CS350 HW2 - rewrite
Section 004
Jiacheng Zhao

What sources did you use for your algorithms?
I use my notebook create code and online source.
What programming language did you choose and why?
I used C++ in Linux. Because I familiar with it.
The three algorithms: selection sort, merge sort and bubble sort.
Did you run into any difficulties with the implementation?
I use the Linux to implement.
How are you timing the algorithms?
I used time command(time ./a.out) to analyze those algorithms for tracking the time.
What hardware are you timing on?
My hardware is MacBook Pro 2018.
I use array as the data structure and I enter the length at the beginning. The range of number is 0-100. I have two options: random and sorted. For the random options, I use "rand()" function to make elements are random numbers. For the sorted option, I use a loop to set the elements in order so that it will not affect the efficiency of the sorting process.

Selection sort:

Selection random	Input Size	10	50	100	200	1000	5000	10000	100000
user		0.001 s	0.003 s	0.003 s	0.004 s	0.007 s	0.004 s	0.102 s	9.132 s
system		0.003 s	0.003 s	0.003 s	0.004 s	0.003 s	0.004 s	0.004 s	0.047 s

Selection sorted	Input Size	10	50	100	200	1000	5000	10000	100000
user		0.001 s	0.003 s	0.003 s	0.003 s	0.008 s	0.039 s	0.109 s	9.112 s
system		0.003 s	0.003 s	0.003 s	0.004 s	0.004 s	0.004 s	0.006 s	0.012 s

There is no big different between random and sorted array. It just traversal array n times for the length n. Though it is already sorted, we still have to traversal the array again and again. The different is sorted need not swap elements. It matches the theoretical analysis.

Merge Sort:

Merge Sort random	Input Size	10	50	100	200	1000	5000	10000	100000
user		0.002 s	0.002 s	0.002 s	0.003 s	0.006 s	0.012 s	0.021 s	0.101 s
system		0.003 s	0.003 s	0.003 s	0.003 s	0.004 s	0.004 s	0.006 s	0.009 s

Merge Sort sorted	Input Size	10	50	100	200	1000	5000	10000	100000
user		0.002 s	0.002 s	0.003 s	0.003 s	0.004 s	0.011 s	0.019 s	0.098 s
system		0.003 s	0.003 s	0.003 s	0.003 s	0.003 s	0.005 s	0.005 s	0.010 s

It is more quick than Selection Sort. Because it is recursive algorithm. It is clearly different when input size become 5000 and above. It is no big different between sorted and random array. Because they both need “divide and conquer”. I will choose Merge sort when sorting the bigger lists.

Bubble Sort:

Bubble Sort random	Input Size	10	50	100	200	1000	5000	10000	100000
user		0.002 s	0.002 s	0.003 s	0.002 s	0.011 s	0.080 s	0.293 s	30.07 s
system		0.004 s	0.005 s	0.003 s	0.004 s	0.004 s	0.005 s	0.005 s	0.058 s

Bubble Sort sorted	Input Size	10	50	100	200	1000	5000	10000	100000
user		0.002 s	0.003 s	0.003 s	0.003 s	0.009 s	0.039 s	0.119 s	12.35 s
system		0.003 s	0.003 s	0.003 s	0.003 s	0.004 s	0.005 s	0.004 s	0.039 s

Sorted array is more quick than random array by using bubble sort. It is not good algorithm by sorting big lists. The code of bubble sort is same with selection sort but slower.

Merge sort is the best after comparing these three. I would like to choose the merge sort in real life. Because selection sort and bubble sort are much slower when dealing with big data lists.