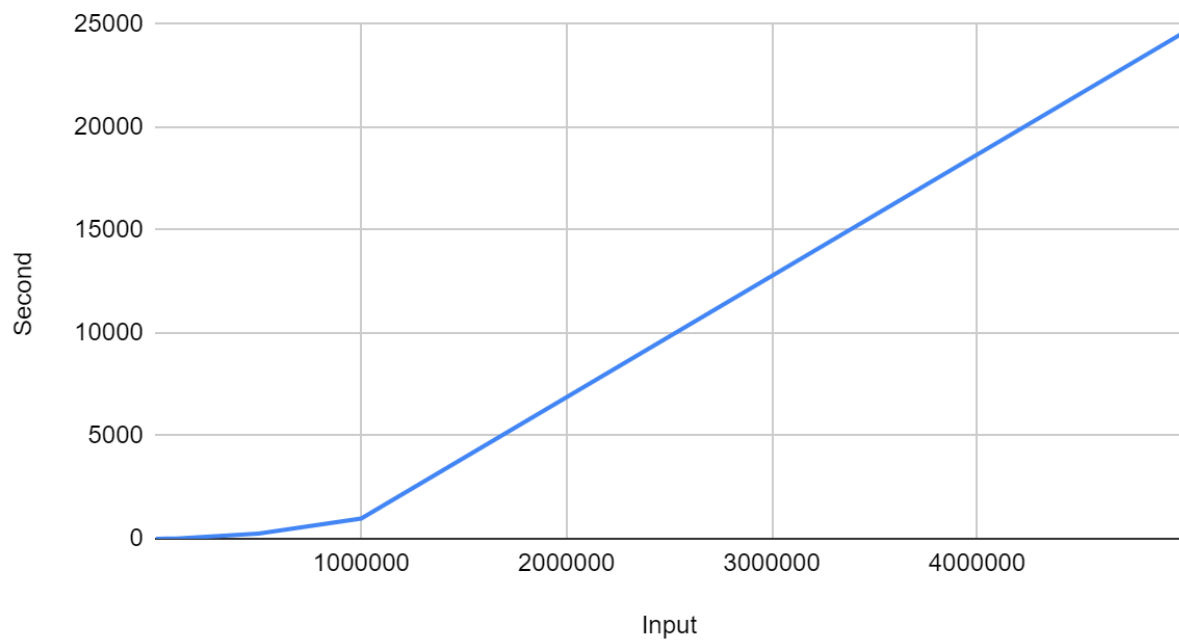
1,000,000	964,569 milli
1,000,000	979,718 milli
1,000,000	956,064 milli
1,000,000	1.03409×10^6 milli
1,000,000	952,992 milli
Average time	977,486.6 milli (977.49 Second)

N = 5000000

5,000,000	2.42438×10^7 milli
5,000,000	2.23015×10^7 milli
5,000,000	2.49733×10^7 milli
5,000,000	2.50898×10^7 milli
5,000,000	2.62046×10^7 milli
Average time	24562600 milli (24562.6 Second)

2b)

Second vs. Input



Insertion Sort result: Big O notation of insertion sort is $O(n^2)$ so it means that the more input it takes in the more time it will take. As we can see the first 50 and 100 items take about 0 second due to that we measure the time in milliseconds but it takes microseconds to do it. But as we increase the input the graph starts to become an exponential graph since the time it took is double of the input itself.

Merge Sort

```
Enter the number of elements (Enter 0 to stop) : 8
```

Elements of array before sorting

Elements of array after sorting

For $n = 8$

```
Enter the number of elements (Enter 0 to stop) : 15
```

Elements of array before sorting

905445 567196 980260 900634 714416 945698 143087 764305

158520	936050	347893	658375	80107	50995	623651
--------	--------	--------	--------	-------	-------	--------

Elements of array after sorting

714416 764305 900634 905445 936050 945698 980260

For $n = 15$

Comparison of Merge sort algorithm

N = 50

[illegible]

N = 100

Number	Time (seconds)
100	0
100	0
100	0
100	0
100	0
Average	0

N = 500

Number	Time (seconds)
500	0
500	0
500	0
500	0
500	0
Average	0

N = 1000

Number	Time (seconds)
1000	0.001
1000	0
1000	0.001
1000	0
1000	0
Average	0.0004

N = 5000

Number	Time (seconds)
5000	0.001
5000	0.001
5000	0.001
5000	0.001
5000	0.001
Average	0.001

N = 10000

Number	Time (seconds)
10000	0.002
10000	0.002
10000	0.003
10000	0.002
10000	0.002
Average	0.0022

N = 50000

Number	Time (seconds)
50000	0.012
50000	0.011
50000	0.011
50000	0.01
50000	0.012
Average	0.0112

N = 100000

Number	Time (seconds)
100000	0.022
100000	0.022
100000	0.023
100000	0.022
100000	0.022
Average	0.0222

N = 500000

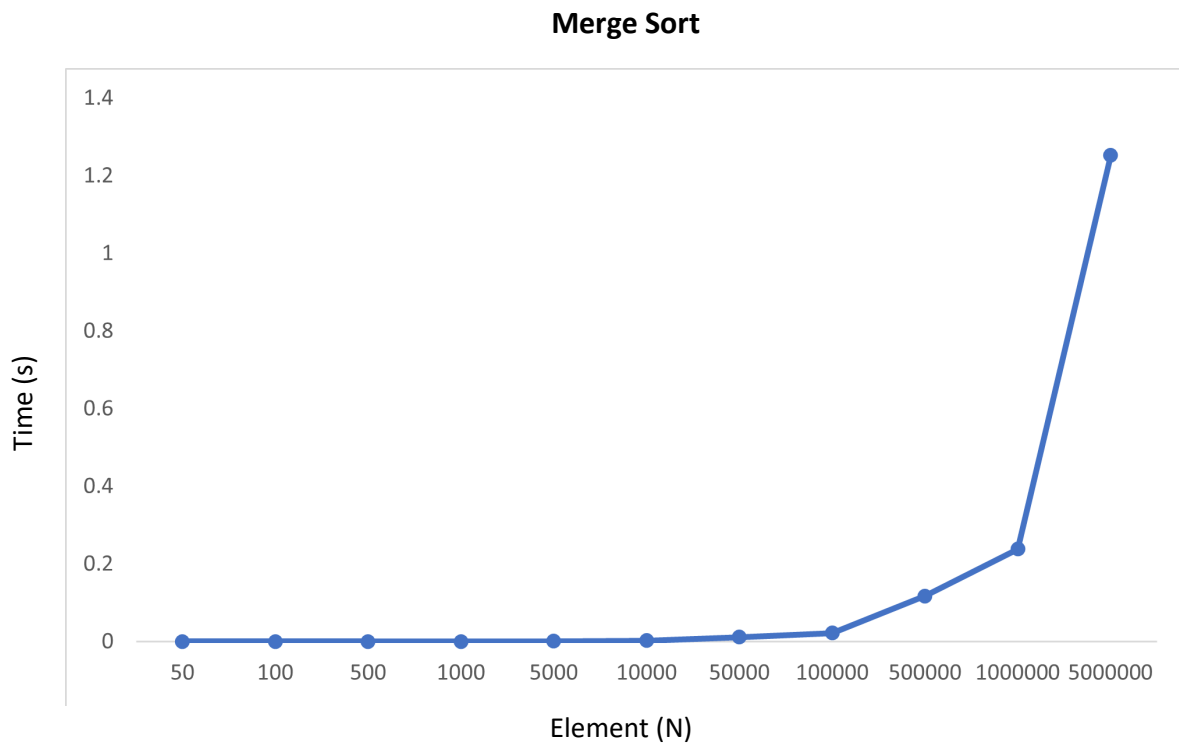
Number	Time (seconds)
500000	0.131
500000	0.116
500000	0.112
500000	0.115
500000	0.112
Average	0.1172

N = 1M

Number	Time (seconds)
1M	0.232
1M	0.259
1M	0.233
1M	0.232
1M	0.236
Average	0.2384

N = 5M

Number	Time (seconds)
5M	1.254
5M	1.25
5M	1.242
5M	1.257
5M	1.26
Average	1.2526



Result: We know that the efficiency of Merge sort is $O(n \log^2 n)$ which looks similar to linear but it's super linear. This is the fastest algorithm of all 4 algorithms, can sort 5 million inputs in 1.2 seconds.

Quick sort

For n = 8

```
Enter the number of elements: 8
Element of the element before sorting
176271 794865 345316 827146 610683 291924 854228 11848

Element of the element after sorting
11848 176271 291924 345316 610683 794865 827146 854228
```

For n = 15

```
Enter the number of elements: 15
Element of the element before sorting
8874 748162 314803 530231 153292 527150 343357 936040 322001
922826 616946 529249 165090 552549 868385

Element of the element after sorting
8874 153292 165090 314803 322001 343357 527150 529249 530231
552549 616946 748162 868385 922826 936040
```

Comparison of quick sort algorithm

n = 50

```
Enter the number of elements: 50
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 50
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 50
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 50
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 50
Running time of quick sort: 0.000000 seconds
```

Numbers	Times (Seconds)
1	0
2	0
3	0
4	0
5	0

Average time of n = 50 is 0 second.

n = 100

```
Enter the number of elements: 100
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 100
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 100
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 100
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 100
Running time of quick sort: 0.000000 seconds
```

Numbers	Times (Seconds)
1	0
2	0
3	0
4	0
5	0

Average time of n = 100 is 0 second.

n = 500

```
Enter the number of elements: 500
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 500
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 500
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 500
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 500
Running time of quick sort: 0.000000 seconds
```

Numbers	Times (Seconds)
1	0
2	0
3	0
4	0
5	0

Average time of n = 500 is 0 second.

n = 1,000

```
Enter the number of elements: 1000
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 1000
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 1000
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 1000
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 1000
Running time of quick sort: 0.000000 seconds
```

Numbers	Times (Seconds)
1	0
2	0
3	0
4	0
5	0

Average time of n = 1,000 is 0 second.

n = 5,000

```
Enter the number of elements: 5000
Running time of quick sort: 0.001000 seconds
Enter the number of elements: 5000
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 5000
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 5000
Running time of quick sort: 0.000000 seconds
Enter the number of elements: 5000
Running time of quick sort: 0.001000 seconds
```

Numbers	Times (Seconds)
1	0.001
2	0
3	0
4	0
5	0.001

Average time for n = 5,000 is 0.0004 second.

n = 10,000

```
Enter the number of elements: 10000
Running time of quick sort: 0.001000 seconds
Enter the number of elements: 10000
Running time of quick sort: 0.002000 seconds
Enter the number of elements: 10000
Running time of quick sort: 0.002000 seconds
Enter the number of elements: 10000
Running time of quick sort: 0.001000 seconds
Enter the number of elements: 10000
Running time of quick sort: 0.002000 seconds
```

Numbers	Times (Seconds)
1	0.001
2	0.002
3	0.002
4	0.001
5	0.002

Average time of n = 10,000 is 0.0016 second.

n = 50,000

```
Enter the number of elements: 50000
Running time of quick sort: 0.008000 seconds
Enter the number of elements: 50000
Running time of quick sort: 0.008000 seconds
Enter the number of elements: 50000
Running time of quick sort: 0.008000 seconds
Enter the number of elements: 50000
Running time of quick sort: 0.008000 seconds
Enter the number of elements: 50000
Running time of quick sort: 0.009000 seconds
```

Numbers	Times (Seconds)
1	0.008
2	0.008
3	0.008
4	0.008
5	0.009

Average time of n = 50,000 is 0.0082 second.

n = 100,000

```
Enter the number of elements: 100000
Running time of quick sort: 0.015000 seconds
Enter the number of elements: 100000
Running time of quick sort: 0.016000 seconds
Enter the number of elements: 100000
Running time of quick sort: 0.017000 seconds
Enter the number of elements: 100000
Running time of quick sort: 0.016000 seconds
Enter the number of elements: 100000
Running time of quick sort: 0.017000 seconds
```

Numbers	Times (Seconds)
1	0.015
2	0.016
3	0.017
4	0.016
5	0.017

Average time of n = 100,000 is 0.0162 second.

n = 500,000

```
Enter the number of elements: 500000
Running time of quick sort: 0.089000 seconds
Enter the number of elements: 500000
Running time of quick sort: 0.090000 seconds
Enter the number of elements: 500000
Running time of quick sort: 0.090000 seconds
Enter the number of elements: 500000
Running time of quick sort: 0.090000 seconds
Enter the number of elements: 500000
Running time of quick sort: 0.091000 seconds
```

Numbers	Times (Seconds)
1	0.089
2	0.09
3	0.09
4	0.09
5	0.091

Average time of n = 500,000 is 0.09 second.

n = 1,000,000

```
Enter the number of elements: 1000000
Running time of quick sort: 0.301000 seconds
Enter the number of elements: 1000000
Running time of quick sort: 0.262000 seconds
Enter the number of elements: 1000000
Running time of quick sort: 0.228000 seconds
Enter the number of elements: 1000000
Running time of quick sort: 0.222000 seconds
Enter the number of elements: 1000000
Running time of quick sort: 0.211000 seconds
```

Numbers	Times (Seconds)
1	0.301
2	0.262
3	0.228
4	0.222
5	0.211

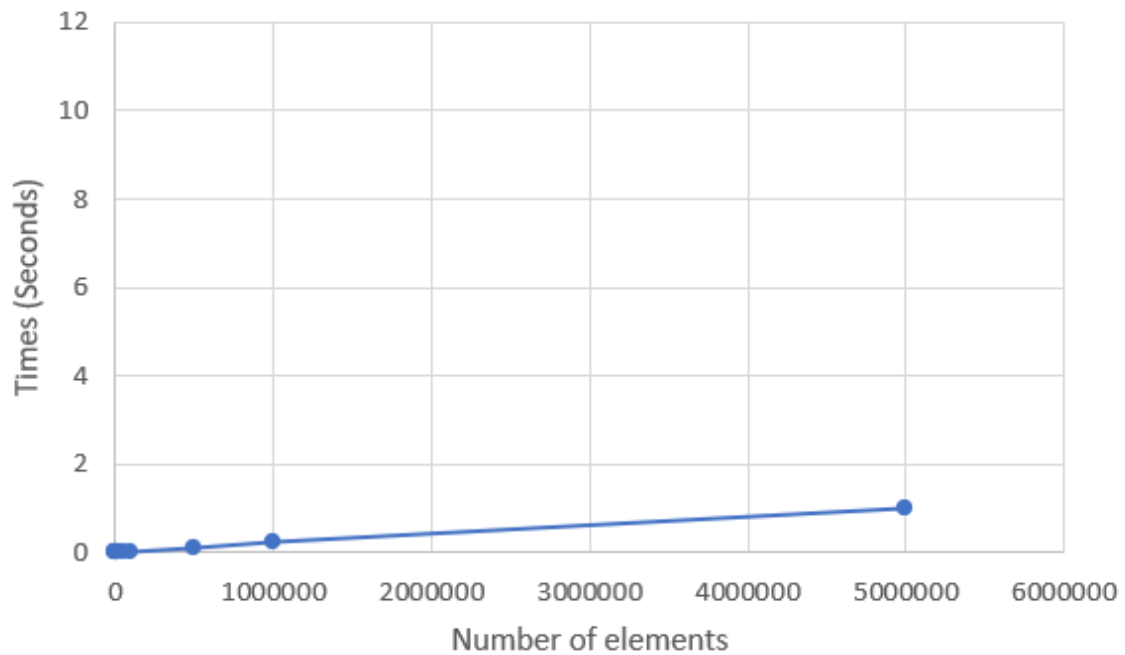
Average time of n = 1,000,000 is 0.2448 second.

N = 5,000,000

```
Enter the number of elements: 5000000
Running time of quick sort: 1.009000 seconds
Enter the number of elements: 5000000
Running time of quick sort: 0.988000 seconds
Enter the number of elements: 5000000
Running time of quick sort: 0.994000 seconds
Enter the number of elements: 5000000
Running time of quick sort: 1.006000 seconds
Enter the number of elements: 5000000
Running time of quick sort: 1.015000 seconds
```

Numbers	Times (Seconds)
1	1.009
2	0.988
3	0.994
4	1.006
5	1.015

Average time of n = 5,000,000 is 1.0024 second.



Result: As we have known from a lecture that an average efficiency of a quick sort is $\Theta(n \log(n))$ which is a superlinear function, and the result of the graph conforms to the lecture. The more number of elements are received, the growth order of running times occurs slightly.