|  |  |
| --- | --- |
| **CSE 331** | **Spring 2018** |

Project 1

100 points

Due September 28th 8:00pm

This is not a team work, do not copy somebody else’s work.

**Reference:** Author of this project is Cyndy Ishida

**Assignment Overview**

Quick Sort is often the preferred sorting algorithm for large scale applications. This assignment will be an implementation of this famous algorithm but, instead of a contiguous array, it will be a done with a Queue.

**Assignment Deliverables**

Be sure to use the specified file name(s) and to submit your files for grading **via D2L Dropbox** before the project deadline.

* Queue.py
* QuickSort.py

**Assignment Specifications**

Your task will be to complete the methods listed below.

in Queue.py

* **\_\_len\_\_(self)**
* return (type int) the number of elements present in queue
* O(1) space
* O(1) time
* **is\_empty(self)**
* return (type Boolean) true if queue is empty
* O(1) space
* O(1) time
* **dequeue(self)**
* return (type Node.val) element that was just removed from queue \* guaranteed built-in scalar type (i.e. ints, floats, chars)
* O(1) space
* O(1) time
* **enqueue(self, element)**
* return (type None)
* add element to end of queue
* O(1) space
* O(1) time
* **\_\_getitem\_\_(self, index)**
* return (type node.val) current node’s val at that index
* O(N) time
* O(1) space
* This is a “under the hood” method that gets invoked called the index operator

i.e. print(queue[4])

in QuickSort.py

* **insertion\_sort(queue)**
* return (type None)
* O(N^2) time, same as lecture, best case should be O(N)
* O(1) space
* Sorting should modify the given container
* **pick\_pivot(queue)**
* return (type Node.val ) \* guaranteed built-in scalar type (i.e. ints, floats, chars) median element between first element, the last element, and the middle index
* O(N) time
* O(1) space
* this function should never be ran with less than 3 elements in queue
* **quick\_sort(queue)**
* return (type None)
* O(n^2\*log(n)) time
* O(n\*log(n)) space
* Sorting should modify given container

**Assignment Notes**

Points will be deducted if your solution has any warnings of type:

* Any use of a container type that isn’t a LinkedQueue Type will result in a 50% decrease to final grade for assignment
* You are not allowed to use any outside module like ‘import sys’ or to modify the set recursion limit
* Mutators for Queue method will not be called on an empty list. i.e. error checking for empty queue is not needed.
* Elements for each node will only be built-in scalar types (int, float, chars)
* You are guaranteed that at least 5 elements will exist in queue to be sorted
* There is a sample testcase provided, that doesn’t count for points ‘testcase00.txt’
* You can add additional functions, however you aren’t allowed to modify an function signatures
* Docstrings/Pre-Post Conditions are required on all function/method signatures
* It is the assumption you are using the python3.6 interpreter

Testing your work

Run your project on Pycharm see sample run below

