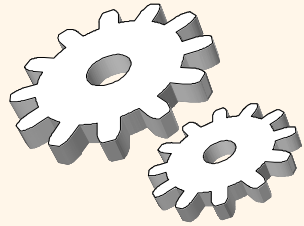
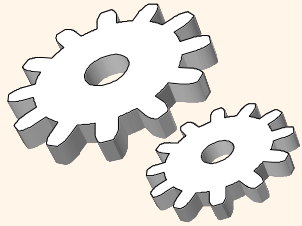
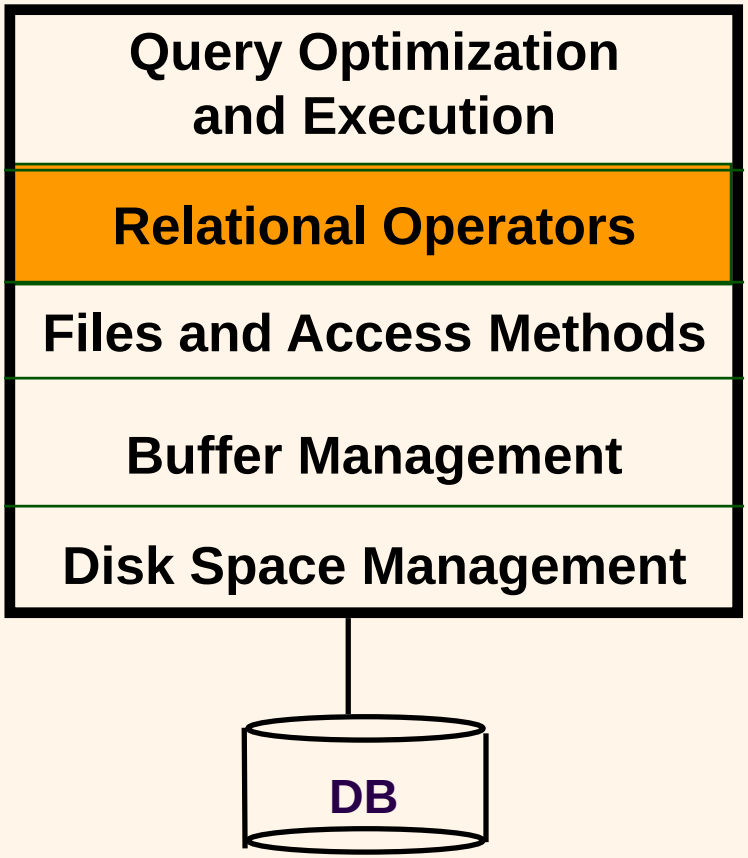


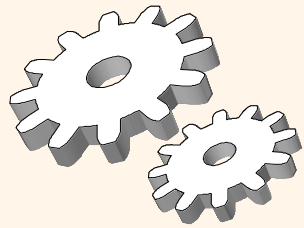
COMP7640
Database Systems & Administration

External Sort

Where Are We Now?



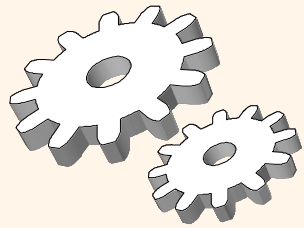




External Sort

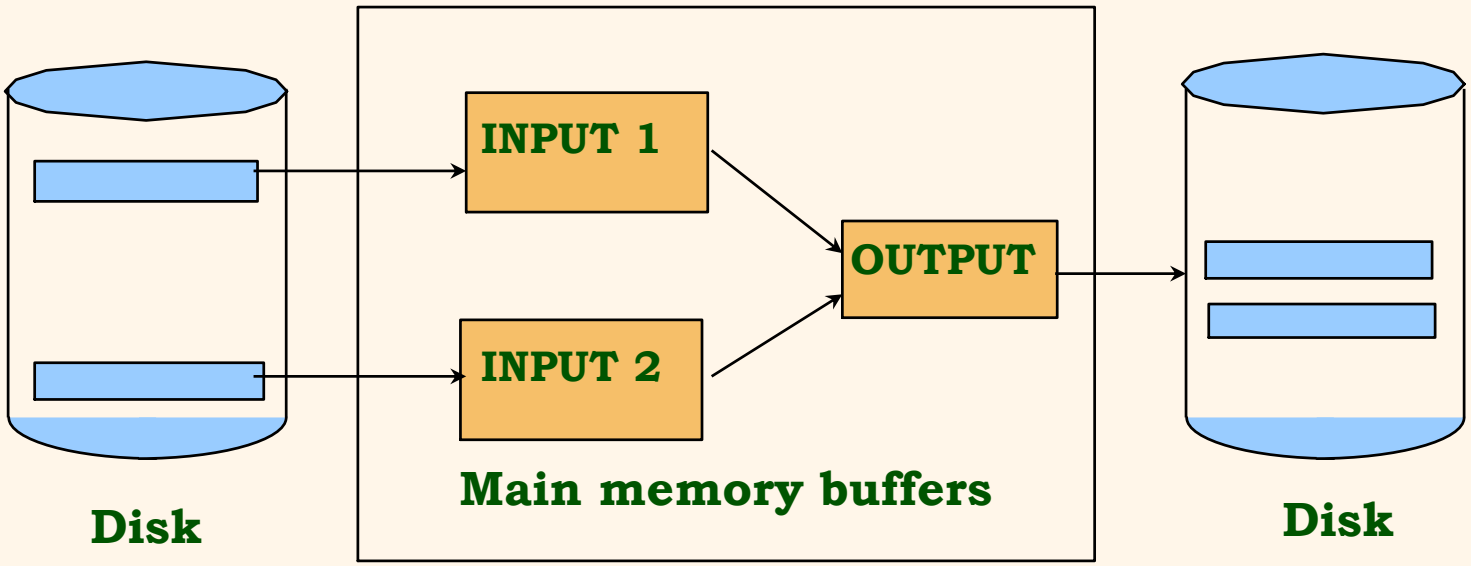
- A classic problem in computer science
  - Data requested in *sorted* order
    - e.g., find students in increasing GPA order
  - Applications:
    - Sorting is first step in bulk loading B+ tree index
    - Sorting useful for eliminating *duplicate* copies in a collection of records
    - Join algorithms involve sorting

External Sort



- Problem:
  - Sort 1GB of data with 1MB of RAM
- Solution:
  - Use external sorting algorithms
  - Need to minimize the number of disk access (i.e., I/Os)
    - Only counts #I/Os
    - No cost for any CPU operations

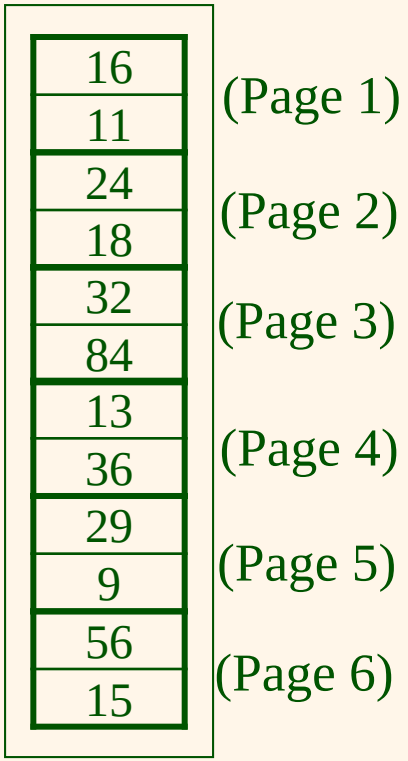
External Merge Sort (3 Buffers)



Only stores 3 pages in the main memory.
2 pages for input. 1 page for output.

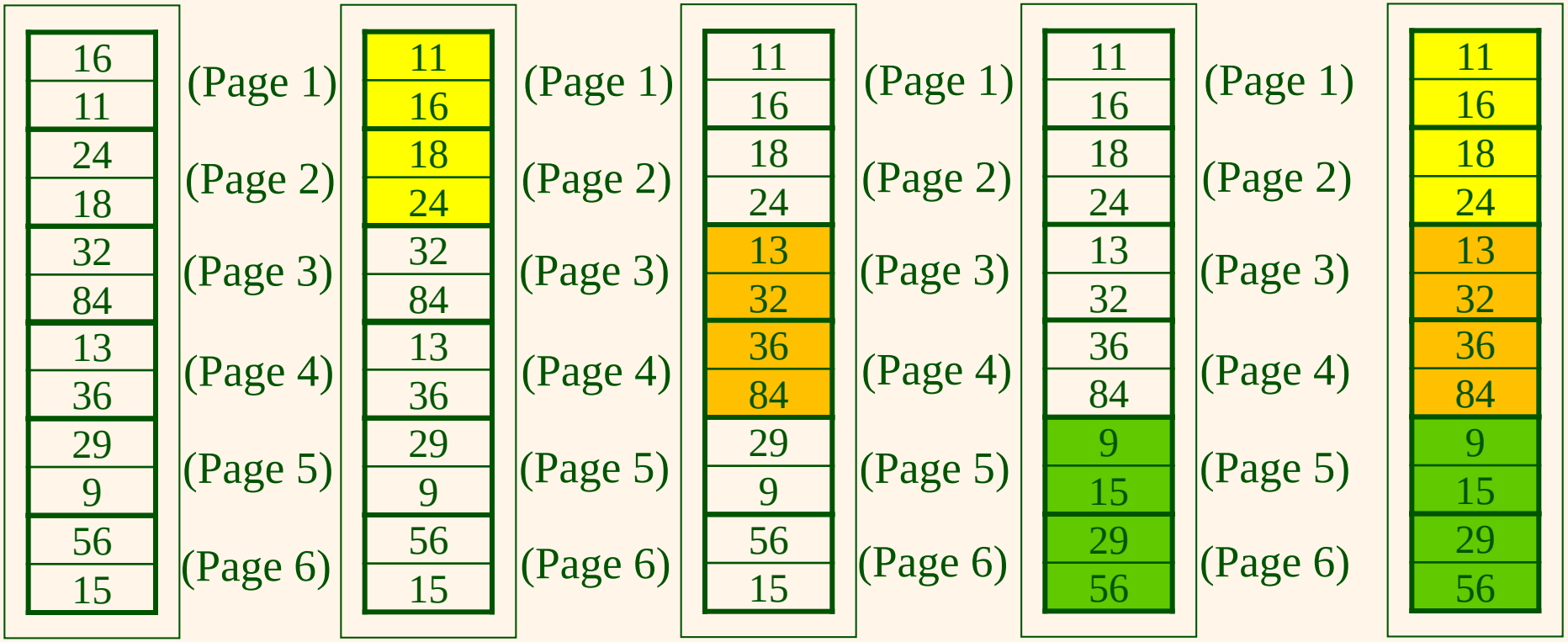
External Merge Sort (3 Buffers)

- Pass 0 (Iteration)
  - Read every 2 pages from the disk.
  - Sort these 2 pages in the memory.
  - Write these pages back to the disk.



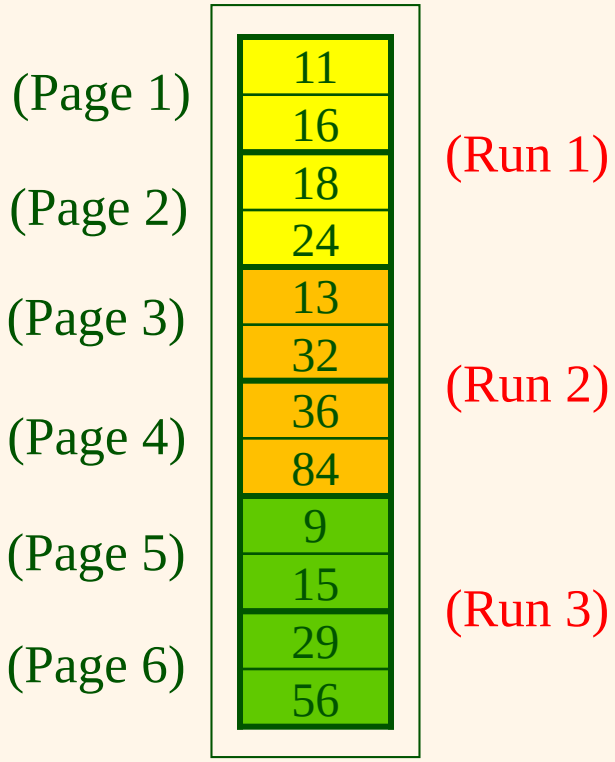
- Example:
  - Each page contains two records.

Pass 0


16
11
24
18
32
84
13
36
29
9
56
15
(Page 1)
(Page 2)
(Page 3)
(Page 4)
(Page 5)
(Page 6)
11
16
18
24
32
84
13
36
29
9
56
15
(Page 1)
(Page 2)
(Page 3)
(Page 4)
(Page 5)
(Page 6)
11
16
18
24
13
32
36
84
29
9
56
15
(Page 1)
(Page 2)
(Page 3)
(Page 4)
(Page 5)
(Page 6)
11
16
18
24
13
32
36
84
9
15
29
56
(Page 1)
(Page 2)
(Page 3)
(Page 4)
(Page 5)
(Page 6)
11
16
18
24
13
32
36
84
9
15
29
56

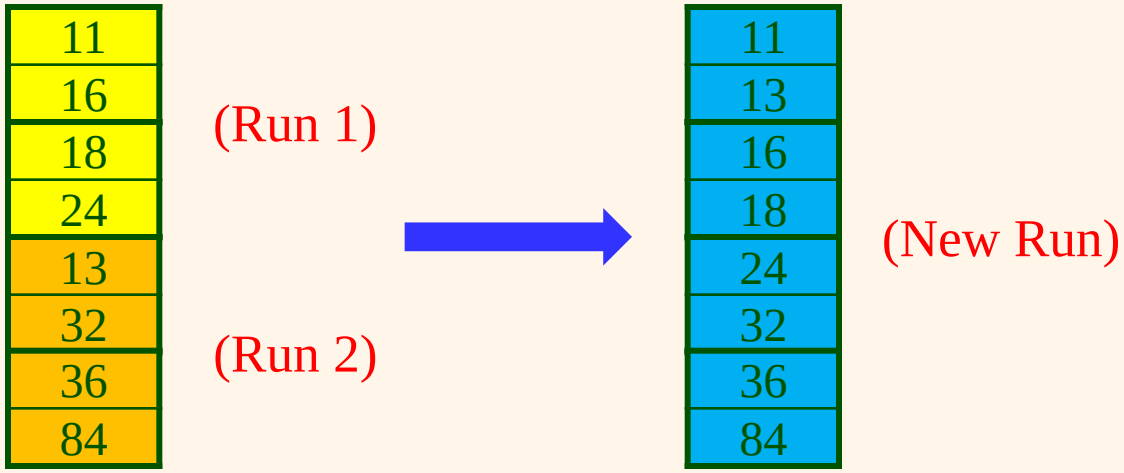
What is Run?

- Run: sorted subfile



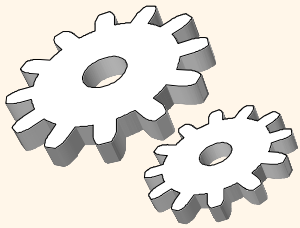
External Merge Sort with 3 Buffers

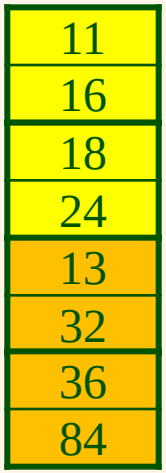
- Pass 1,2… (Iteration)
  - Identify the next 2 runs.
  - Merge these 2 runs (with an increasing order).



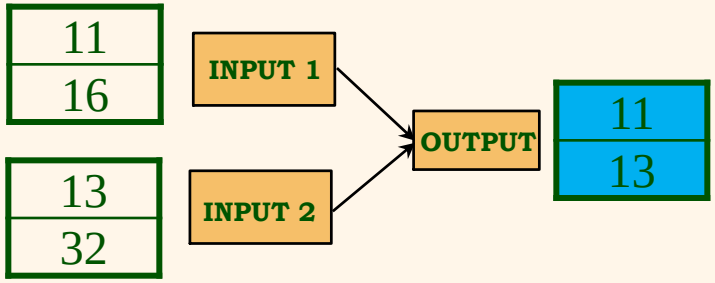
- Question:
  - How to merge these two runs?

How to Merge These Two Runs?

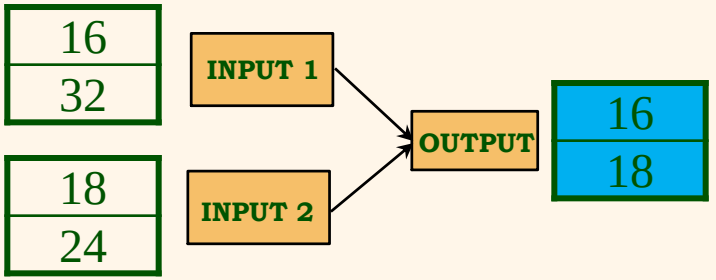




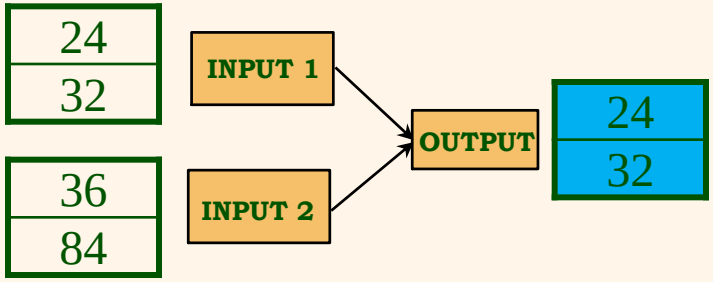
Two runs



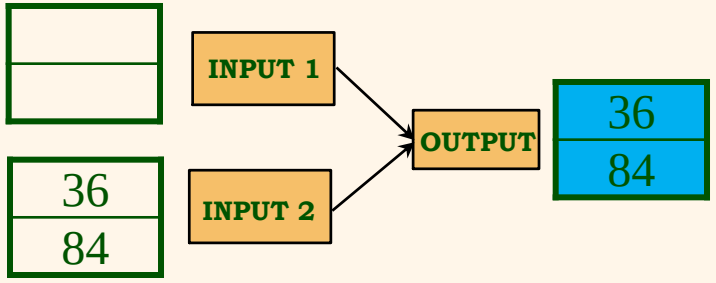
(Step 1)



(Step 2)

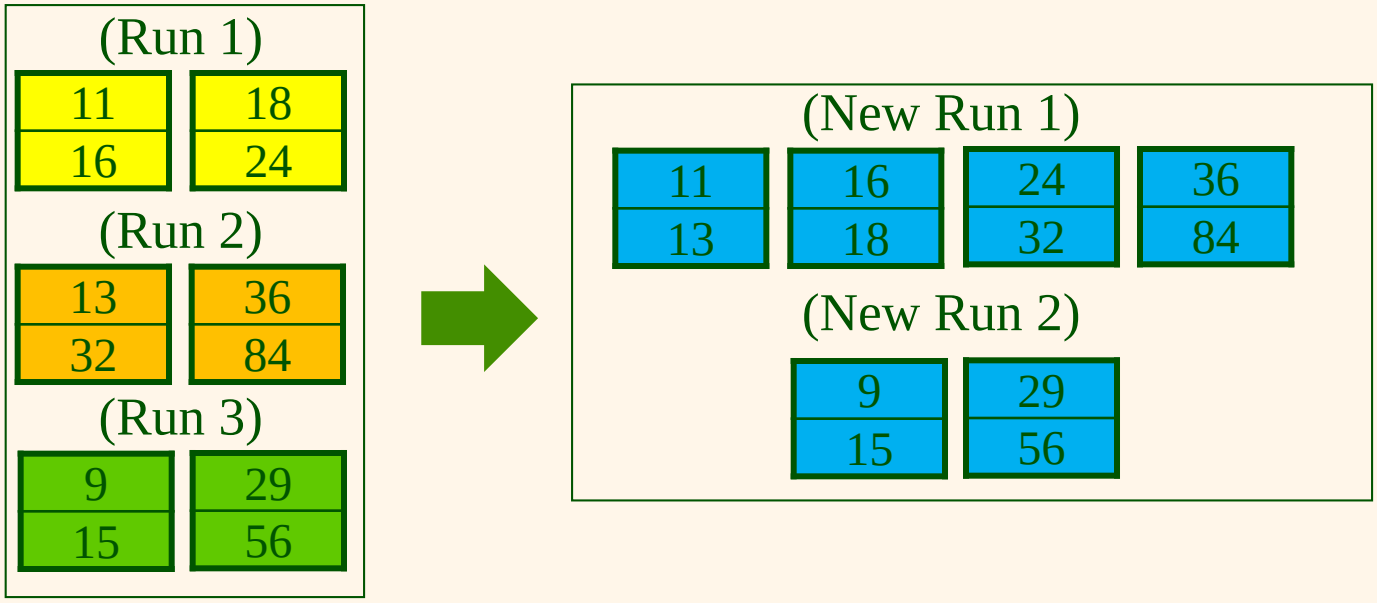


(Step 3)

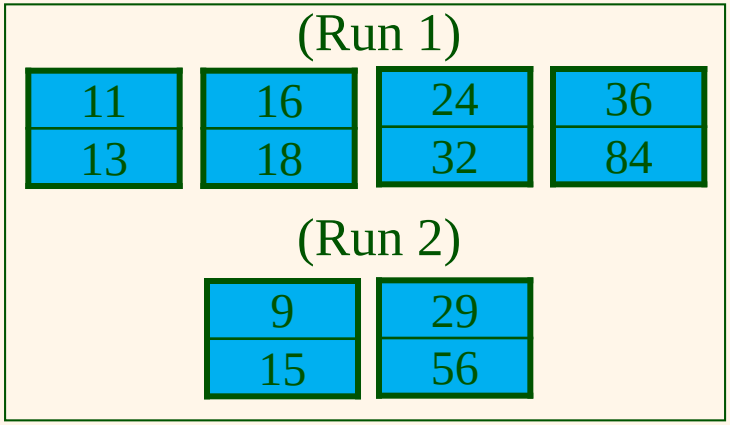


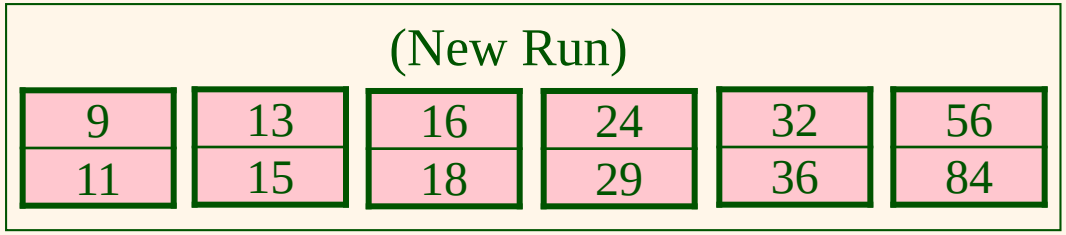
(Step 4)

Pass 1

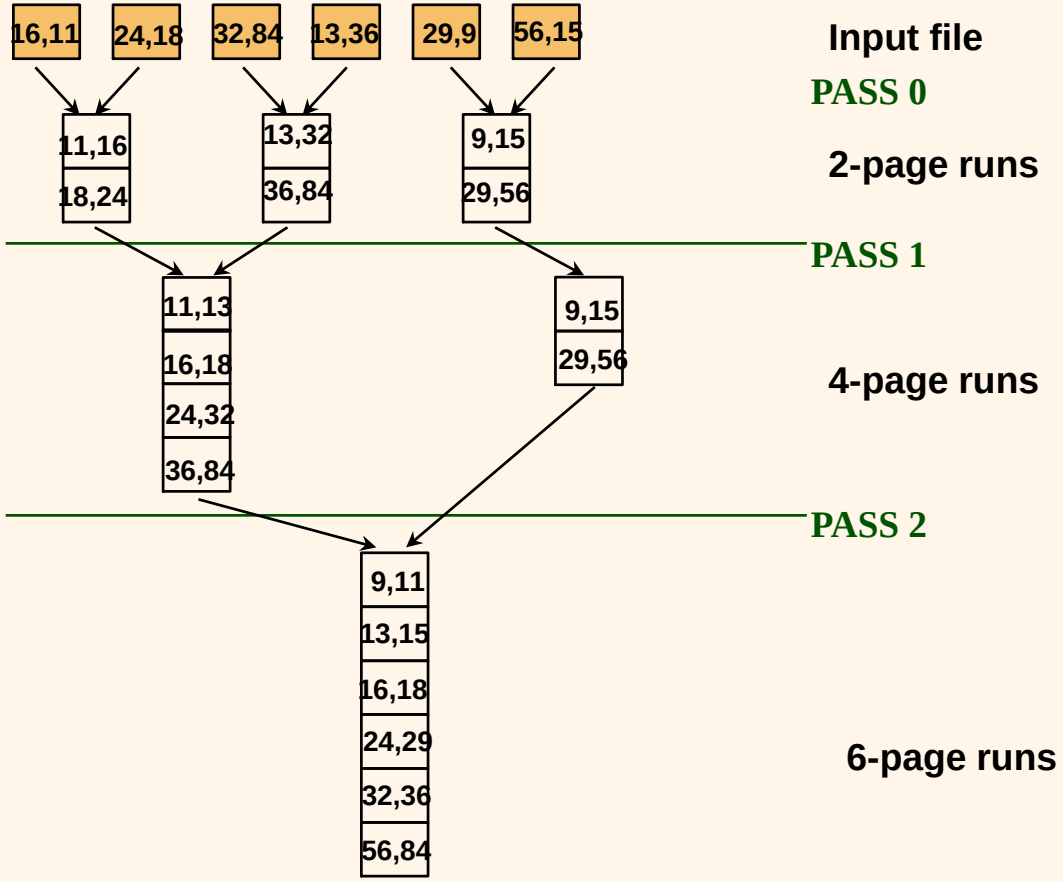

(Run 1)
11
16
18
24
(Run 2)
13
32
36
84
(Run 3)
9
15
29
56
(New Run 1)
11
13
16
18
24
32
36
84
(New Run 2)
9
15
29
56

Pass 2

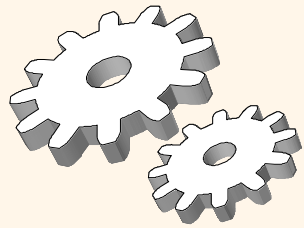

(Run 1)
11
13
16
18
24
32
36
84
(Run 2)
9
15
29
56


(New Run)
9
11
13
15
16
18
24
29
32
36
56
84

External Merge Sort (3 Buffers)

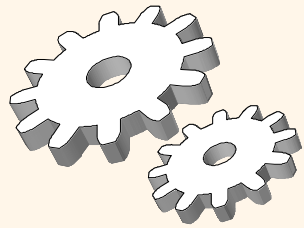


I/O Cost per Pass



- **Pass 0**: 2 x 6 = 12 I/Os.
- **Pass 1**: 2 x 6 = 12 I/Os.
- **Pass 2**: 2 x 6 = 12 I/Os.

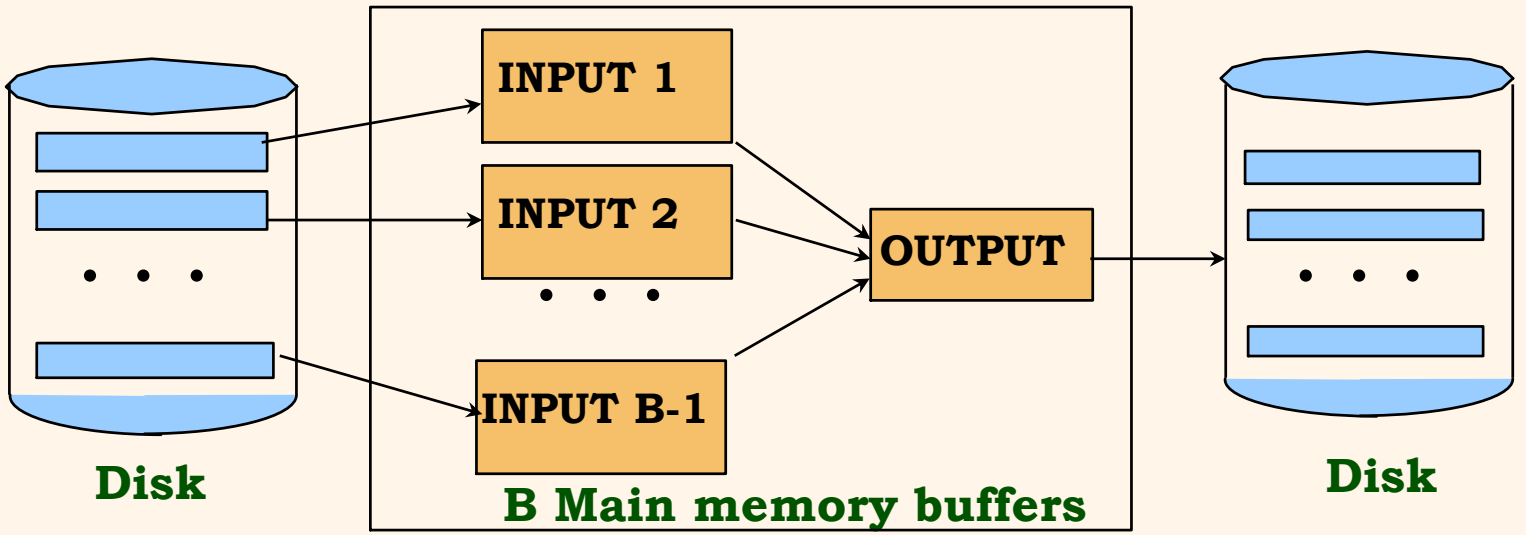
External Merge Sort (3 Buffers)



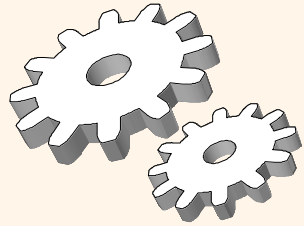
- Number of pages in file = N
- Each pass, read + write each page in file
  - Two IOs per page, per pass
  - Total I/O cost per pass: $2N$
- Number of passes = $\lceil \log_2 N \rceil$
- Total I/O cost = $2N\lceil \log_2 N \rceil$

External Sort (B Buffers)

- Sort a file of *N* pages given *B* buffer pages, *B* > 3

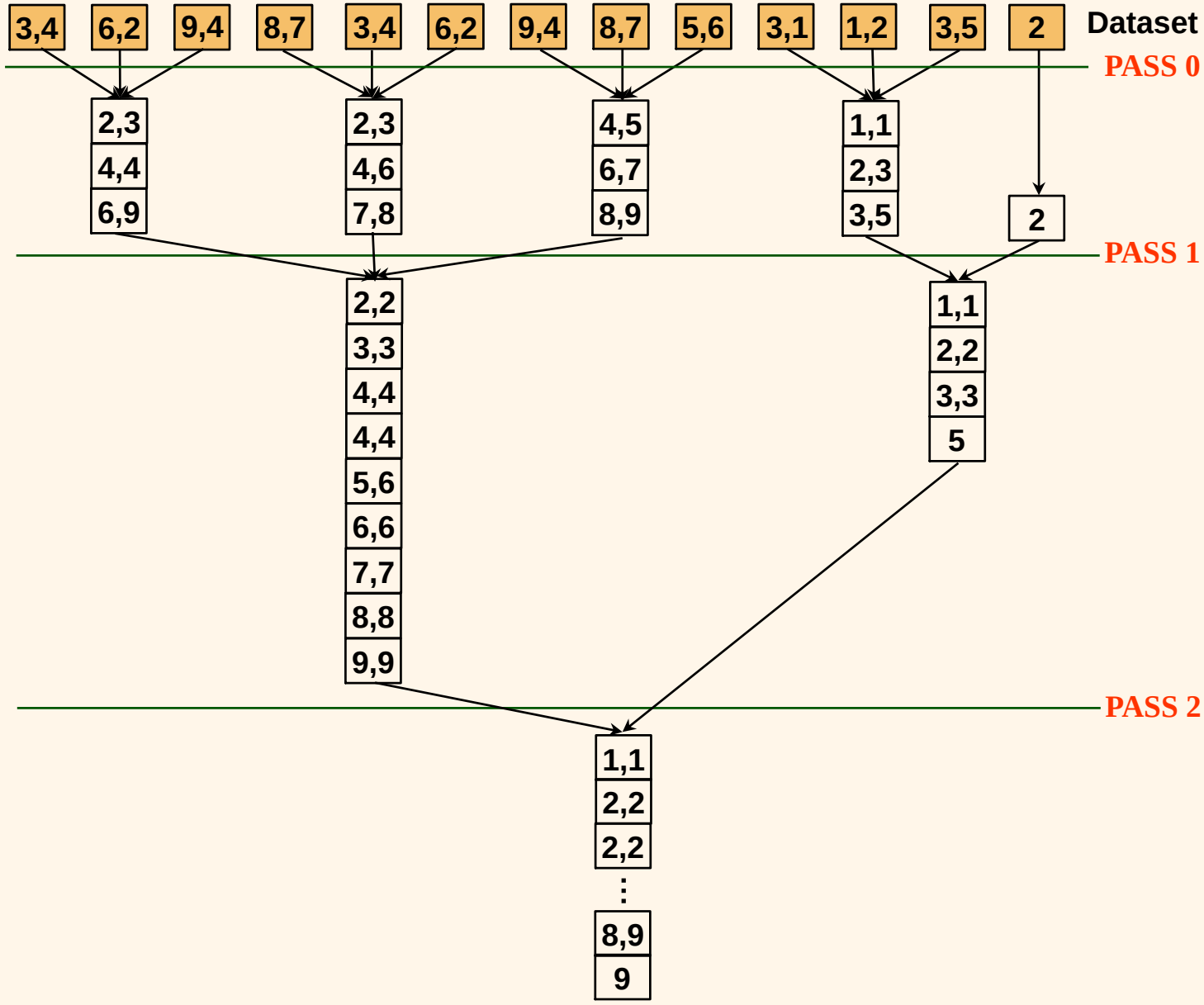


External Sort (B Buffers)

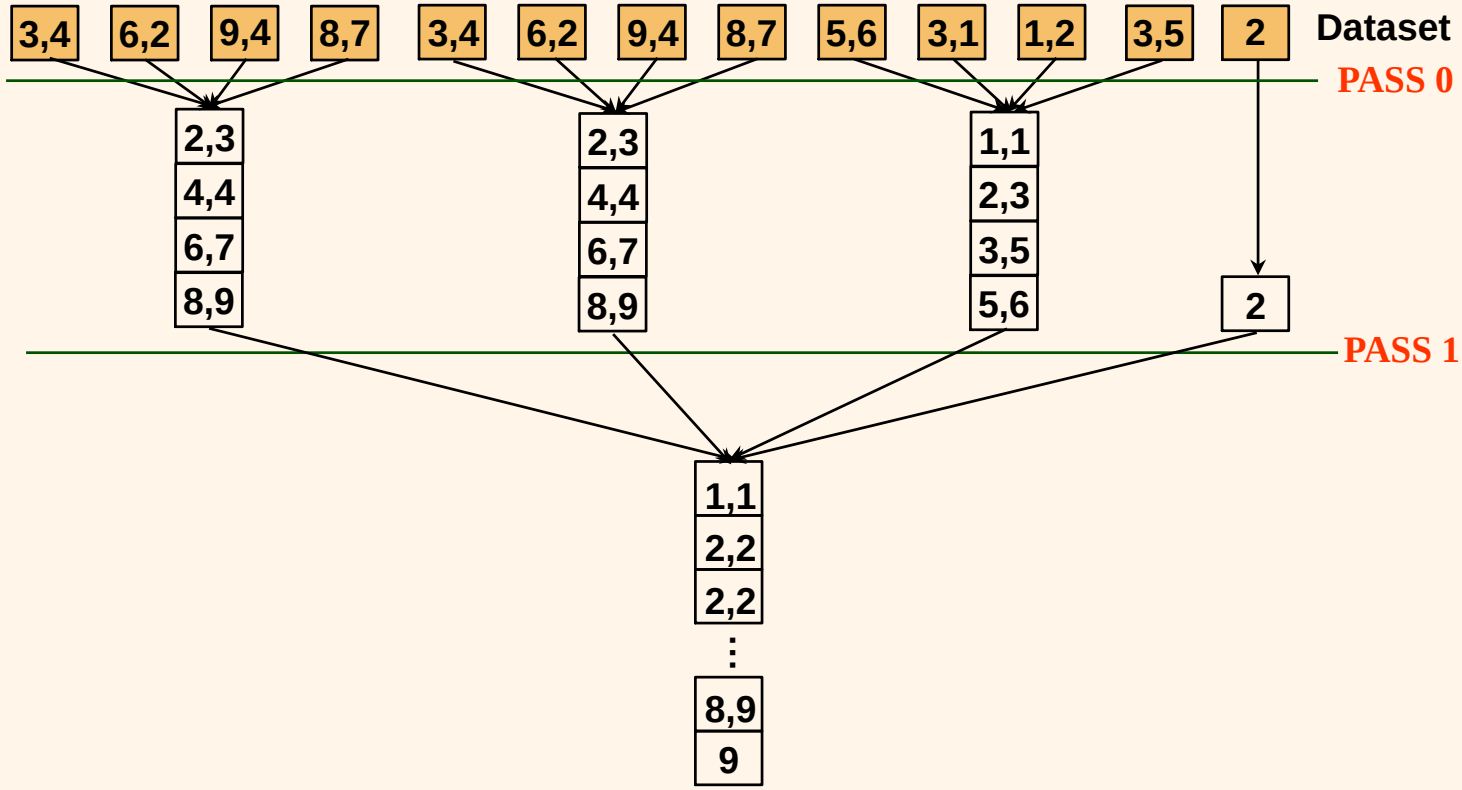


- Pass 0
  - Read in *B-1* pages of file
  - Produce sorted **runs of *B-1* pages** each
- Pass 1, 2, 3, …
  - In each pass
    - Read in *B-1* sorted runs, merge them
    - Use remaining buffer page for output

External Merge Sort (4 Buffers)



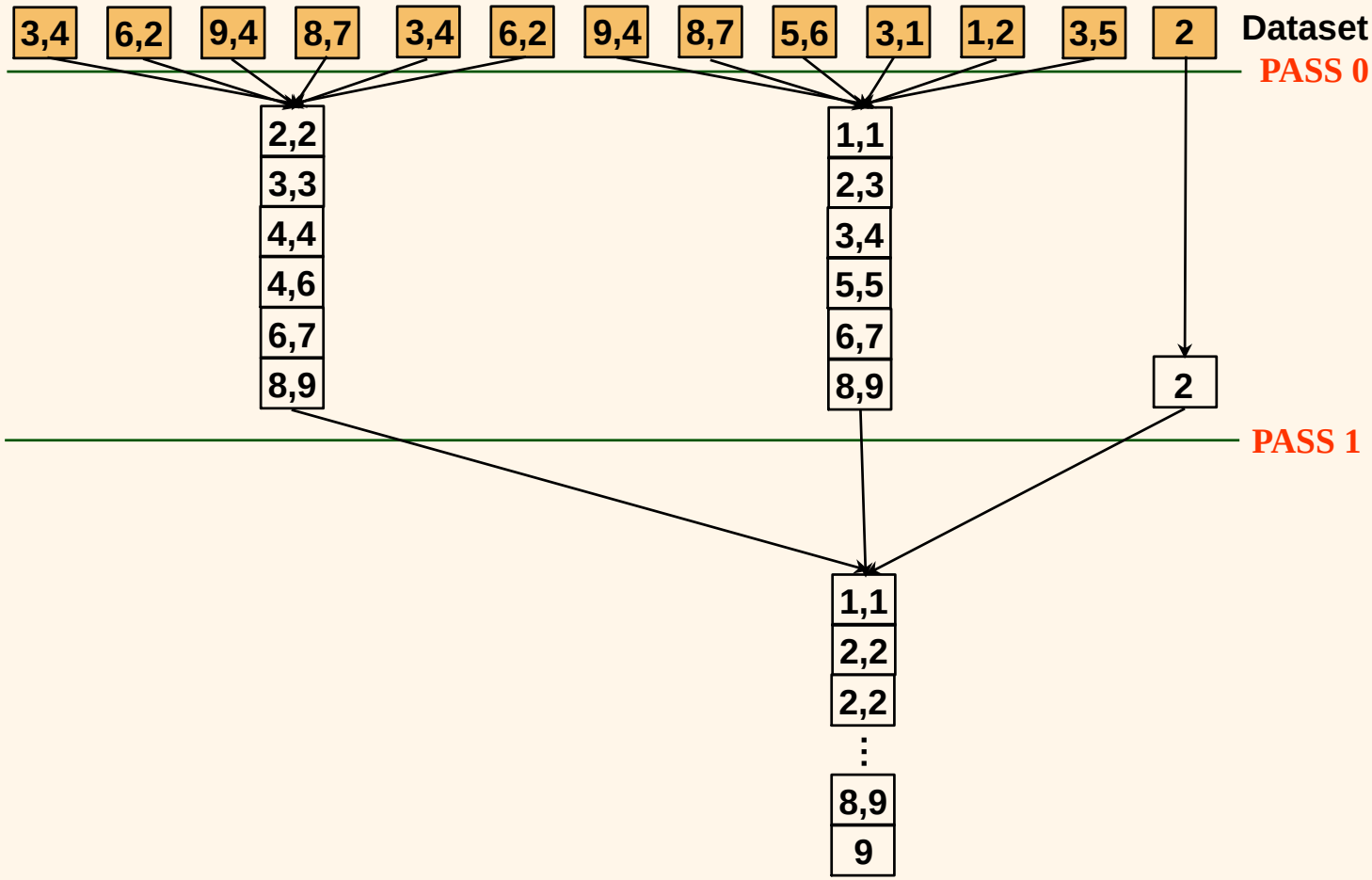
External Merge Sort (5 Buffers)



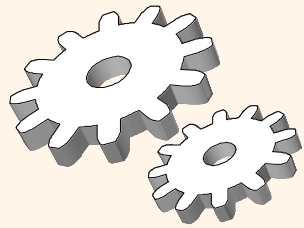
4-page runs

Done

External Merge Sort (7 Buffers)

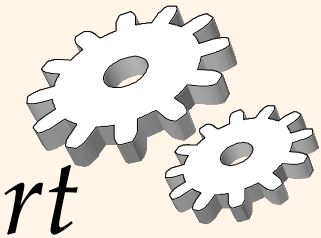


External Merge Sort (B Buffers)



- Number of pages in file = N
- Each pass, read + write each page in file
  - Two IOs per page, per pass
  - Total I/O cost per pass: $2N$
- Number of passes = $\lceil \log_{B-1} N \rceil$
- Total I/O cost = $2N\lceil \log_{B-1} N \rceil$

Number of Passes of External Sort



N	B=3	B=5	B=9	B=17	B=129	B=257
100	7	4	3	2	1	1
1,000	10	5	4	3	2	2
10,000	13	7	5	4	2	2
100,000	17	9	6	5	3	3
1,000,000	20	10	7	5	3	3
10,000,000	23	12	8	6	4	3
100,000,000	26	14	9	7	4	4
1,000,000,000	30	15	10	8	5	4