



9- External Sorting

External sorting is a term for a class of sorting algorithms that can handle massive amounts of data. External sorting is required when the data being sorted do not fit into the main memory of a computing device (usually RAM) and instead they must reside in the slower external memory (usually a hard drive). External sorting typically uses a hybrid sort-merge strategy. In the sorting phase, chunks of data small enough to fit in main memory are read, sorted, and written out to a temporary file. In the merge phase, the sorted sub-files are combined into a single larger file.

One example of external sorting is the **external merge sort algorithm**, which sorts chunks that each fit in RAM, then merges the sorted chunks together. We first divide the file into runs such that the size of a run is small enough to fit into main memory. Then sort each run in main memory using merge sort sorting algorithm. Finally merge the resulting runs together into successively bigger runs, until the file is sorted.

*Note: **Merge** works without loading either input list into memory all at once.*

Below are the steps used in implementation:

Inputs:

input_file : Name of input file. input.txt
output_file : Name of output file, output.txt
run_size : Size of a run (can fit in RAM)
num_ways : Number of runs to be merged

Output:

- 1) Read input_file such that at most 'run_size' elements are read at a time.
Do following for the every run read in an array:
 - a) Sort the run using MergeSort.
 - b) Store the sorted run in a temporary file, say 'i' for i'th run.
- 2) Merge the sorted files using the approach discussed here

Suppose we have a RAM which can only hold 2 chunks of data and we have 6 chunks of data to sort. Please see the below diagram:

