CS0455 – ALGORITHMS AND INFORMATION STRUCTURES

Assignment 4 – Queues/Deques

1. Suppose an array is to be used to implement a regular FIFO queue. The implementation will have two instance variables: an array **`queue[]`** to store items of type **`T`** and **`size`** which stores the number of items in the queue (**`null`** represents unused entries in the array). The front of the queue will always be **`queue[0]`** and the back of the queue is always **`queue[size - 1]`**. The array will double in size if necessary.

   By referring to **`queue[]`** and **`size`**, and any parameters and/or return data, describe what happens during each of the following methods: **[10]**

   `public void enqueue(T newEntry)`

   `public T dequeue()`

   `public T getFront()`

   `public boolean isEmpty()`

   `public void clear()`

2. Suppose an array is to be used to implement a deque (double-ended queue). The implementation will have two instance variables: an array **queue[ ]** to store items of type **T** and **size** which stores the number of items in the queue (**null** represents unused entries in the array). The front of the queue will always be **queue[0]** and the back of the queue is always **queue[size – 1]**. The array will double in size if necessary.

   A deque has the same five methods as a queue with three extra ones added. By referring to **queue[ ]** and **size**, and any parameters and/or return data, describe what happens during each of the following methods: **[6]**

   public void addToFront(T newEntry)

   public T removeBack()

   public T getBack()

3a. In the **queue** class (Question 1) which of the operations **enqueue**, **dequeue** and **getFront** take the longest time in this implementation? Explain your answer. **[2]**

b. In the **queue** class (Question 1) which of the operations **enqueue**, **dequeue** and **getFront** take the shortest time in this implementation? Explain your answer. **[2]**