# COMP-202 FALL 202 HOMEWORK 3

**I have completed this assignment individually, without support from anyone else. I hereby accept that only the below listed sources are approved to be used during this assignment:**

**(i) Course textbook,**

**(ii) All material that is made available to me by the professor (e.g., via Blackboard for this course, course website, email from professor / TA),**

**(iii) Notes taken by me during lectures.**

**I have not used, accessed or taken any unpermitted information from any other source. Hence, all effort belongs to me.**

**I signed as Sinan Cem Erdoğan (68912).**

```
char front()
    if(front.next!= null)
       data = front.next.data
       return data
       }
    return '\0'
  }
```

Time Complexity = O(1) - Space Complexity = O(1)

We do 1 2 basic operaitons. We allocate memory for a few temporary variable.

```
char dequeue()
    if(front.next == null)
      print("Queue is empty.")
      return '\0'

    data = front.next.data
    front = front.next.next
    size--
    if(this.front.next == null)
      back = null

    return data
```

Time Complexity = O(1) - Space Complexity = O(1)

We do a few basic operaitons. We allocate memory for a few temporary variable.

```
public boolean isEmpty()
    if(front.next == null)
      return true
    return false
```

Time Complexity = O(1) - Space Complexity = O(1)

Just returning the return value of the function.

```
void enqueue(data)
    Node node = new Node(data)
    if(back.next == null)
      back = node
      front = node
      size++
      return
    }
    back.next = node
    back = node
    size++
  }
```

Time Complexity = O(1) - Space Complexity = O(1)

We only change a few pointers and we have some auxiliary values.

```
int size()
    return size
```

Time Complexity = O(1) - Space Complexity = O(1)

Just returning the return value of the function.

```
void ordering(array)
    Queue queue = new Queue()
    myQueue.enqueue(charArray[0])
    for(i = 1 to array.length )
      for(j = queue.size() to 0)
          data = dequeue
          if(data == array[i]) {
            continue
        enqueue(data)
      enqueue(array[i])
    System.out.println(queue)
```

Time Complexity

We have 4x (3y (4) ) operations. X denostes length of the given array, y denoted size of the queue. (Since Time Complexity of dequeue and enqueue O(1) we can treat them like they are basic operaitons.)

**So, we have T(n) ≈ 48n^2  Time Complexity of this function is O(n^2).**

Space Complexity

For the given array, we allocate n * (2 bytes). (n denotes length of the given array)

For the queue, we allocate n* c (size of a node which is constant). (For every element in array we create a node)

For the auxiliary values, we allocate 4 + 4 + 2 = 10 bytes.

**S(ordering) ≈ n (2 + c) + 10  which is O(n).**