Algorithms & Complexity 2/6/2017 – 2/10/2017

0145-344-001

Note Taker: Jai Punjwani

Topic: External Sorting

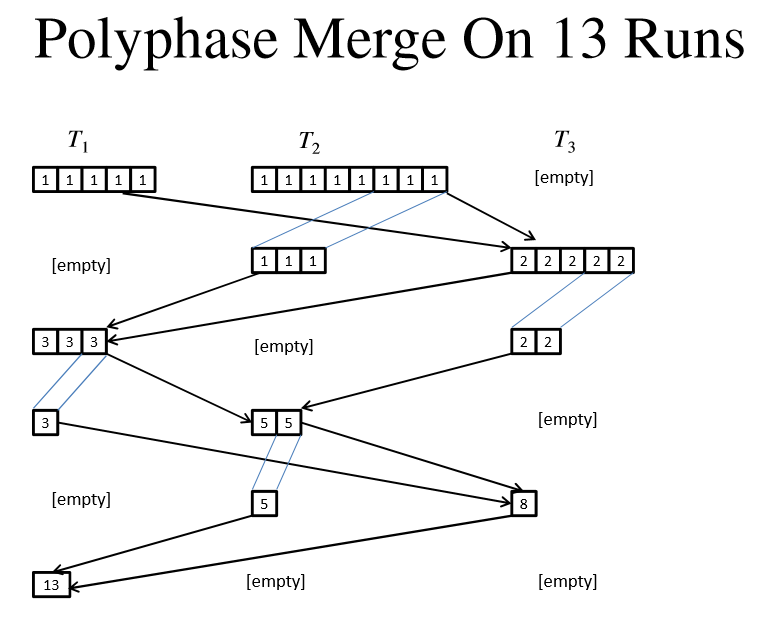
PowerPoint: <http://home.adelphi.edu/~siegfried/cs344/344l4.pdf>

NOTES

* Sometimes a file cannot all fit in memory so it has to be broken into chunks, which are sorted separately and then merged
* In order to make this process as efficient as possible, we minimize the read/write operations
* We break our data into chunks called “**runs**”
* Which **internal sort** do we use when sorting our runs? Quick sort has a worst case of n2 if the data is sorted. Merge sort uses an order n space complexity, but our issue here is space (hence we break up our data). Thus, we settle for a **heap sort** which always has complexity ***O***(n log n).

**Polyphase Merging**

* We have p tapes. p-1 tapes are used to distribute the data and sort it. 1 tape is always left empty. After sorting our p-1 tapes, we empty the tape with the least runs, and that becomes our empty tape.



Note that 13 runs here means that there are 13 elements. We merge the number of elements after sorting them.