Supplemental Instruction Fall 2022
CSCI 141 Practice Exam 2

1. A stack works like a line of cars at the gas station. [TRUE / FALSE]
2. You can use a Python list to implement a stack by making the end of the list the top of your stack [TRUE / FALSE]
3. A queue is a LIFO structure [TRUE / FALSE]
4. The *top* function mutates a stack. [TRUE / FALSE]
5. The undo button operates like a stack. [TRUE / FALSE]
6. The keys in a dictionary are mutable. [TRUE / FALSE]
7. Follow the stack!

   stk = mkEmptyStack()

   push(stk, “A”)

   push(stk, “B”)

   push(stk, “C”)

   pop(stk)

   push(stk, “D”)

   push(stk, “E”)

   pop(stk)

   pop(stk)

   push(stk, “F”)

8. Follow the queue!

```
q = mkEmptyQueue()
enqueue(q, 3)
enqueue(q, 2)
enqueue(q, 1)
print(front(q))
print(back(q))
print(dequeue(q))
dequeue(q)
dequeue(q)
print(emptyQueue(q))
enqueue(q, “cue”)
```

9. Which of the following Python collections are immutable?

- Set
- List
- Tuple

10. Match the following:

____ | < 1, 2, 3, 4, 5, 6 >	1) List
____ | { 1, 2, 3, 4, 5, 6 }	2) Tuple
____ | (1, 2, 3, 4, 5, 6)	3) Set
____ | 1 2 3 4 5 6	4) Doesn’t match any types
____ | [1, 2, 3, 4, 5, 6]	
____ | 1, 2, 3, 4, 5, 6	

11. More matching:

1) Insertion Sort
2) Selection Sort
3) Merge Sort
4) Quick Sort

____ | Divide the list in half, recursively sort each half, then figure out how to interweave the two sorted halves such that you have one sorted list at the end

____ | Using some value already in the list, split up the list into the values that are smaller than, equal to, and greater than the value. Repeat the process on the first and third parts recursively, and then concatenate all the pieces together.

____ | Repeatedly take the values of the unsorted end of the list and find the right spot for them in the sorted end. Repeat until the entire list is sorted.

____ |Start from the left of the unsorted list and then compare that value to the value left of it. If the current value is less than the one you're comparing it to, swap the values and keep swapping until your value is greater than the one to the left.

12. Fill in the Time Complexity Table:

	Best Case	Average Case	Worst Case
Merge Sort			
Quick Sort			
Insertion Sort			
Selection Sort			
Binary Search			
Linear Search			

13. Searching for a value in a sorted list takes _____ time.

14. Searching for a value in an unsorted list takes _____ time.

15. Accessing the last element in a Python list takes _____ time.

16. Accessing a value in a Python dictionary takes _____ time.

17. Create a Data structure named *Student* that has a name, GPA, ID number, and major.

    Create two unique students using your new data structure

18. How can you check to see if ‘instant_ramen’exists in a dictionary, and if it doesn't how would you add it? Hint: ‘instant_ramen’ is a key

```
cookbook = dict()
```

19. Use the following data structure to finish the function:

```
@dataclass(frozen=False)
class LinkedNode:
    value: Any
    next: Union[None, 'LinkedNode']

def insert_in_order(value, lnk):
```