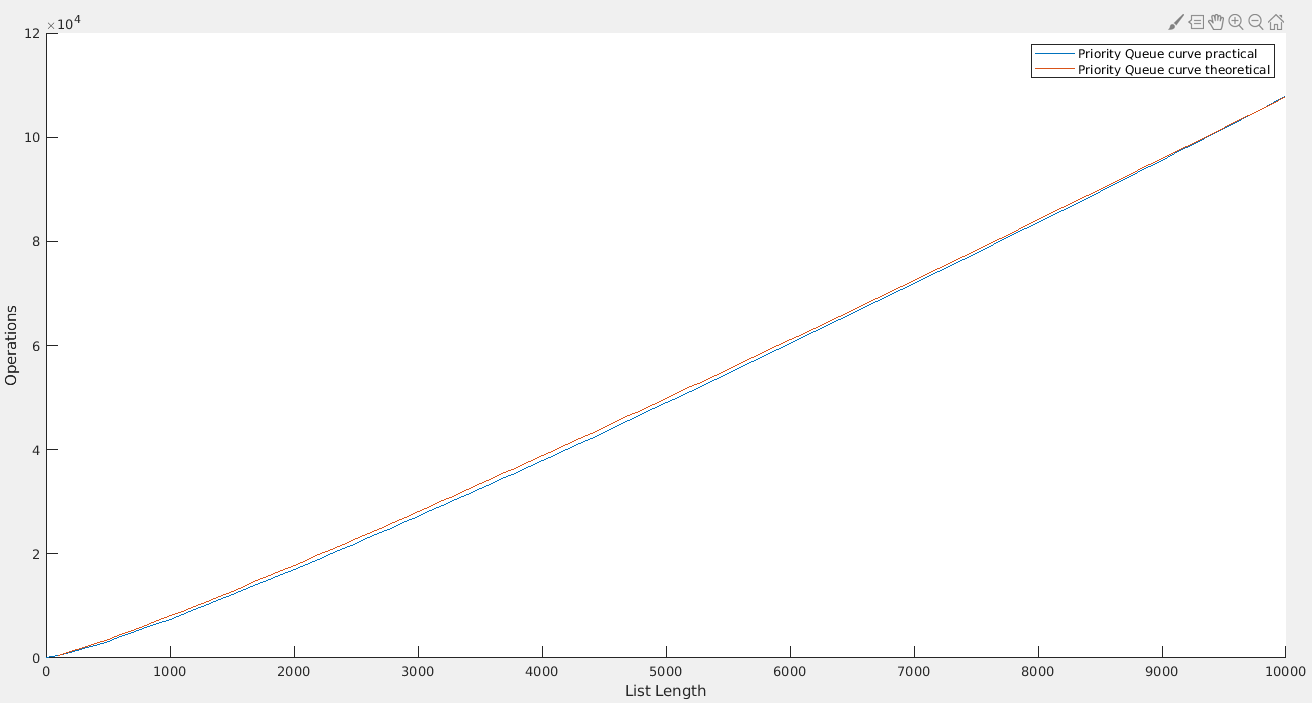
Portfolio 2

1. Write a method that solves the selection problem using a priority queue and conduct a series of experiments that indicate that the time complexity of your method is O(N log N).
2. Implement the quickSelect method and conduct a series of experiments that indicate that the time complexity of the method is O(N).
3. Solve the average case recurrence equation for quickSelect.

# 1: Priority Queue

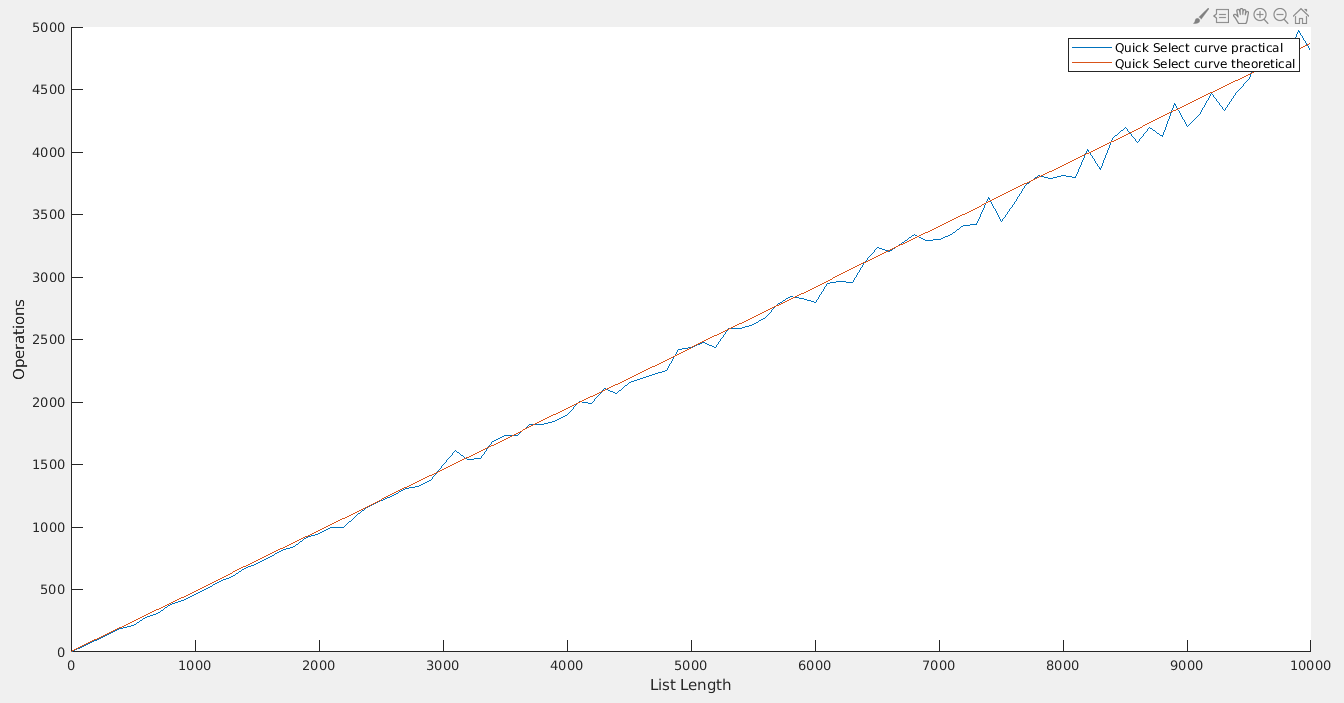
The experiment is done on a random generate list from 0 to 10 000 with a step size of 100. Each iteration is done 100 times. The average of each iteration is plotted in figure 1.

  
Figure 1: Priority Queue: Operation versus list length.

Theoretical line is plotted using function: where factor is to get the two graphs intersect each other in endpoints. The practical data is a variable counted one up each time perculate\_down is called after buildheap is done. It can be seen from the plot that the data follow pretty good

# 2: Quick Select

The experiment is done on a random generate list from 0 to 10 000 with a step size of 100. Each iteration is done 100 times. The average of each iteration is plotted in figure 2.

  
Figure 2: Quick Select: Operation versus list length

Theoretical line is plotted using function: where factor is to get the two graphs intersect each other in endpoints. The practical data is a variable counted one up each time it swaps in the while(true) inside quick\_select. This means we do not count one up if it is using insertions\_sort (only for small list sizes).

# 3: Average case Quick Select

Average case:

(1)

Multiple by N:

(2)

Substitute N with N-1:

(3)

Subtract (3) from (2):

Rearrange and drop insignificant c:

(4)

(5)

Divide equation (5) by N:

(6)

Telescope:

Adding all equations:

(7)