Samuel Jackson – 2520998J

ADS2 - Assessed Exercise 2

How to run the code:  
All the necessary code is in the following java files: ArrayQueue.java, BinaryTreeQueue.java, BinaryTreeQueue.java, BinaryTreeConstantQueue.java.

Important to note: Unless stated otherwise log n refers to log\_2 n, not log\_10 n.

Part 1 :

1. Explanation of implementation of array-based heap

The heap data structure is typically visualised as a tree but implemented as an array in this case. I will follow standard terminology for a tree-based heap for ease of understanding.

There are two primary possible structures for the heap:

* Min-heap
* Max-heap

The min-heap is a data structure such that each child is less than the parent. The max-heap data structure is symmetric in that each child is greater than the parent.

My implementation, for the sake of min-priority queue, has taken the approach of min-heap.

I have taken this implementation because it enables more efficient operations.

Class Structure:

The class contains three fields:

* int[] q (integer array containing the values).
* int n (integer representing current size of array).

It is important to note that n is not the max number of elements in q but rather the tail of the queue. The constructor requires an integer argument, passing the size of array q and creating it on the heap. You cannot insert beyond the max size of the array.

Insert:

Insert takes the argument of an integer and then ends if there is an overflow.

If there is not an overflow, then the integer is assigned to the list (according to the n integer). The n is incremented (because we the tail of the list increases).

If the element was not added at the root, then it loops over the tree. It starts from the bottom of tree (from the parent of the added element) and the heapify function is used on each element of the array.

Therefore, given we are looping each element of the tree we have O(n). However, the heapify function has a time complexity of O(log n). Hence, the overall complexity of insert is O(n log n).