Xiaoying Li
CS 325 -Winter 2020
Homework #1

- **Problem 1**

Order the functions by growth rate:
37 < $\sqrt{N}$ < N/2 < N < NloglogN < NlogN < $Nlog(N^2)$ < $Nlog^2N$ < $N^{1.5}$ < N^2 < N^2logN < N^3 < $2^{N/2}$ < 2^N

Functions that grow at the same rate:
1. N/2 and N grow at the same rate.

Because the big-O notation of N/2 is O(N). And the big-O notation of N is also O(N). Therefore, N/2 and N have the same growth rate.

2.. NlogN and $Nlog(N^2)$ grow at the same rate.

Because $Nlog(N^2) = N(2logN) = 2NlogN$, so the big-O notation of $Nlog(N^2)$ is O(NlogN). And the big-O notation of NlogN is also O(NlogN). Therefore, NlogN and $Nlog(N^2)$ have the same growth rate O(NlogN).

- **Problem 2**

$N^{1+\varepsilon/\sqrt{logN}}$ grows faster than NlogN.

To show $N^{1+\varepsilon/\sqrt{logN}}$ grows faster than NlogN, we need to show $\lim_{N\to\infty} \frac{NlogN}{N^{1+\frac{\varepsilon}{\sqrt{logN}}}} = 0$. Then,

$$\lim_{N\to\infty} \frac{NlogN}{N^{1+\frac{\varepsilon}{\sqrt{logN}}}} = \lim_{N\to\infty} \frac{NlogN}{N \cdot N^{\frac{\varepsilon}{\sqrt{logN}}}}$$

$$= \lim_{N\to\infty} \frac{logN}{N^{\frac{\varepsilon}{\sqrt{logN}}}}$$

$$= \lim_{N\to\infty} \frac{\alpha^{\log(logN)}}{\alpha^{\log(N^{\varepsilon/\sqrt{logN}})}}, \alpha\ is\ the\ base\ number\ of\ log, and\ \alpha > 1$$

$$= \lim_{N\to\infty} \frac{\alpha^{\log(logN)}}{\alpha^{\frac{\varepsilon}{\sqrt{logN}}logN}}$$

$$= \lim_{N\to\infty} \frac{\alpha^{\log(logN)}}{\alpha^{\frac{\varepsilon}{\sqrt{logN}}\cdot(\sqrt{logN})^2}}$$

$$= \lim_{N\to\infty} \frac{\alpha^{\log(logN)}}{\alpha^{\varepsilon\sqrt{logN}}}$$

Let $x = \sqrt{logN}$, and $\lim_{N\to\infty} \sqrt{logN} = \infty$, then,

$$\lim_{N\to\infty} \frac{NlogN}{N^{1+\frac{\varepsilon}{\sqrt{logN}}}} = \lim_{N\to\infty} \frac{\alpha^{\log(logN)}}{\alpha^{\varepsilon\sqrt{logN}}}$$

$$= \lim_{x\to\infty} \frac{\alpha^{\log x^2}}{\alpha^{\varepsilon x}}$$

$$= \lim_{x\to\infty} \frac{\alpha^{2\log x}}{\alpha^{\varepsilon x}}$$

And $2logx = O(logx)$, $\varepsilon x = O(x)$.

Because log x grows slower than x and $\varepsilon > 0$, so $\alpha^{2\log x}$ grows slower than $\alpha^{\varepsilon x}$, so $\lim_{x\to\infty} \frac{\alpha^{2\log x}}{\alpha^{\varepsilon x}} = 0$.

Thus $\lim_{N\to\infty} \frac{NlogN}{N^{1+\frac{\varepsilon}{\sqrt{logN}}}} = 0$. As was to be shown, $N^{1+\varepsilon/\sqrt{logN}}$ grows faster than NlogN.

- **Problem 3**

(1) The loop runs n times.

So, the Big-Oh is O(n).

(2) The outer loop runs n times; the inner loop runs n times for every fixed i.

Thus the total running time is $n \cdot n = n^2$. So, the Big-Oh is $O(n^2)$.

(3) The outer loop runs n times; the inner loop runs n^2 times for every fixed i.

Thus the total running time is $n \cdot n^2 = n^3$. So, the Big-Oh is $O(n^3)$.

(4) The outer loop runs n times totally;

the inner loop runs $1 + 2 + 3 + \cdots + n = \frac{n(1+n)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$ times totally.

Thus the total running time is $n + \frac{1}{2}n^2 + \frac{1}{2}n = \frac{1}{2}n^2 + \frac{3}{2}n$, so the Big-Oh is O($n^2$).

(5) The total running time is

$$\sum_{i=0}^{n-1} \sum_{j=0}^{i^2-1} \sum_{k=0}^{j-1} 1$$

$$= \sum_{i=0}^{n-1} \sum_{j=0}^{i^2-1} j$$

$$= \sum_{i=0}^{n-1} 0 + 1 + 2 + 3 + \cdots + i^2 - 1$$

$$= \sum_{i=0}^{n-1} \frac{i^2(i^2-1)}{2}$$

$$= \frac{1}{2}[(0 + 1^4 + 2^4 + 3^4 + \cdots + (n-1)^4) + (0 + 1^2 + 2^2 + 3^2 + \cdots + (n-1)^2)]$$

$$= \frac{1}{2}[\left(\frac{(n-1)^3}{3} + \frac{(n-1)^2}{2} + \frac{n-1}{6}\right) + \left(\frac{(n-1)^5}{5} + \frac{(n-1)^4}{2} + \frac{(n-1)^3}{3} - \frac{n-1}{30}\right)]$$

Ignore all low-order terms and leading constants, so the Big-Oh is $O(n^5)$.

(6) The innermost loop runs only when j=i, so it runs i times for every fixed i.

The total running time is

$$\sum_{i=0}^{n-1} \sum_{j=0}^{i^2-1} i = \sum_{i=0}^{n-1} i^3 = \frac{(n-1)^2 n^2}{4} = \frac{n^4 - 2n^3 + n^2}{4}$$

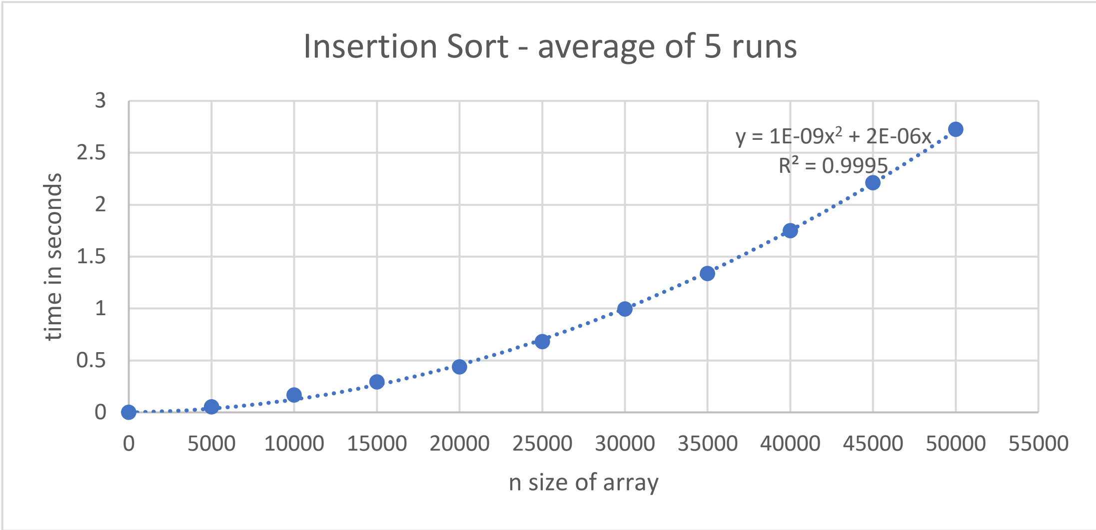
Ignore all low-order terms and leading constants, so the Big-Oh is $O(n^4)$.

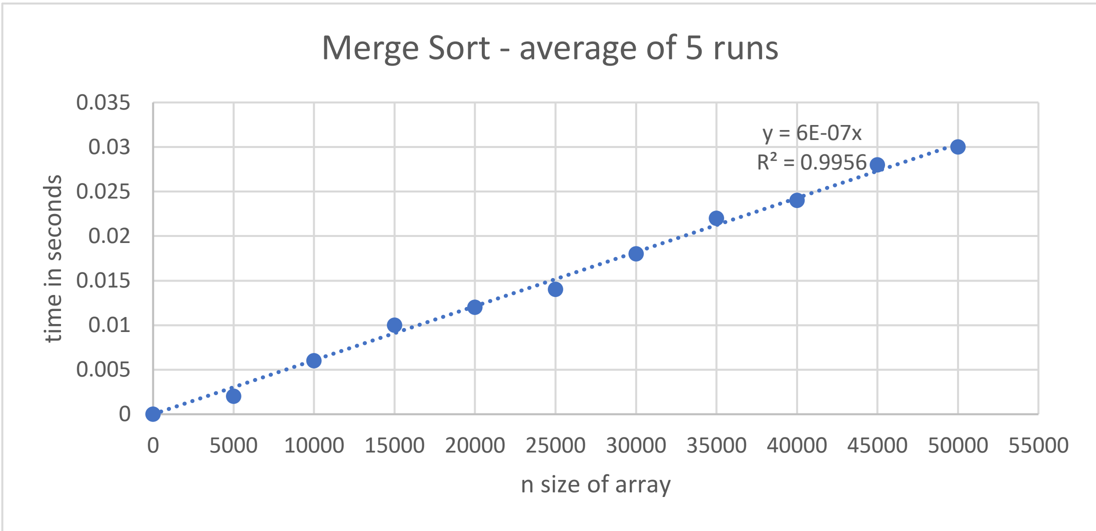
- **Problem 5**

(b)

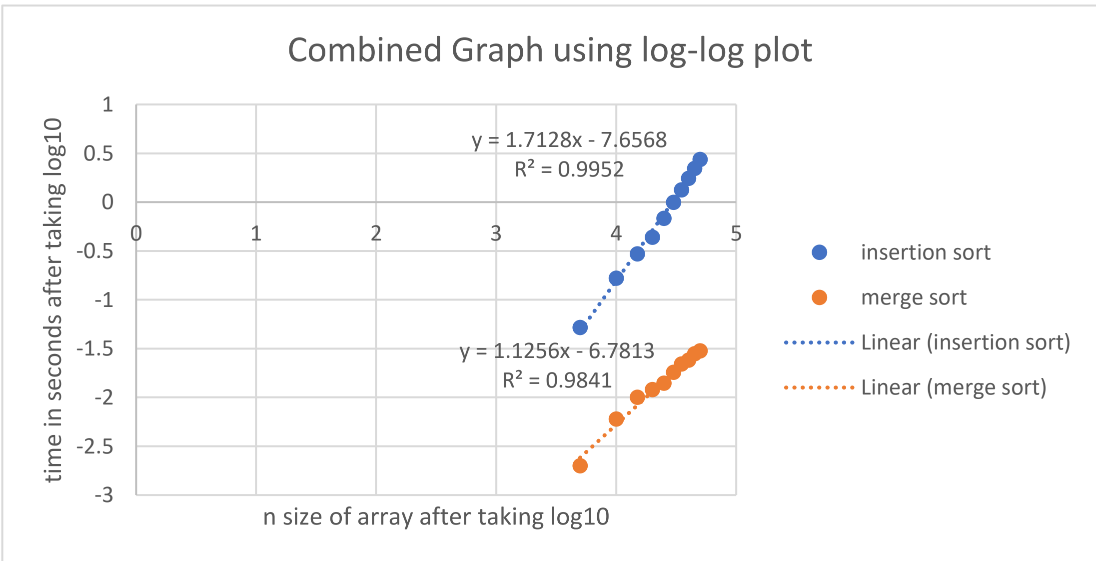
Size	Time(s)	
	Insertion Sort	Merge Sort
0	0	0
5000	0.052	0.002
10000	0.166	0.006
15000	0.294	0.01
20000	0.438	0.012
25000	0.68	0.014
30000	0.994	0.018
35000	1.336	0.022
40000	1.75	0.024
45000	2.21	0.028
50000	2.724	0.03

(c)


Insertion Sort - average of 5 runs
y = 1E-09x² + 2E-06x
R² = 0.9995
time in seconds
3
2.5
2
1.5
1
0.5
0
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000
55000
n size of array


Merge Sort - average of 5 runs
y = 6E-07x
R² = 0.9956
time in seconds
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000
55000
n size of array

(d)


Combined Graph using log-log plot
y = 1.7128x - 7.6568
R² = 0.9952
y = 1.1256x - 6.7813
R² = 0.9841
time in seconds after taking log10
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
1
2
3
4
5
n size of array after taking log10
insertion sort
merge sort
Linear (insertion sort)
Linear (merge sort)

(e)

The experimental running times of insertion sort and merge sort can be described by the equations we concluded in part (c). The theoretical running times of the two algorithms can be described by their Big O.

For insertion sort, the experimental running time equation is $y = 10^{-9}x^2 + 2 \cdot 10^{-6}x = O(x^2)$. And the theoretical Big O of insertion sort is also $O(n^2)$. So, our experimental result is very close to the theoretical running times, almost same.

For merge sort, the experimental running time equation is $y = 6 \cdot 10^{-7}x = O(x)$. And the theoretical Big O of merge sort is $O(nlogn)$. It seems that our experimental result is different from the theoretical running times. But in my opinion, the curve of nlogn is very similar to a linear graph and the excel cannot generate a trendline of nlogn, so in fact our experimental result is similar to the theoretical running times.