**External Sorting**

**Example of Two-Way Sorting:**

N = 14, M = 3 (14 records on tape Ta1, memory capacity: 3 records.)

**Ta1:** 17, 3, 29, 56, 24, 18, 4, 9, 10, 6, 45, 36, 11, 43

1. **Sorting of runs:**
   1. **Read 3 (M) records in main memory, sort them and store them on Tb1:**

17, 3, 29 -> 3, 17, 29

**Tb1: 3, 17, 29**

* 1. **Read the next 3 records in main memory, sort them and store them on Tb2**

56, 24, 18 -> 18, 24, 56

**Tb2: 18, 24, 56**

* 1. **Read the next 3 records in main memory, sort them and store them on Tb1**

4, 9, 10 -> 4, 9, 10

**Tb1:** 3, 17, 29, **4, 9, 10**

* 1. **Read the next 3 records in main memory, sort them and store them on Tb2**

6, 45, 36 -> 6, 36, 45

**Tb2:** 18, 24, 56, **6, 36, 45**

* 1. **Read the next 3 records in main memory, sort them and store them on Tb1**

**(there are only two records left)**

11, 43 -> 11, 43

**Tb1:** 3, 17, 29,4, 9, 10, **11, 43**

At the end of this process we will have three runs on Tb1 and two runs on Tb2:

**Tb1:** 3, 17, 29 | 4, 9, 10 | 11, 43

**Tb2:** 18, 24, 56 |6, 36, 45 |

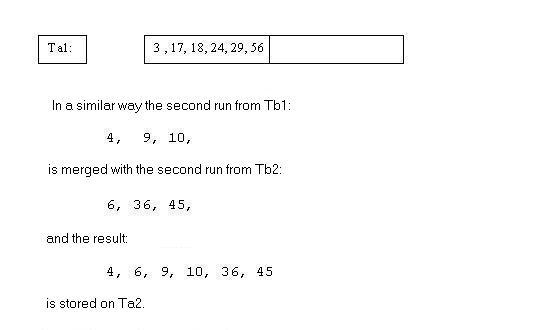
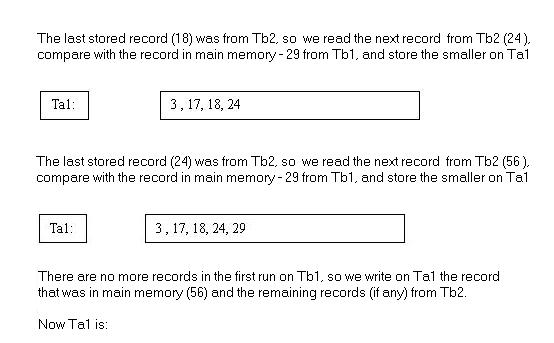
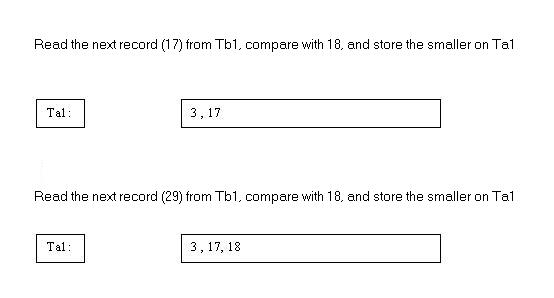
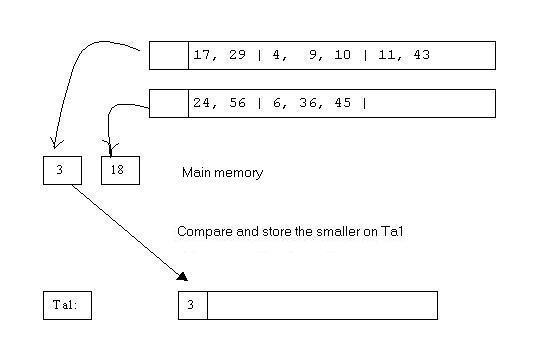
1. **Merging of runs**

**B1. Merging runs of length 3 to obtain runs of length 6.**

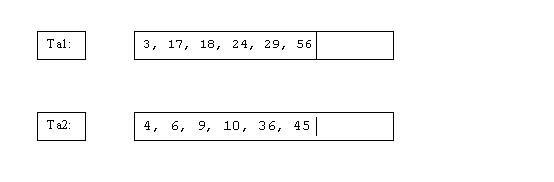
**Source tapes:** Tb1 and Tb2, result on Ta1 and Ta2.

Merge the first two runs (on Tb1 and Tb2) and store the result on Ta1.

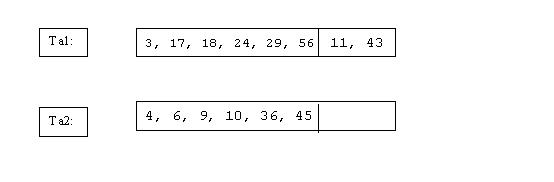
**Tb1:** 3, 17, 29 |4, 9, 10 | 11, 43

**Tb2:** 18, 24, 56 |6, 36, 45 |

Thus we have the first two runs on Ta1 and Ta2, each twice the size of the original runs:



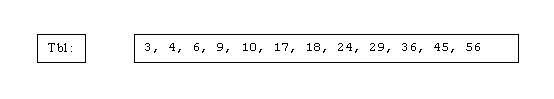
Next we merge the third runs on Tb1 and Tb2 and store the result on Ta1. Since only Tb1 contains a third run, it is copied onto Ta1:



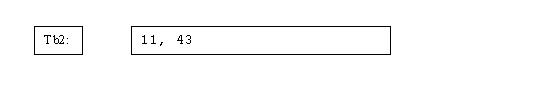
**B2. Merging runs of length 6 to obtain runs of length 12.**

**Source tapes:** Ta1 and Ta2. Result on Tb1 and Tb2:

After merging the first two runs from Ta1 and Ta2, we get a run of length 12, stored on Tb1:



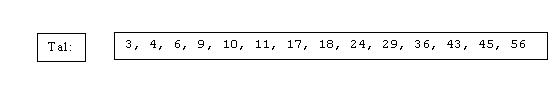
The second set of runs is only one run, copied to Tb2



Now on each tape there is only one run. The last step is to merge these two runs and to get the entire file sorted.

**B3. Merging the last two runs.**

The result is:



Number of passes: **log(N/M)**

In each pass the size of the runs is doubled, thus we need **[log(N/M)]+1** to get to a run equal in size to the original file. This run would be the entire file sorted.

In the example we needed three passes (B1, B2 and B3) because [Log(14/3)] + 1 = 3.