DATA-Lab3

Question 5:

The Big-O notation of each functions are listed in the table 1. For more details, please check Python scripts.

Table 1

Name	Array	Singly linked list
emptyStack()	O(1)	O(1)
push()	O(1)	O(1)
pop()	O(1)	O(1)
peek()	O(1)	O(1)
enqueue()	O(1)	O(1)
dequeue()	O(1)	O(n)

The results of all attempts below showed that no p value is smaller than 0.05, which means that H0 of those tests could not be rejected.

Then, I came to the following conclusion:

- For ***Stack***, there is no significant difference in runtime (efficiency) when implementing four functions of stack (***emptyStack(), push(), pop(), peek()***) with array and singly linked list.
- For ***Queue***, there is no significant difference in runtime (efficiency) when implementing ***enqueue()*** functions of queue with array and singly linked list. However, implementing ***dequeue()*** function with array is more efficient than implementing queue with singly link list.

Result from ex5_comparative_test.py

```
Attempt no.1:
        p_value_emptyStack: 0.47555
        p_value_push: 0.02506
        p_value_pop: 0.20063
        p_value_peek: 0.40354
        p_value_enqueue: 0.92598
        p_value_dequeue: 0.00098
Attempt no.2:
        p_value_emptyStack: 0.75856
        p_value_push: 0.30918
```

```
        p_value_pop: 0.427
        p_value_peek: 0.07816
        p_value_enqueue: 0.17664
        p_value_dequeue: 0.05078
Attempt no.3:
        p_value_emptyStack: 0.30105
        p_value_push: 0.40359
        p_value_pop: 0.96069
        p_value_peek: 0.58678
        p_value_enqueue: 0.40097
        p_value_dequeue: 0.55104
Attempt no.4:
        p_value_emptyStack: 0.37354
        p_value_push: 0.03807
        p_value_pop: 0.4055
        p_value_peek: 0.64888
        p_value_enqueue: 0.88081
        p_value_dequeue: 0.44431
Attempt no.5:
        p_value_emptyStack: 0.17888
        p_value_push: 0.16095
        p_value_pop: 0.58035
        p_value_peek: 0.61382
        p_value_enqueue: 0.09363
        p_value_dequeue: 0.59634
Attempt no.6:
        p_value_emptyStack: 0.20597
        p_value_push: 0.76528
        p_value_pop: 0.61748
        p_value_peek: 0.11298
        p_value_enqueue: 0.75919
        p_value_dequeue: 0.73237
Attempt no.7:
        p_value_emptyStack: 0.73367
        p_value_push: 0.49176
        p_value_pop: 0.14054
        p_value_peek: 0.45326
        p_value_enqueue: 0.65076
        p_value_dequeue: 0.12446
Attempt no.8:
        p_value_emptyStack: 0.90827
        p_value_push: 0.96588
        p_value_pop: 0.91319
        p_value_peek: 0.73757
        p_value_enqueue: 0.45836
        p_value_dequeue: 0.23154
Attempt no.9:
        p_value_emptyStack: 0.5124
        p_value_push: 0.14513
        p_value_pop: 0.06784
        p_value_peek: 0.82277
        p_value_enqueue: 0.56571
        p_value_dequeue: 0.74344
Attempt no.10:
        p_value_emptyStack: 0.07509
        p_value_push: 0.83952
        p_value_pop: 0.83156
        p_value_peek: 0.28278
```

```
        p_value_enqueue: 0.62608
        p_value_dequeue: 0.30066
```