DATA-Lab6

Question 6.

Both mergeSort() and heapSort() run O(nlog(n)). To compare running time of those functions, **Wilcoxon two-sided tests** were performed:

1. Tests at different size of input
2. Tests at a specific size of input

H0: mergeSort() and heapSort() do not express significant difference in running time (The median difference in runtime of 2 implementations (T_merge-T_heap) is 0)

⇒ As you can see from the result below, most of the p-value are all greater than 0.05, which means that H0 of each test is not rejected. Then, there is no significant difference in running time of mergeSort() and heapSort(). This agrees with the analysis of complexity time.

Old result

```
Wilcoxon two-sided at different size of input
Attempt no.1
        Wilcoxon p-value: 0.9939
Attempt no.2
        Wilcoxon p-value: 0.8595
Attempt no.3
        Wilcoxon p-value: 0.9750
Attempt no.4
        Wilcoxon p-value: 0.4422
Attempt no.5
        Wilcoxon p-value: 0.3158
```

```
Wilcoxon two-sided at input size of 15
Attempt no.1
        Wilcoxon p-value: 0.0970
Attempt no.2
        Wilcoxon p-value: 0.8179
Attempt no.3
        Wilcoxon p-value: 0.4608
Attempt no.4
        Wilcoxon p-value: 0.0080
Attempt no.5
        Wilcoxon p-value: 0.7438
Wilcoxon two-sided at input size of 50
Attempt no.1
        Wilcoxon p-value: 0.1934
Attempt no.2
        Wilcoxon p-value: 0.9659
Attempt no.3
        Wilcoxon p-value: 0.2544
Attempt no.4
        Wilcoxon p-value: 0.6894
Attempt no.5
        Wilcoxon p-value: 0.3260
```

```
Wilcoxon two-sided at input size of 100
Attempt no.1
        Wilcoxon p-value: 0.6465
Attempt no.2
        Wilcoxon p-value: 0.1327
Attempt no.3
        Wilcoxon p-value: 0.4309
Attempt no.4
        Wilcoxon p-value: 0.5450
Attempt no.5
        Wilcoxon p-value: 0.1549
```

```
Wilcoxon two-sided at input size of 1000
Attempt no.1
        Wilcoxon p-value: 0.1784
Attempt no.2
        Wilcoxon p-value: 0.6302
Attempt no.3
        Wilcoxon p-value: 0.0372
Attempt no.4
        Wilcoxon p-value: 0.3508
Attempt no.5
        Wilcoxon p-value: 0.3592
```

Question 6. Modified

Method:

1. The running time of each function on an array of n integers is collected and plotted.
2. Wilcoxon one sided or two sided tests are performed suitable with the result in the diagram.

⇒ ***Result:***

With n=15, there is no significant difference in the running time of mergeSort() and heapSort()

With n=50 or 100 or 1000, T_merge is significantly smaller than T_heap

With n=[15, 50, 100, 1000], mergeSort() is significantly faster than heapSort()

```
Wilcoxon one-sided at different size of input (T_merge<T_heap)
Attempt no.1
  pvalue: 1.0
Attempt no.2
  pvalue: 1.0
Attempt no.3
  pvalue: 1.0
Attempt no.4
  pvalue: 1.0
Attempt no.5
  pvalue: 1.0
```

```
Wilcoxon two-sided at input size of 15 (T_merge=T_heap)
Attempt no.1
  pvalue: 0.5298
Attempt no.2
  pvalue: 0.5191
Attempt no.3
  pvalue: 0.8587
Attempt no.4
  pvalue: 0.4366
Attempt no.5
  pvalue: 0.8527
```

```
Wilcoxon one-sided at input size of 50 (T_merge<T_heap)
Attempt no.1
  pvalue: 0.9346
Attempt no.2
  pvalue: 0.8674
Attempt no.3
  pvalue: 0.9726
Attempt no.4
  pvalue: 0.9774
Attempt no.5
  pvalue: 0.9942
```

```
Wilcoxon one-sided at input size of 100 (T_merge<T_heap)
Attempt no.1
  pvalue: 0.9947
Attempt no.2
  pvalue: 0.9739
```

```
Attempt no.3
  pvalue: 0.9982
Attempt no.4
  pvalue: 1.0
Attempt no.5
  pvalue: 0.9798
```

```
Wilcoxon one-sided at input size of 1000 (T_merge<T_heap)
Attempt no.1
  pvalue: 1.0
Attempt no.2
  pvalue: 1.0
Attempt no.3
  pvalue: 1.0
Attempt no.4
  pvalue: 1.0
Attempt no.5
  pvalue: 1.0
```