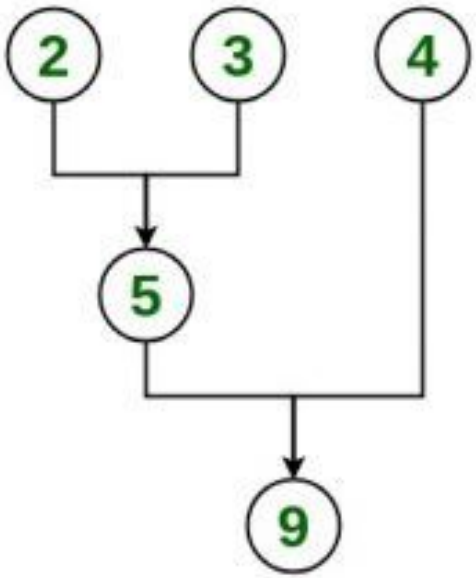
Optimal File Merge Patterns

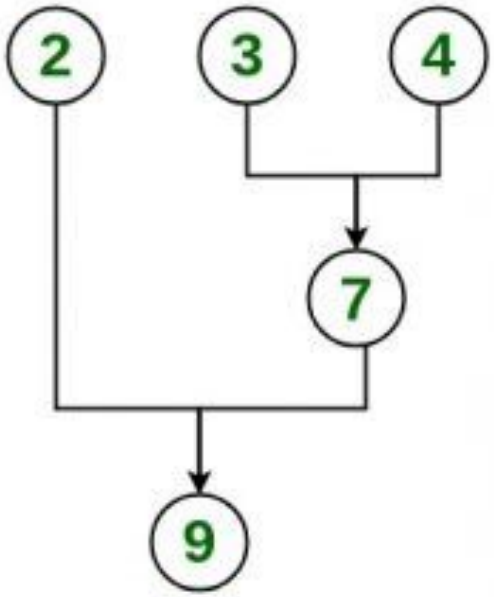
- Given n number of sorted files, the task is to find the minimum computations done to reach the Optimal Merge Pattern.
  When two or more sorted files are to be merged altogether to form a single file, the minimum computations are done to reach this file are known as **Optimal Merge Pattern**.
- If more than 2 files need to be merged then it can be done in pairs. For example, if need to merge 4 files A, B, C, D. First Merge A with B to get X1, merge X1 with C to get X2, merge X2 with D to get X3 as the output file.
- If we have two files of sizes m and n, the total computation time will be m+n. Here, we use the greedy strategy by merging the two smallest size files among all the files present.

- Given 3 files with sizes 2, 3, 4 units. Find an optimal way to combine these files
- ***Input:*** *n = 3, size = {2, 3, 4}*
  Output: *14*
  Explanation: *There are different ways to combine these files:*

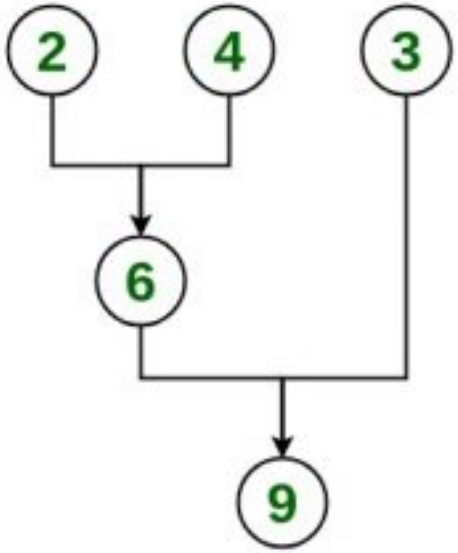
3 Methods



Cost = 5 + 9 = 14



Cost = 7 + 9 = 16

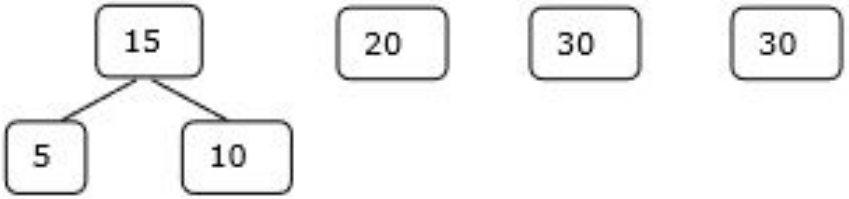


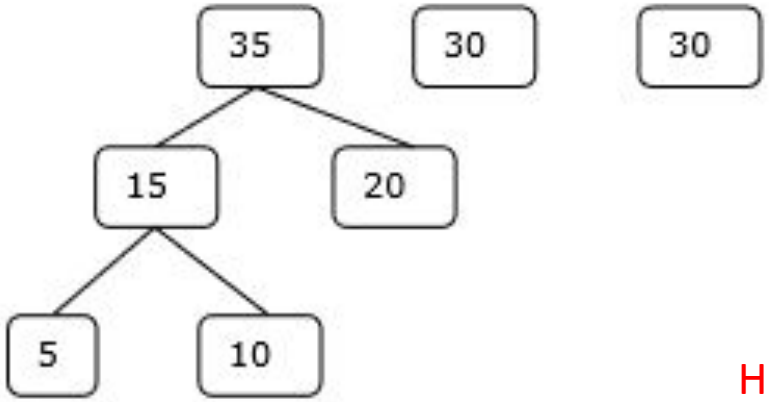
Cost = 6 + 9 = 15

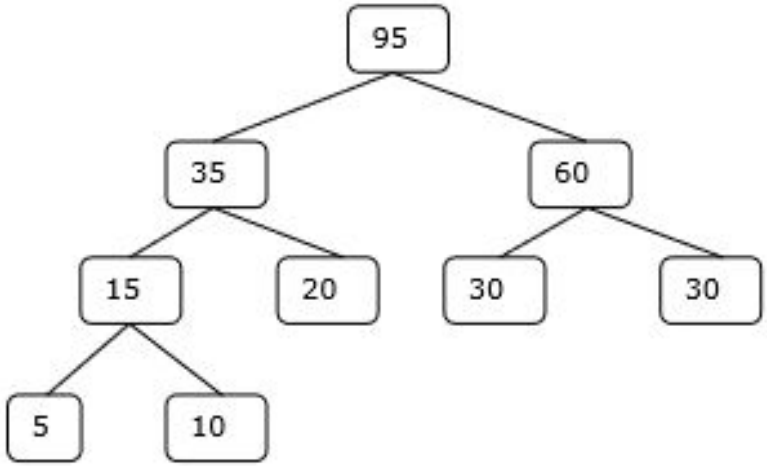
- As, different pairings require different amounts of time, in this strategy we want to determine an optimal way of merging many files together. At each step, two shortest sequences are merged.

- Let us consider the given files, f_1, f_2, f_3, f_4 and f_5 with 20, 30, 10, 5 and 30 number of elements respectively.
- If merge operations are performed according to the provided sequence, then
- **M_1 = merge f_1 and f_2** => 20 + 30 = 50
- **M_2 = merge M_1 and f_3** => 50 + 10 = 60
- **M_3 = merge M_2 and f_4** => 60 + 5 = 65
- **M_4 = merge M_3 and f_5** => 65 + 30 = 95
- Hence, the total number of operations is
- 50 + 60 + 65 + 95 = 270

- Sorting the numbers according to their size in an ascending order, we get the following sequence −
- **f_4, f_3, f_1, f_2, f_5**
- Hence, merge operations can be performed on this sequence
- **M_1 = merge f_4 and f_3** => 5 + 10 = 15
- **M_2 = merge M_1 and f_1** => 15 + 20 = 35
- **M_3 = merge M_2 and f_2** => 35 + 30 = 65
- **M_4 = merge M_3 and f_5** => 65 + 30 = 95
- Therefore, the total number of operations is
- 15 + 35 + 65 + 95 = 210







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