DATA STRUCTURES:

servingQueue – Queue – The queue data structure is used for because it offers O(1) insert, peek and remove time. It gives O(1) time for the entire implementation of servingQueue, because servingQueue never has any other function performed on it.

floorsQ –Queue[][]– This data structure provides O(1) access to PassengerRequests on the given floor. The first index of the queue array correlates to each floor the elevator visits and the second index is for the direction that PassengerRequest wants to travel. 0 in the second index correlates with down and 1 with up. The advantage to splitting them this way is to hold PassengerRequest in a way that can be accessed without any searching. There is no need to look past the first element in the queue to find any other.

Carriage – Queue[] – This data type was used to achieve O(1) access to PassengerRequests in the elevator carriage. Each index of carriage corresponds to the floors the elevator can visit. This way PassengerRequest can be added to the queue associated with their destination floor. A Queue was chosen also as a way to add a bit of extra protection of the order of elements in addition to O(1) access times.

Stops – LinkedList – DEPRICATED, originally this structure was to be used as a sort of track for the elevator to follow giving relatively fast access time to where the next stop is. This however required being updated very frequently and became quite expensive.

ALGORITHMS:

Linear Search – In many cases when trying to look ahead of the elevator in its direction a linear search was used to learn about the path and whether or not the elevator needs to travel in that direction and finding calls on other floors. The largest cost of this is O(n) where n is the number of floors and considering that the tallest building in the world has only 163 floors this is a very acceptable cost.

Insertion Sort – This Algorithm was originally used in a now deprecated method to provide at a reasonable cost an insert that kept a LinkedList sorted. With a cost O(n) where n is the number of elements in the list. This Algorithm, however, was excluded from the final run of the elevator because the LinkedList it acted on, stops, was deprecated in the final version.