

Merge N files using GREEDY approach

- A. Show two different inputs for your implementation and analyse each step of your algorithm.

Enter the number of files: 3

Enter the sizes of the files:

Size of file 1: 30

Size of file 2: 20

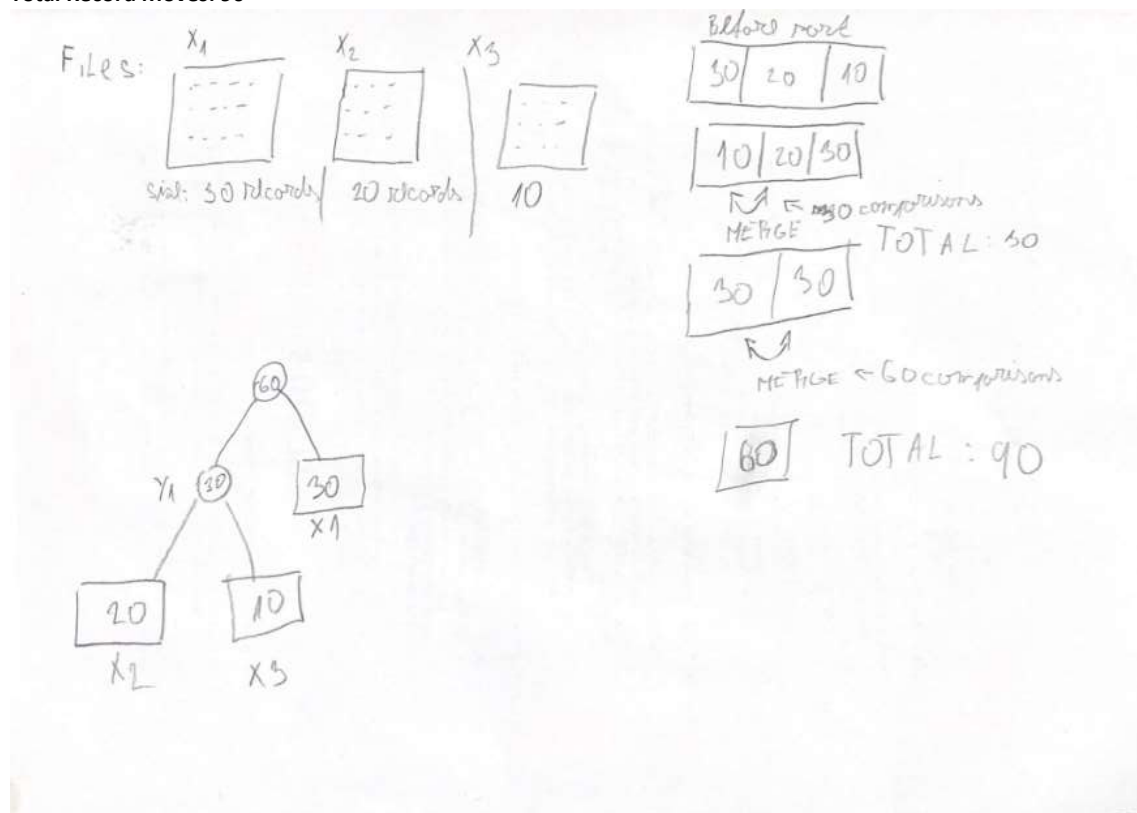
Size of file 3: 10

Merge Steps:

Merged 10 and 20 into 30 (Total Record Moves: 30)

Merged 30 and 30 into 60 (Total Record Moves: 90)

Total Record Moves: 90

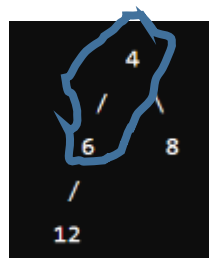


Input 2:

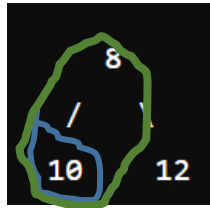
8,4,6,12

Initialization –

Representation of min heap



Action : Merging 2 smallest



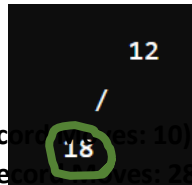
Merge Steps:

Merged 4 and 6 into 10 (Total Record Moves: 10)

Merged 8 and 10 into 18 (Total Record Moves: 28)

Merged 12 and 18 into 30 (Total Record Moves: 58)

Total Record Moves: 58



B: Analyse your algorithm and show the results using order notation.

```
void buildMinHeap(vector<int> &heap)
{
    int size = heap.size();
    // Start from the last non-leaf node and heapify each node
    for (int i = size / 2 - 1; i >= 0; i--)
        heapify(heap, size, i);
}
```

Time complexity $O(n)$

Merge operations:

Number of Merge Steps: $n-1$ (for n files)

Extracting the Two Smallest Files: $O(\log n)$

Inserting the Merged File: $O(\log n)$

Time complexity per merge $O(2\log n) = O(\log n)$

Time complexity for all merge steps $O((n-1)\log n) = O(n\log n)$

Sources of my study info: <https://www.vaia.com/en-us/textbooks/computer-science/foundations-of-algorithms-using-c-pseudocode-3-edition/chapter-4/problem-37-use-a-greedy-approach-to-write-an-algorithm-that-/>

<https://www.youtube.com/watch?v=xXSm2hiDWWQ>

<https://www.geeksforgeeks.org/optimal-file-merge-patterns/>