

## EI331 COURSE PROJECT

---

By: Wang Haoxuan

Instructor: Wu ChenTao

November 4, 2018

## I. PROJECT 3

### A. Multithreaded Sorting Application

In this part, we are required to sort a list of integers using multiple threads. Three threads are needed: one for each part of the list, and the last one to merge the two sorted lists together. Two *runner* functions are created. The whole list is defined as a global variable. In the code, I created a simple list that has integers from 1 to 5, and resulted in sorting it out. Result is shown below:

```
jerrywang@ubuntu:~/Documents/EI338/ch4$ ./sort
1 2 3 4 5
```

FIG. 1: Multithreaded Sorting

### B. Fork-Join Sorting Application

This section is written in Java since we already have a Java template of adding a list of numbers together. The Java template provided the recursive version for summing up a list of numbers. In a similar manner, we can derive the Quicksort and Mergesort algorithm from the template. When the total number of the elements in a sublist is smaller than a range point, we use bubble sort to get the result. When the number is larger, Quicksort and Mergesort algorithm is used. The two algorithms are shown in the two Java files. We input a random array of numbers and sort them. Result is shown below:

```
jerrywang@ubuntu:~/Documents/EI338/ch4$ java QuickSort
0
0
0
1
2
2
3
4
6
9
jerrywang@ubuntu:~/Documents/EI338/ch4$ java MergeSort
0
0
4
5
5
5
7
8
8
9
```

FIG. 2: Fork-Join Sorting