

Lab 6: Sort Algorithms

Sorting Algorithm Comparisons on Sorted and not Sorted List

This Lab is based on group of three. Check the list bellow to find your group members. The first person will be group leader. Leader must divide the job evenly between group members.

This lab will compare three different sort algorithms. You are allow to use the given code in the class or code book offers or write your own code.

(Note: You are testing these sort algorithms on integer numbers)

Comparison Sorting Visualizations link can help you to have some perspective on sort comparison:
<https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>

Your goal for this lab is to implement simple versions of Insertion Sort - `insert_sort(alist)`, Selection Sort - `select_sort(alist)`, and Merge Sort - `merge_sort(alist)` that will sort an array of integers and count the number of comparisons. Include your test cases as lists in the same file as your functions. Each function takes as input a list of integers, sorts the list counting the comparisons at the same time, and returns the number of comparisons. After the function completes the “alist” should be sorted. Submit in a file called `comparison_sort.py` to PolyLearn. **Do not do any improvement in the sort algorithms which can cause reducing the number of comparison. Use the original algorithms. Make sure you are using the same list for all sort algorithms.**

The runtime complexity is $O(n^2)$ for selection and insertion sort; and $O(n \log(n))$ for merge sort, Why? You will submit your answers in a pdf file and submit with the results. Write out the summation that represents the number of comparisons. Note that you will need to run test cases of different sizes to show that Insertion and Selection sort are $O(n^2)$ and Merge sort is $O(n \log(n))$. To do this, plot the number of comparisons (y-axis) vs. the size of the list (x-axis). Since the plot is shaped like a parabola for selection and insertion sort this indicates that it is quadratic. In the pdf file, for each Insertion, Selection, and Merge sort submit a table showing list size and number of comparisons from your test cases and a plot of that data.

For generating the list you can create a list of random numbers.

```
import random
alist = []
#10 random numbers between 10 and 70
for i in range(10):
    #integer random numbers between 10 and 70
    n=random.randint(10,70)
    alist.append(n)
print (alist)
```

[10, 31, 39, 36, 49, 40, 35, 69, 66, 51]

Submission:

- 1) comparison_sort.py
- 2) Comparison_analysis.pdf

Only Leader of group need to submit the files. Write group members name in header of both files.

The pdf file will include:

- 1) Describe responsibility of each group member
- 2) Your analysis of comparisons for sorted and unsorted list
- 3) The results of sorted and unsorted comparison
- 4) Plot the number of comparisons (y-axis) vs. the size of the list (x-axis). Since the plot is shaped like a parabola for selection and insertion sort this indicates that it is quadratic.

Size	Unsorted List		
N	Insertion Sort	Selection Sort	Merge Sort
100			

Size	Sorted List		
N	Insertion Sort	Selection Sort	Merge Sort
100			