**Optimal Merge Pattern:**

Merge a set of sorted files of different length into a single sorted file. We need to find an optimal solution, where the resultant file will be generated in minimum time.

If many sorted files are given, there are many ways to merge them into a single sorted file. This merge can be performed pair wise. Hence, this type of merging is called as 2-way merge patterns.

As, different pairings require different amounts of time, in this strategy we want to determine an optimal way of merging many files together. At each step, two shortest sequences are merged.

Let us consider the given files, f1, f2, f3, f4 and f5 with 20, 30, 10, 5 and 30 number of elements respectively.

If merge operations are performed according to the provided sequence, then

M1 = merge f1 and f2 => 20 + 30 = 50

M2 = merge M1 and f3 => 50 + 10 = 60

M3 = merge M2 and f4 => 60 + 5 = 65

M4 = merge M3 and f5 => 65 + 30 = 95

Hence, the total number of operations is

50 + 60 + 65 + 95 = 270

We can have a better solution, Sorting the numbers according to their size in an ascending order, we get the following sequence

f4, f3, f1, f2, f5

Hence, merge operations can be performed on this sequence

M1 = merge f4 and f3 => 5 + 10 = 15

M2 = merge M1 and f1 => 15 + 20 = 35

M3 = merge M2 and f2 => 35 + 30 = 65

M4 = merge M3 and f5 => 65 + 30 = 95

Therefore, the total number of operations is

15 + 35 + 65 + 95 = 210

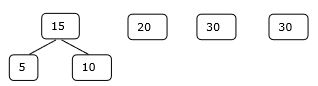
Obviously, this is better than the previous one.

In this context, we are now going to solve the problem using this algorithm.

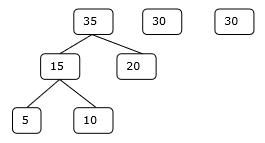
Initial Set

Initial Set

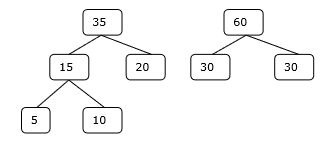
Step-1



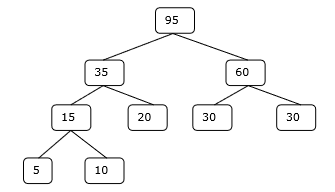
Step-2



Step-3



Step-4



Hence, the solution takes 15 + 35 + 60 + 95 = 205 number of comparisons.