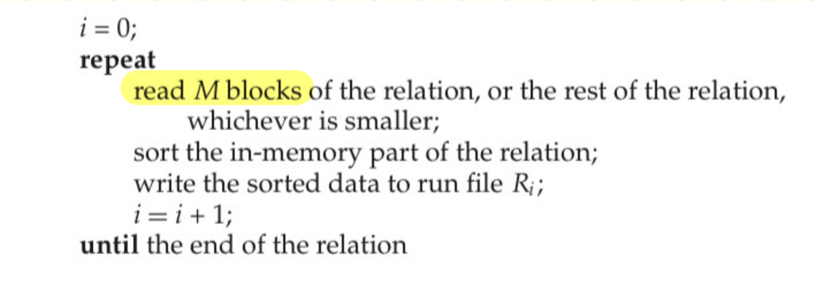
Sorting (external)

9/17

External merge sort

Let M be the # of blocks in main memory buffer for sorting

① Fill buffer, sort buffer, write buffer
each iteration is called a run



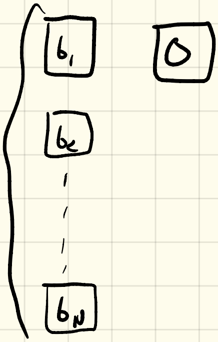
```
$i = 0$;
repeat
    read $M$ blocks of the relation, or the rest of the relation,
        whichever is smaller;
    sort the in-memory part of the relation;
    write the sorted data to run file $R_i$;
    $i = i + 1$;
until the end of the relation
```

② merge step

Let N be the # of runs that we perform

$i = 3$

assume $N < M$



the hard case
$N >> M$

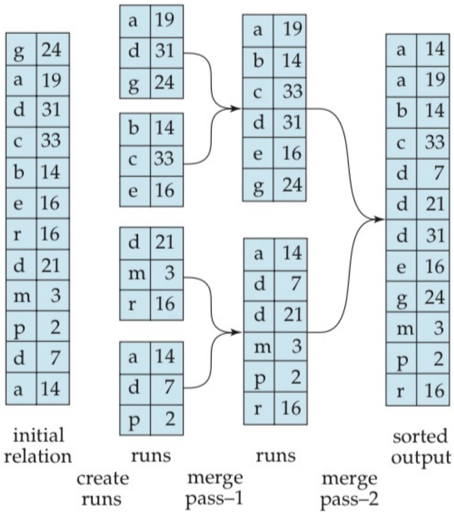
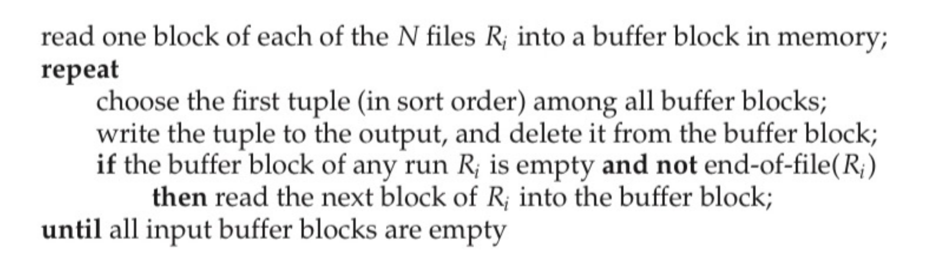


Figure 12.4 External sorting using sort–merge.

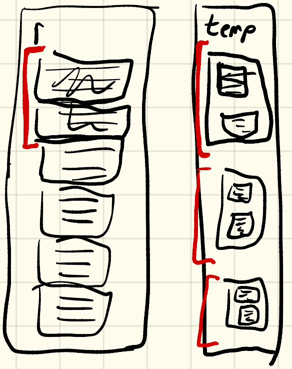
each pass reduces $M-1$ runs
* we need 1-block for output buffer



```
read one block of each of the $N$ files $R_i$ into a buffer block in memory;
repeat
    choose the first tuple (in sort order) among all buffer blocks;
    write the tuple to the output, and delete it from the buffer block;
    if the buffer block of any run $R_i$ is empty and not end-of-file($R_i$)
        then read the next block of $R_i$ into the buffer block;
until all input buffer blocks are empty
```

Analysis of external merge sort
(disk access)

b_r is # of blocks containg relation r



① read all blocks
(sort)
write all blocks
→ $2b_r$ transfers

② # of runs $\lceil b_r/M \rceil$

③ # of merge passes $\lceil \log_{M-1}(b_r/M) \rceil$

④ each merge pass reads and writes all blocks
($2b_r$ blocks)

$\Rightarrow b_r\left(2\left\lceil \log_{M-1}\left(b_r/M\right)\right\rceil + 1\right)$ ⟵ not couting the final write

✱ Account for # of seeks