External sorting

Hans Philippi

March 27, 2025

External sorting

- Sorting algorithms have been extensively studied
- In most cases, the focus is on internal sorting
- We will deal with the situation that the amount of data exceeds internal memory capacity
- Traffic between internal and external memory is done by block sized memory buffers
- Typical block sizes are 4 to 64 kB
- We will show a variant of merge sort
- The algorithm consists of two phases
- Phase 1: split the input into several ordered files (sublists), each fitting in main memory
- Phase 2: merge these sublists into the final resulting file

External merge sort: toy example

- Each block contains two data items
- Main memory has room for four blocks (and if necessary a buffer window) [1]
- Initial situation: a file of sixteen (consecutive) blocks

13	27	9	33	5	29	7	83
76	39	44	56	47	24	91	88
57	36	3	41	33	81	19	6
74	1	68	18	92	64	21	27

[1] Please do not feel confused by the unrealistic small figures of this example.

External merge sort: toy example

13	27	9	33	5	29	7	83
76	39	44	56	47	24	91	88
57	36	3	41	33	81	19	6
74	1	68	18	92	64	21	27

- Phase 1: repeatedly load chunks of four blocks into main memory to create ordered sublists, applying a main memory sorting algorithm

5	7	9	13	27	29	33	83
24	39	44	47	56	76	88	91
3	6	19	33	36	41	57	81
1	18	21	27	64	68	74	92

External merge sort: toy example

- Phase 2: load the leading block from each sublist into main memory

5	7	9	13	27	29	33	83
24	39	44	47	56	76	88	91
3	6	19	33	36	41	57	81
1	18	21	27	64	68	74	92

5	7	24	39	3	6	1	18

External merge sort: toy example

- Write the smallest values in a ordered way to output
- If necessary, load the next block from the corresponding sublist

5	7	9	13	27	29	33	83
24	39	44	47	56	76	88	91
3	6	19	33	36	41	57	81
1	18	21	27	64	68	74	92

__	7	24	39	19	33	__	18

1	3	5	6

External merge sort: toy example

- And repeat ...

5	7	9	13	27	29	33	83
24	39	44	47	56	76	88	91
3	6	19	33	36	41	57	81
1	18	21	27	64	68	74	92

9	13	24	39	19	33	__	18

1	3	5	6	7	

External merge sort: toy example

- And repeat ...

5	7	9	13	27	29	33	83
24	39	44	47	56	76	88	91
3	6	19	33	36	41	57	81
1	18	21	27	64	68	74	92

__	13	24	39	19	33	__	18

1	3	5	6	7	9

External merge sort: toy example

- And repeat ...

5	7	9	13	27	29	33	83
24	39	44	47	56	76	88	91
3	6	19	33	36	41	57	81
1	18	21	27	64	68	74	92

27	29	24	39	19	33	21	27

1	3	5	6	7	9	13	18

- ... until all sublists are empty

External merge sort: analysis

- Cost of phase 1, scanning input file : $B(R)$
- Cost of phase 1, writing ordered sublists : $B(R)$
- Cost of merge scan: $B(R)$
- Cost of writing result: $B(R)$
- Total cost: $4B(R)$
- Two phase merge sort algorithms are applicable as long as $B(R) < M^2$
- M^2 is quite a lot