let n_i be the length of the i-th sorted list, and let $n = \sum_{i=1}^{k} n_i$.
1. use heap. $O(n \log k)$.
2. merge sort, use $\log k$ rounds. $O(n \log k)$.
3. we can merge two sorted lists with length n and m in $O(n \log \frac{m}{n})$ (wlog assume $n \leq m$) [1], see 021. Merge Two Sorted Lists. for $i = 1 \dots k$, merge the i-th list with the result list in $O(n_i \log \frac{n}{n_i})$. in total $O(n \log n - \sum_{i=1}^{k} n_i \log n_i)$ time.

for comparison based algorithms, the information-theoretic lower bound is $\log \frac{n!}{\prod_{i=1}^{k} n_i!} = \Omega(n \log n - \sum_{i=1}^{k} n_i \log n_i)$, so the algorithm is tight.

References

[1] Mark R Brown and Robert E Tarjan. A fast merging algorithm. Technical report, STANFORD UNIV CALIF DEPT OF COMPUTER SCIENCE, 1977.