Name: RAJESH REDDY THIYYAGURA GWID: 25658738

8: Merging Sorted Lists

See textbook, Section 5.3
You are given an array a[] of numbers, where a[i] is the size of the i-th list to merge. You have to produce the sequence in which to merge the lists, and the total cost of merging all the lists. *Implementation Notes: Use a heap data structure. (You don't have to implement your own heap structure, you can simply use the inbuilt one in Java/C#.)*

The above problem consists of 2 sub problems

1. Sequence to merge the sorted lists
2. Cost of Merging the lists

Sequence to merge the sorted lists:

Let's take an example array consists of the lengths of the sorted lists

Int [] a = {4M,8M,2M} -- M-> Millions

To get the best sequence for merging the sorted lists by taking 2 at a time

1. If we take 4M and 8M as first operation, we get 4M +8M = 12M comparisons
   a. after that the resultant 12M is merged with 2M to get the final sorted list which takes 12M+2M = 14M comparison's
   b. In total we have 12M + 14M = 26M comparison to merge the sorted lists.

2. On the other hand,
   a. if we take 2M and 4M as first operation, we get 2M +4M = 6M comparisons
   b. after that the resultant 6M is merged with 8M to get the final sorted list which takes 6M+8M = 14M comparison's
   c. In total we have 14M + 6M = 20M comparison's

So, the second one has the least number of comparison's which will give the optimal merging for sorted lists.

Based on the example, we can see that we need to generate smallest cumulative sum, which is possible only when we sort the array and do the cumulative sum

Ex: a [] = {4,8,2}

Value	Cumulative sum
4	4
8	4+8=12
2	12+2 =14
Total comparisons	**26--million**

Ex: a [] = {2,4,8}

Value	Cumulative sum
2	2
4	2+4=6
8	6+8 =14
Total comparisons	**20--million**

,

SO, we just need to sort in ascending the initial array consisting of lengths of sorted lists to be merged

Which gives the optimal sequence for merging the sorted lists

Cost of Merging the lists:

We have used priority queue in java to implement the heap data structure in java produce the minheap,

To get the optimal solution we provided the sequence of lists based on the sequence we generated(ascending length)