

# ACM SIGMOD Programming Contest 2019

Team: PaperCup

Hyunsoo Cho, Kihwang Kim, Jaeseon Yu

Hyungsoo Jung(Advisor)

{hyunsoocho, kihwangkim, jaeseonyu}@hanyang.ac.kr

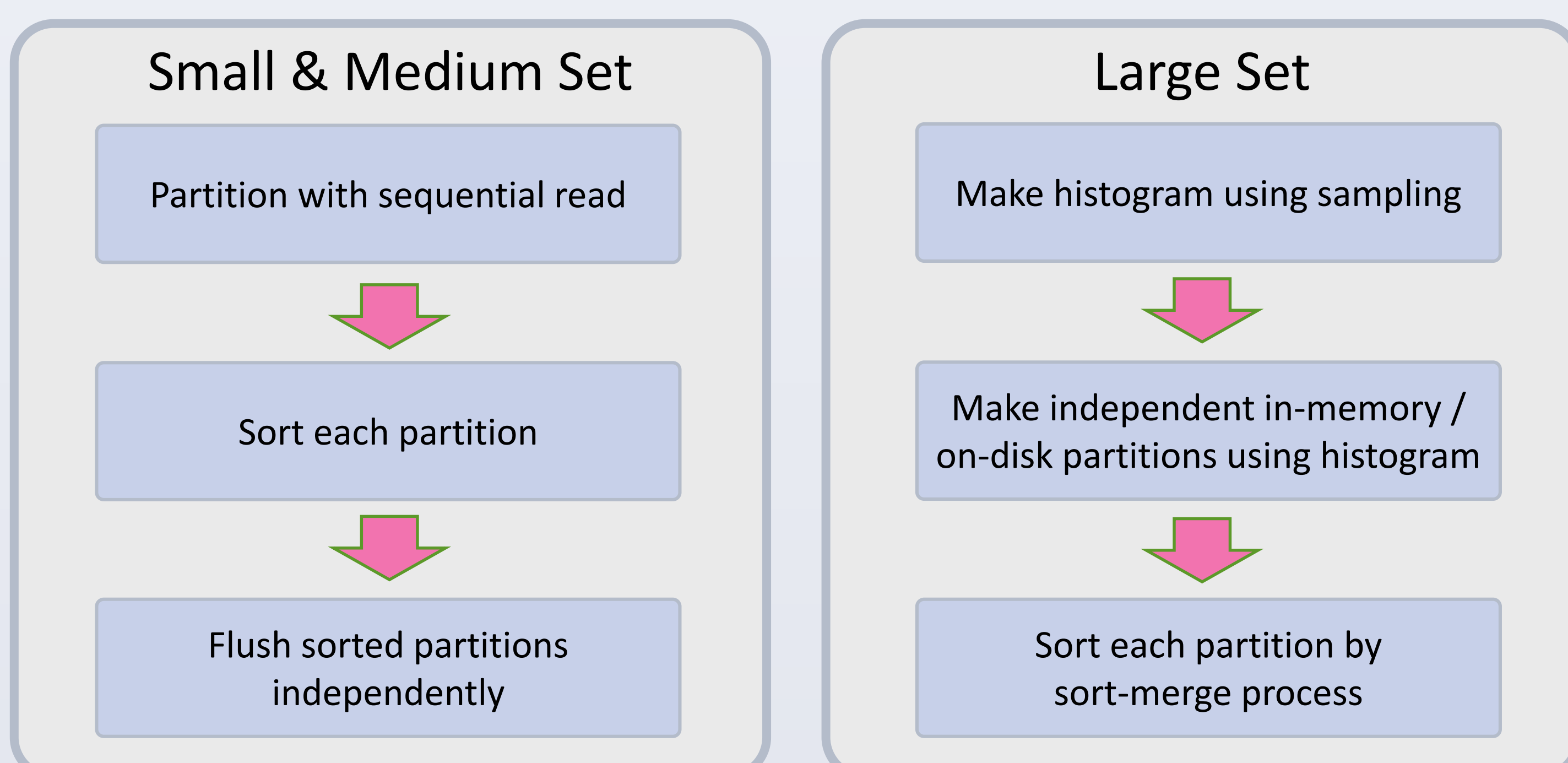
{hyungsoo.jung}@hanyang.ac.kr



## 1. Task Overview

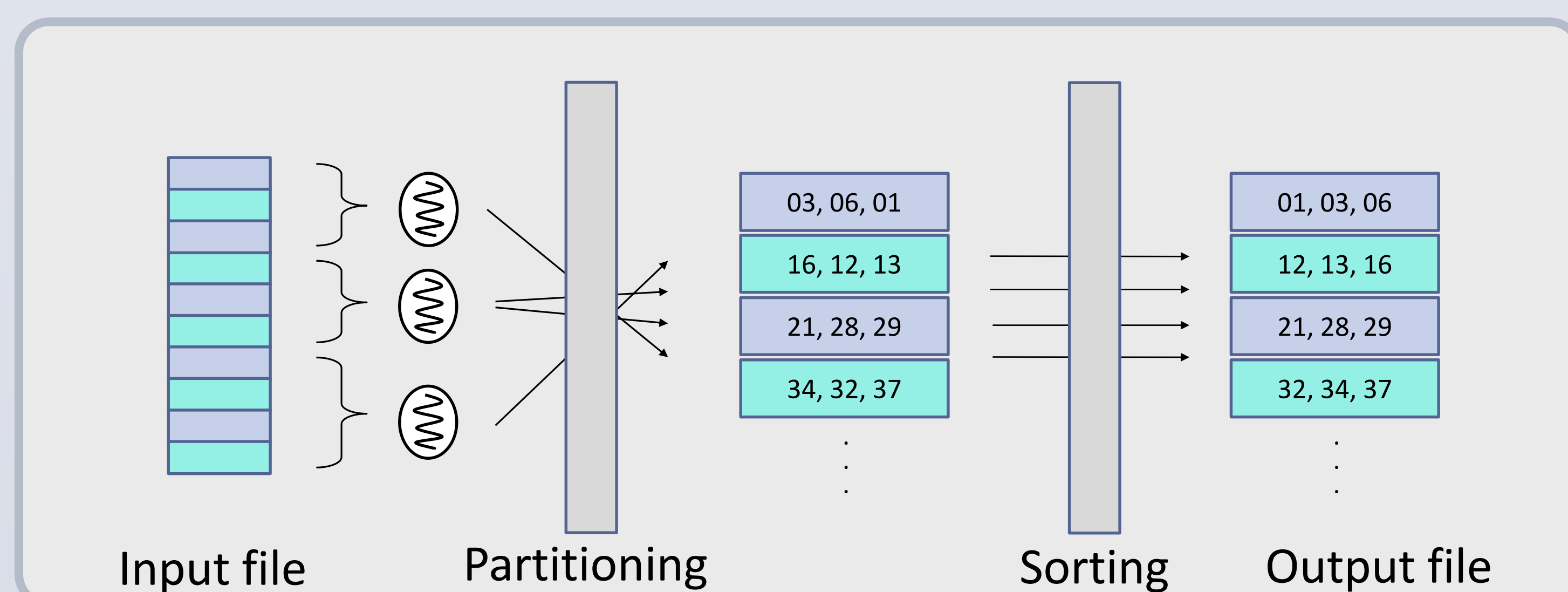
- The task is to sort given databases, as quickly as possible.
- Firstly, The files to be used are in the format generated by gensort, where the first 10 bytes are key and the last 90 bytes the payload.
- Next, sort the given input file and then write to output file. Sorting will be validated by valsart.

## 2. Solution Overview



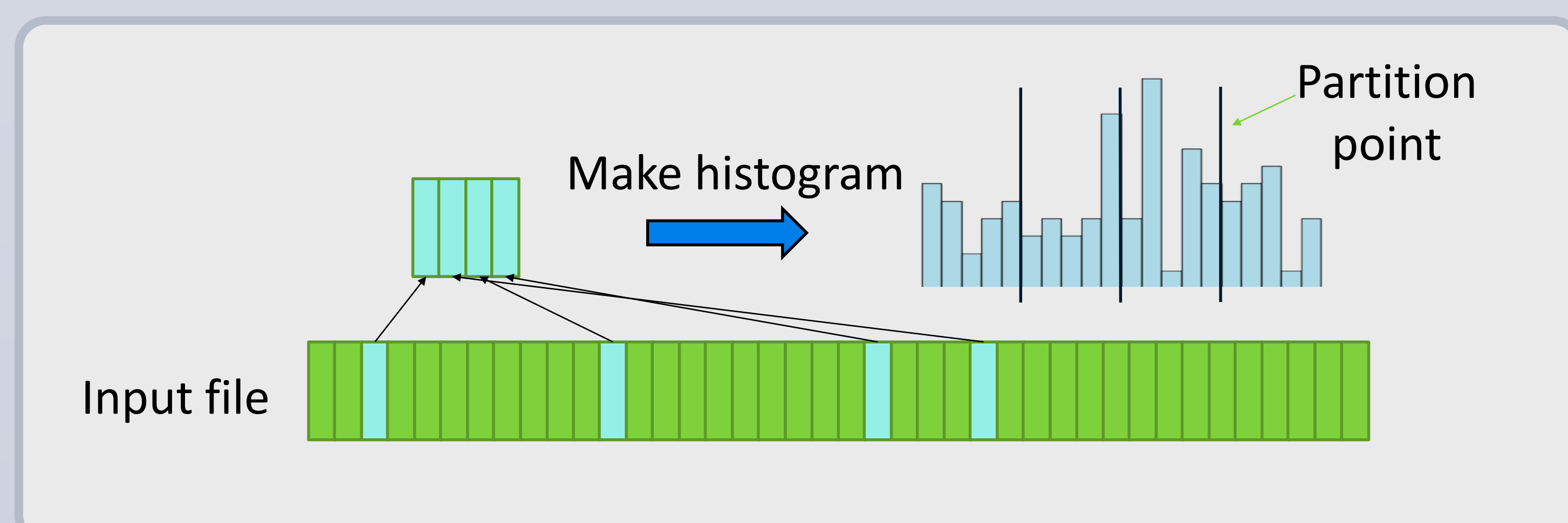
## 3. Small & Medium Set

- Sorting is processed by following order
  - Make independent partitions based on the leftmost 15 bits of key.
  - Sort each partition by quick sort and write to output file.

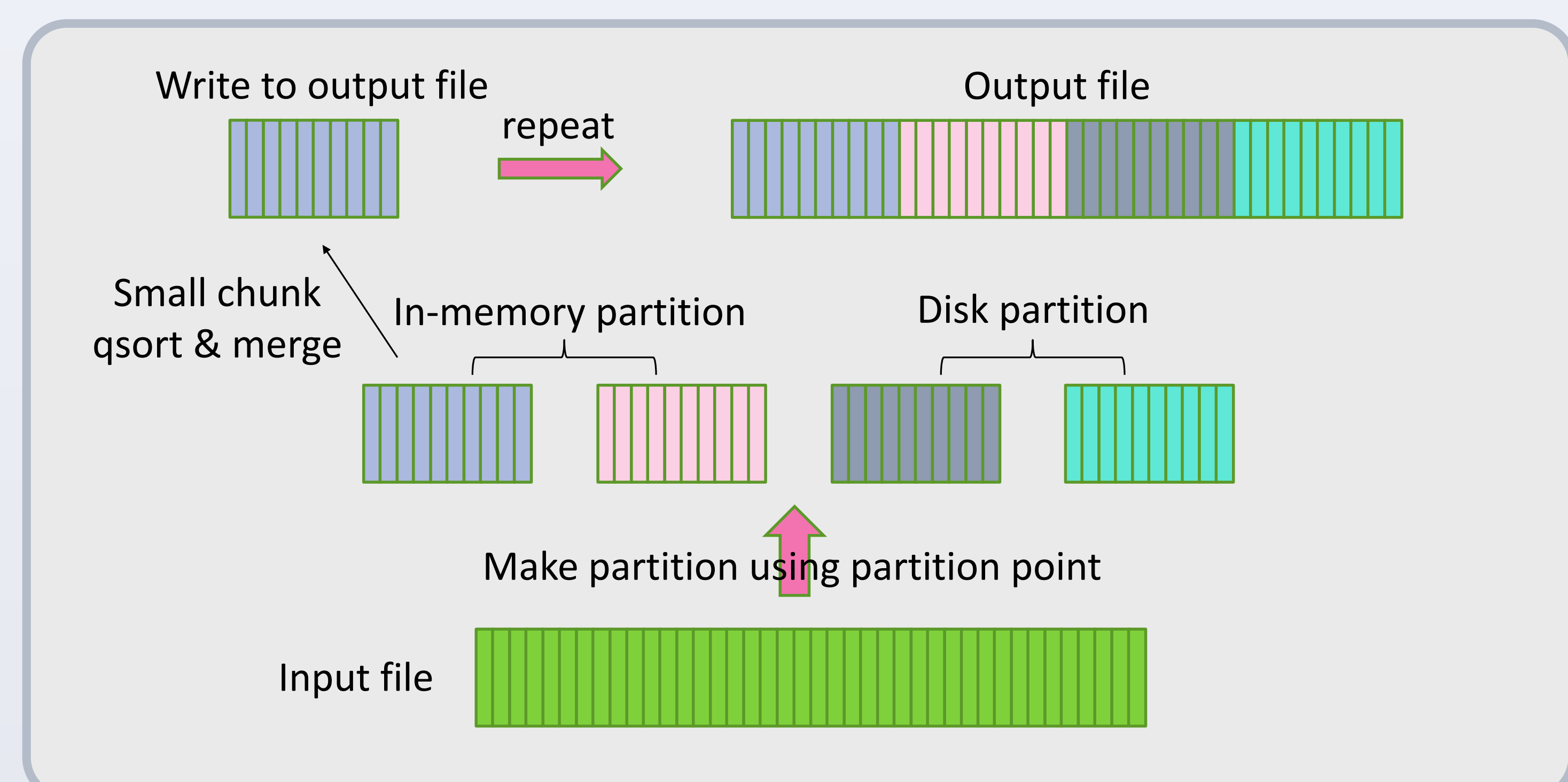


## 4. Large Set

- Partitioning is processed by following order
  - Sample 4G of the input file with 100MB chunk.
  - Make a histogram based on the leftmost 3 bytes of sampling keys.
  - Using histogram accumulation, find the partition points.
  - Make independent partitions using partition points.



- Sorting is processed by following order
  - Read a partition from disk to memory.
  - Order a partition by quick sort and merge sort. Write to disk when it's done.
  - Repeat 1-2 until every partitions is sorted.



## 5. Optimizations

### In General

- All operations, including sorting, merging and so on, are executed as a task in the thread pool.
- Use Chunk Manager for pre-memory allocation and efficient memory usage.
- Use Direct I/O for bypassing the operation system read caches.
- Use byte comparison instead of bit.

### In Small & Medium

- Adjust each partition to L3 cache size because of quick sort's performance.
- (Unreleased) Apply prefetch to pre-allocate kernel cache buffer for writing.

### In Large

- Use multi-way(15) merge for efficiency.
- Use tournament tree instead of heap in merge task for performance.
- While a partition is written to disk, the next partition file is read simultaneously for using full I/O bandwidth.

