NAME: KARTHIK RAMAKRISHNAN
NETID: kxr150530

SHORT PROJECT - 0

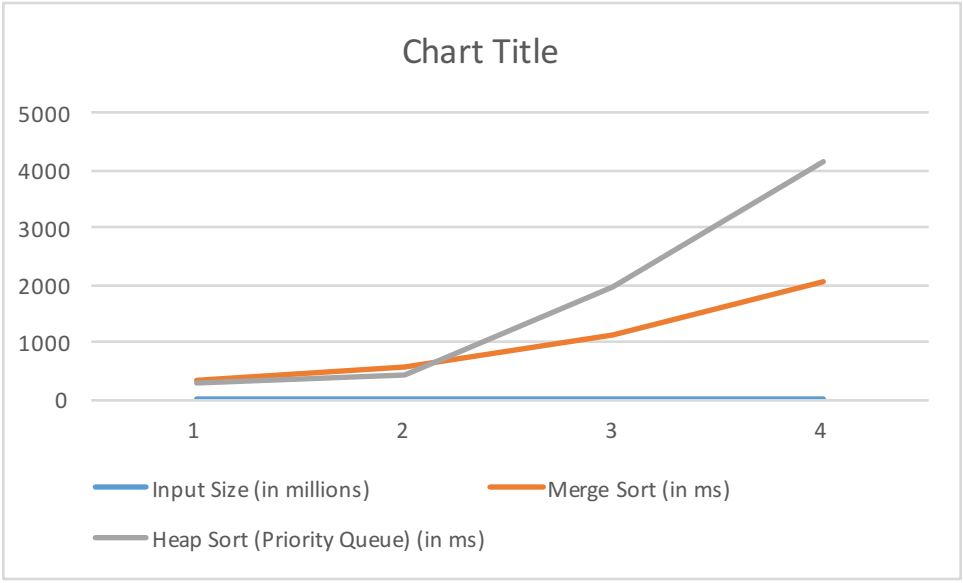
COMPARISON OF RUNNING TIMES OF MERGE SORT AND HEAP SORT (PRIORITY QUEUE)

In this project, I have implemented a simple application that compares the performance of merge sort and heap sort (using priority queue) which are both O(nlogn) algorithms over large inputs (> 1 million).

In this study, I have found the following data over multiple runs of the application.

Input Size (in millions)	Time (in milliseconds)		Memory (in MB)	
	Merge Sort	Heap Sort	Merge Sort	Heap Sort
1	305	271	39	33
2	543	426	81	67
5	1111	1977	134	173
10	2033	4125	354	297

A pictorial representation of the algorithm running times are as follows:



From the above data, we can conclude the following:

- Merge Sort is slower than heap sort for smaller inputs as is evident when input size is 1 million and 2 million
- As size increases, merge sort becomes more efficient than heap sort in terms of running time.
- Heap sort uses more memory but eventually, the memory used by merge sort is high as input size increases.

So, we can conclude that for very large inputs (> 5 million), merge sort produces quick and efficient running time than heap sort.