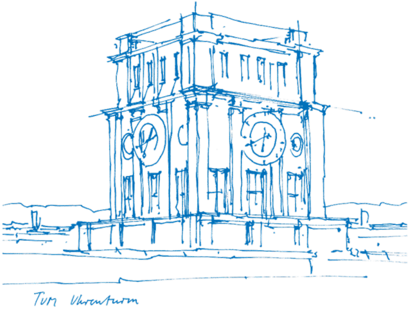
Database Systems on Modern CPU Architectures

Philipp Fent, André Kohn
Database Systems on Modern CPU Architectures
April 23, 2019



External Sort

Problem:

- Sorting an arbitrary amount of data, stored on disk

External Sort

Problem:

- Sorting an arbitrary amount of data, stored on disk
- Accessing disk is slow – so we do not want to access each value individually

External Sort

Problem:

- Sorting an arbitrary amount of data, stored on disk
- Accessing disk is slow – so we do not want to access each value individually
- Sorting in main memory is fast

External Sort

Problem:

- Sorting an arbitrary amount of data, stored on disk
- Accessing disk is slow – so we do not want to access each value individually
- Sorting in main memory is fast – but main memory size is limited

External Sort

Problem:

- Sorting an arbitrary amount of data, stored on disk
- Accessing disk is slow – so we do not want to access each value individually
- Sorting in main memory is fast – but main memory size is limited

Solution:

- Load pieces (called “runs”) of the data into main memory

External Sort

Problem:

- Sorting an arbitrary amount of data, stored on disk
- Accessing disk is slow – so we do not want to access each value individually
- Sorting in main memory is fast – but main memory size is limited

Solution:

- Load pieces (called “runs”) of the data into main memory
- and sort them

External Sort

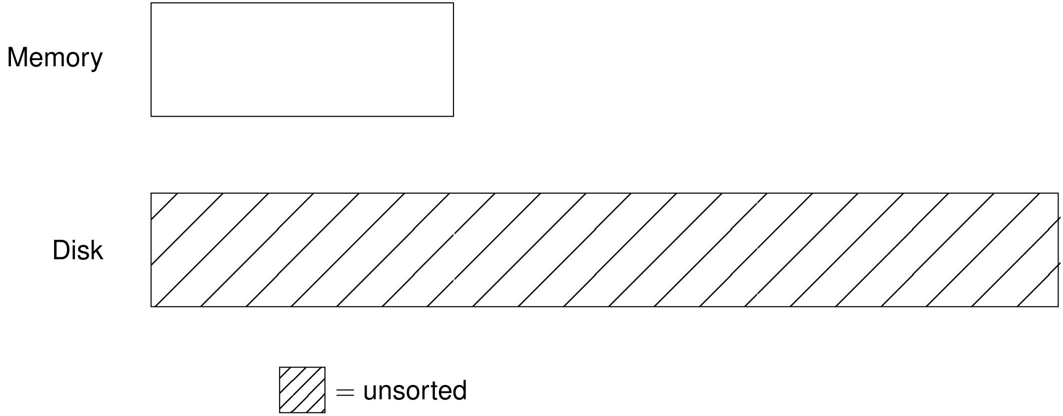
Problem:

- Sorting an arbitrary amount of data, stored on disk
- Accessing disk is slow – so we do not want to access each value individually
- Sorting in main memory is fast – but main memory size is limited

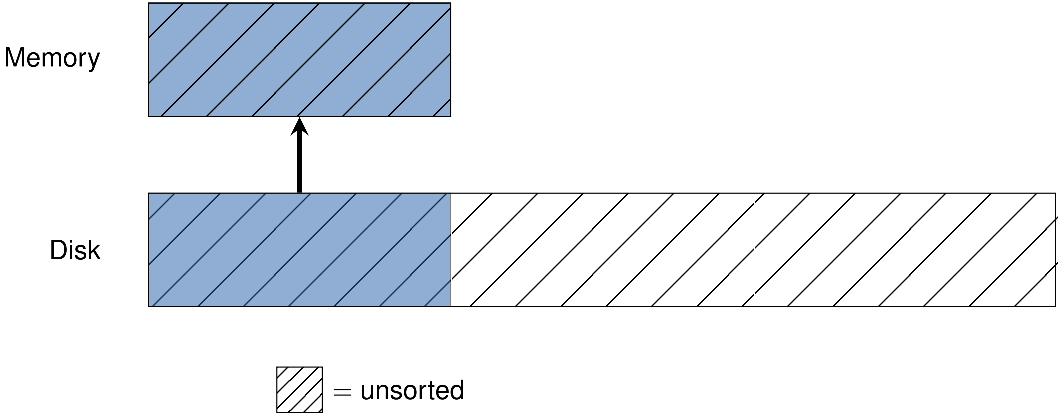
Solution:

- Load pieces (called “runs”) of the data into main memory
- and sort them
- With m values fitting into main memory and d values that should be sorted: $k = \lceil \frac{d}{m} \rceil$ sorted runs

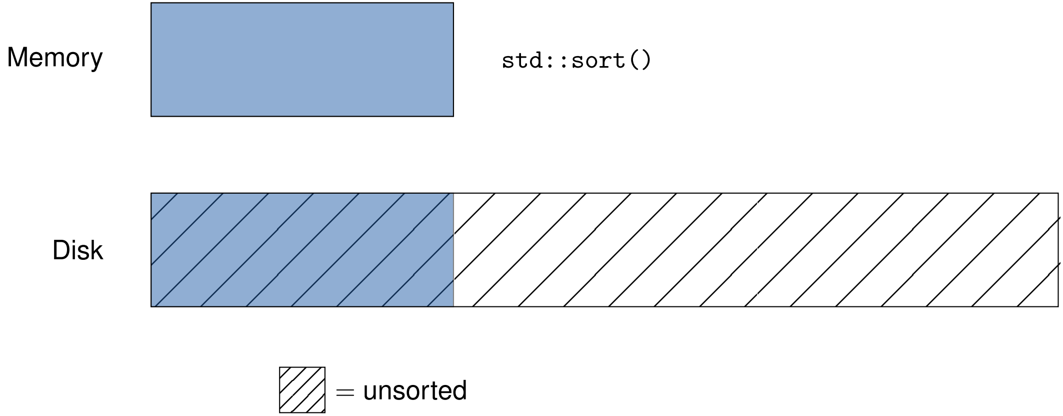
Sort k runs



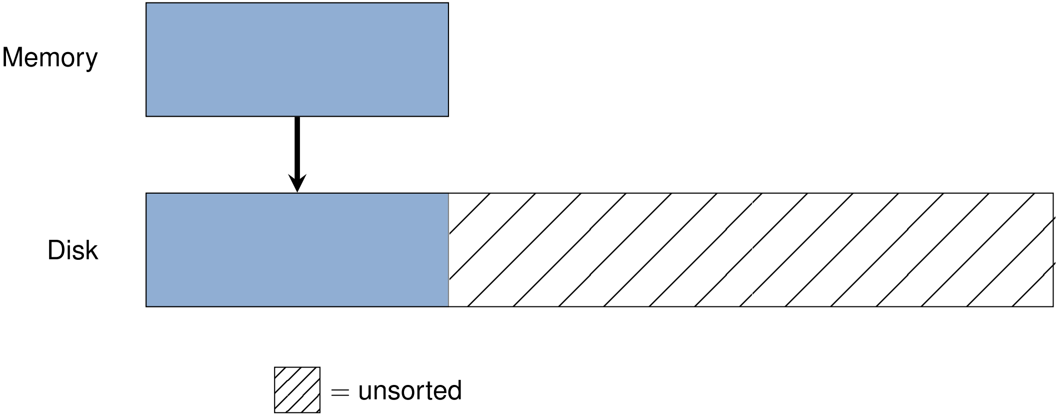
Sort k runs



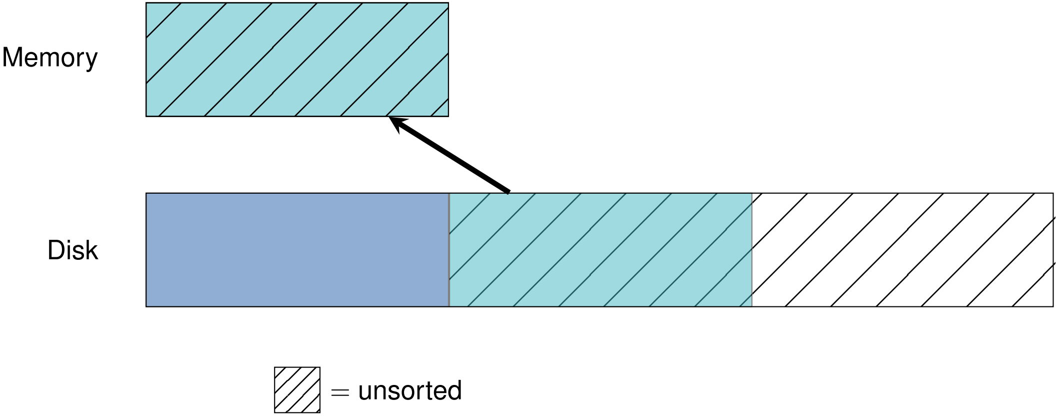
Sort k runs



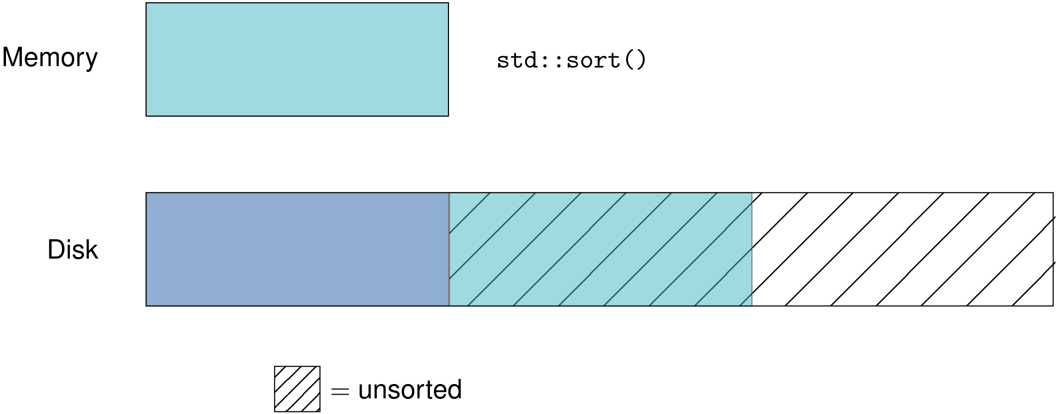
Sort k runs



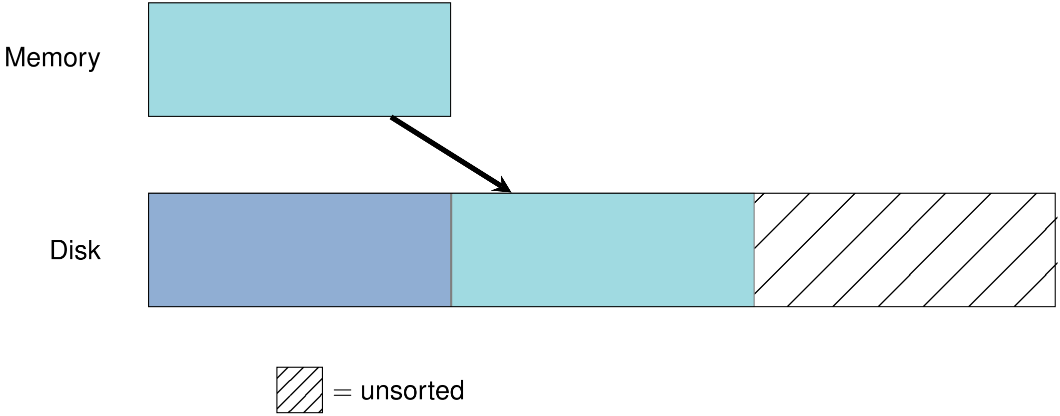
Sort k runs



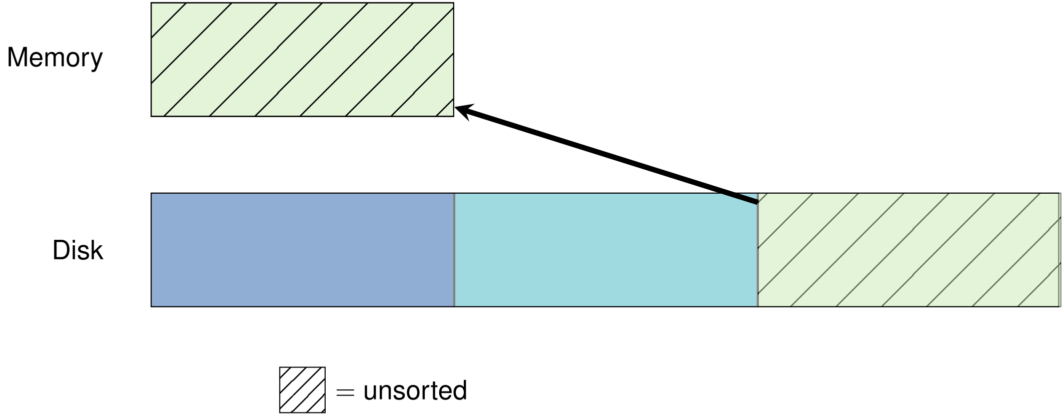
Sort k runs



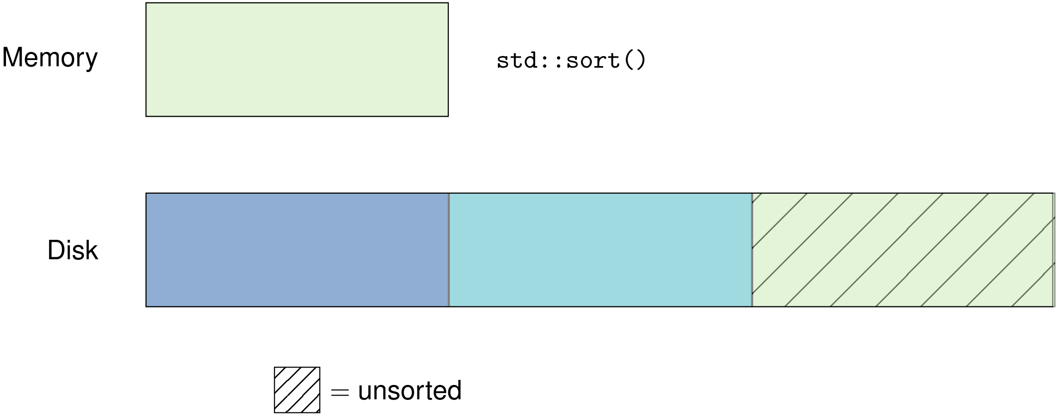
Sort *k* runs



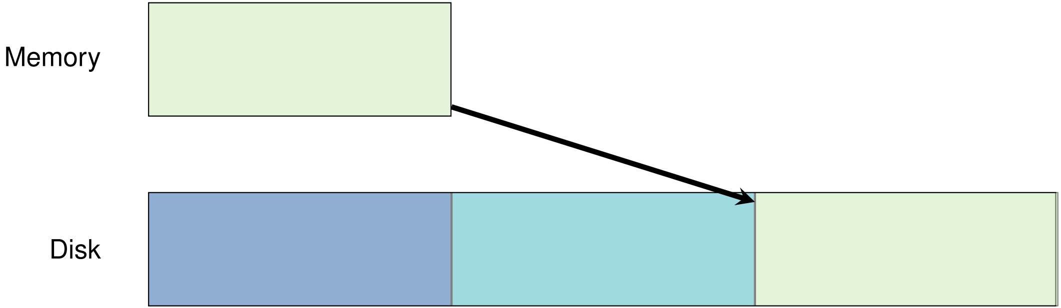
Sort k runs



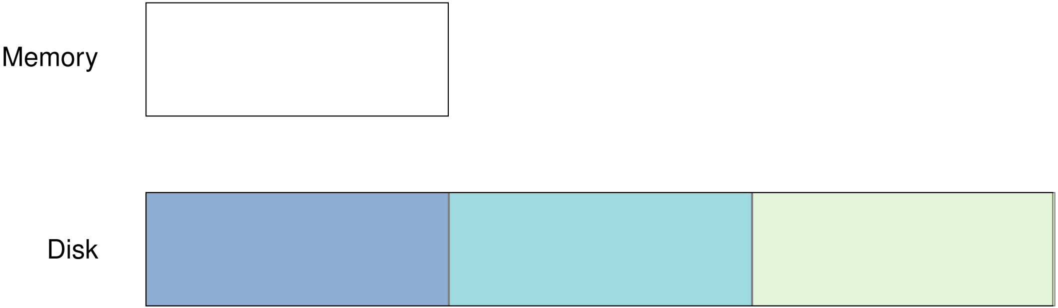
Sort k runs



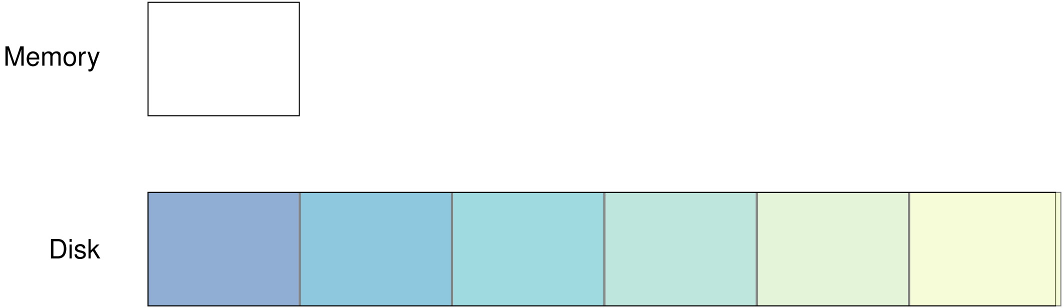
Sort k runs



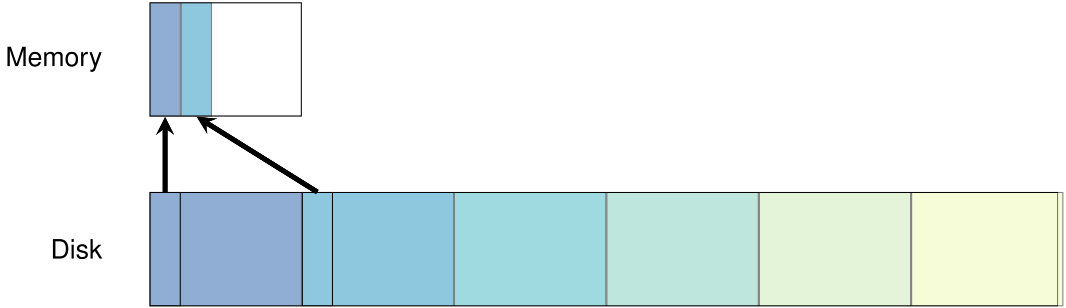
Sort k runs



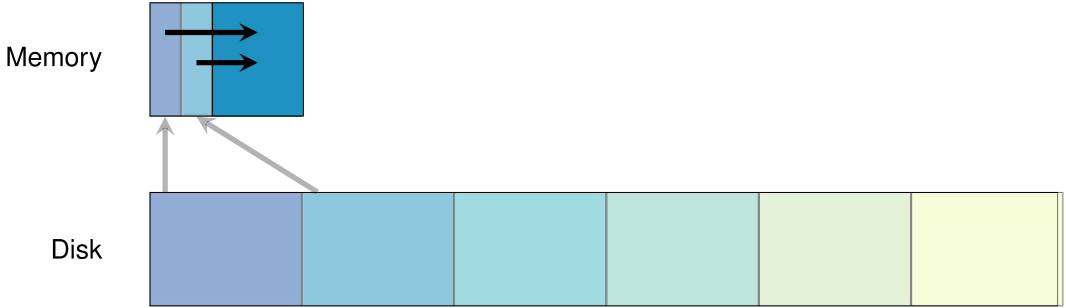
Merge



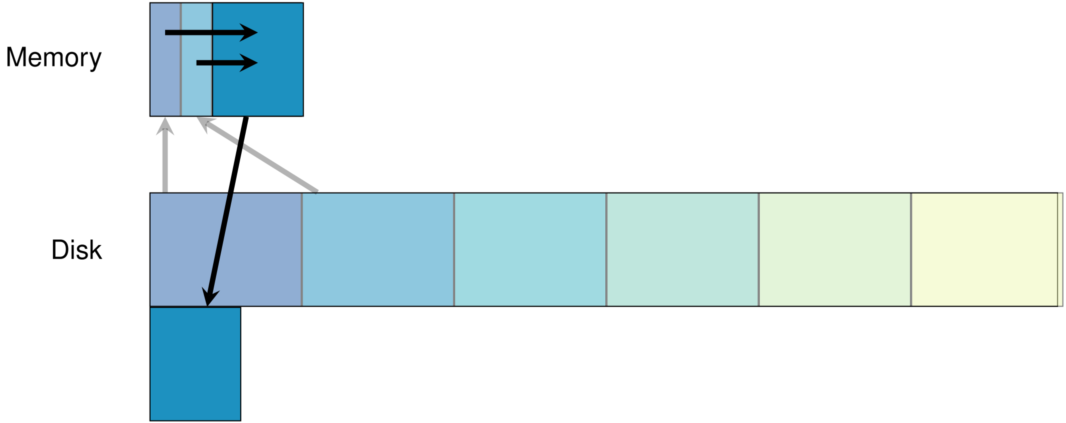
Merge



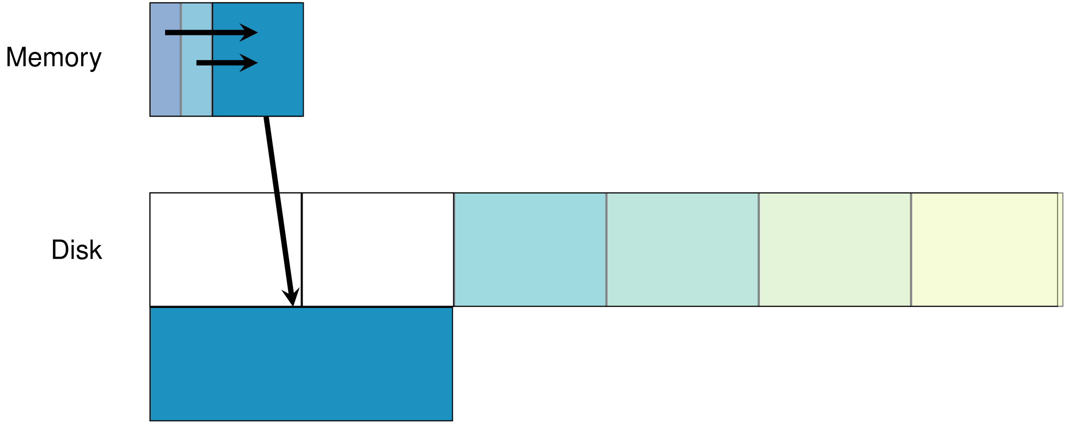
Merge



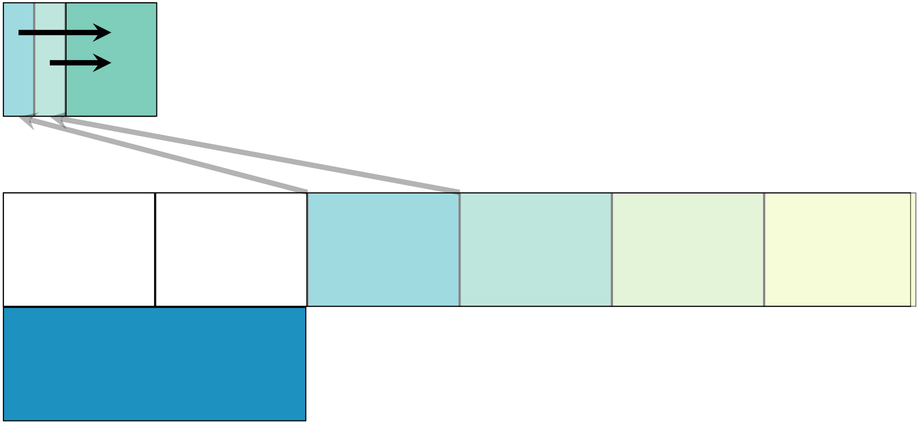
Merge



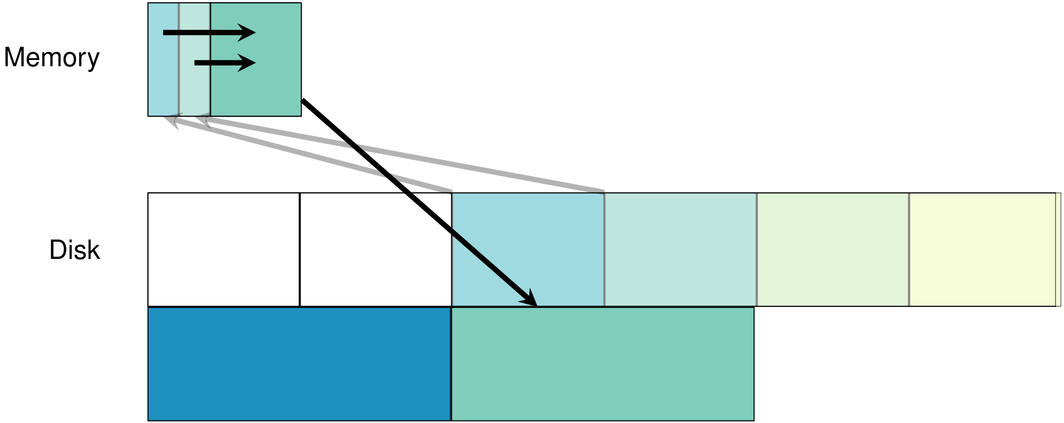
Merge



Merge

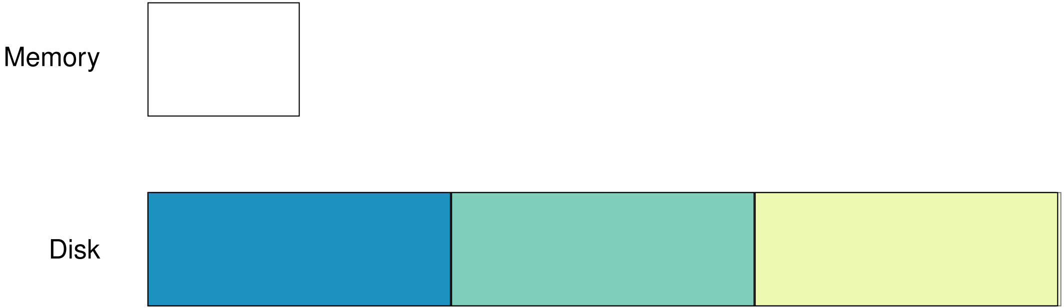


Merge



Merge

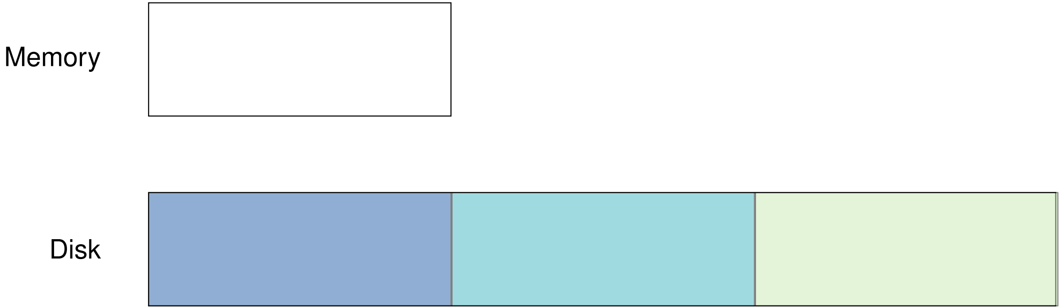
Merge



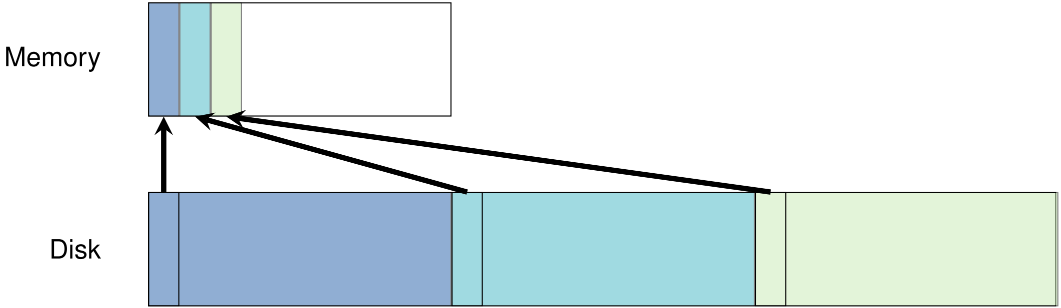
Merge

- Produces sorted result after $\log k$ phases
- Each merge phase reads the whole input data from disk
- Still expensive

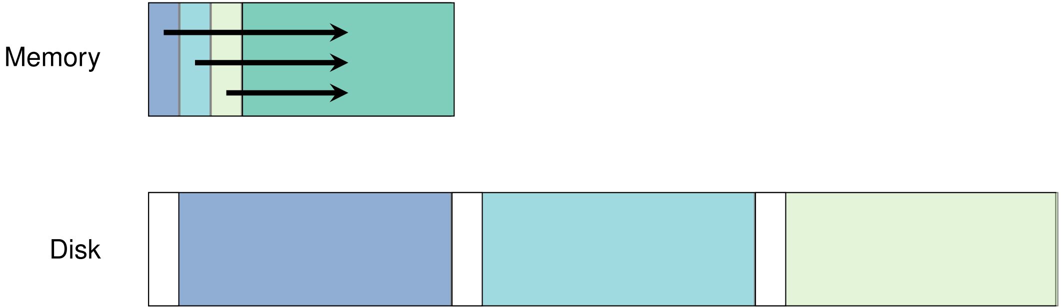
k-way merge



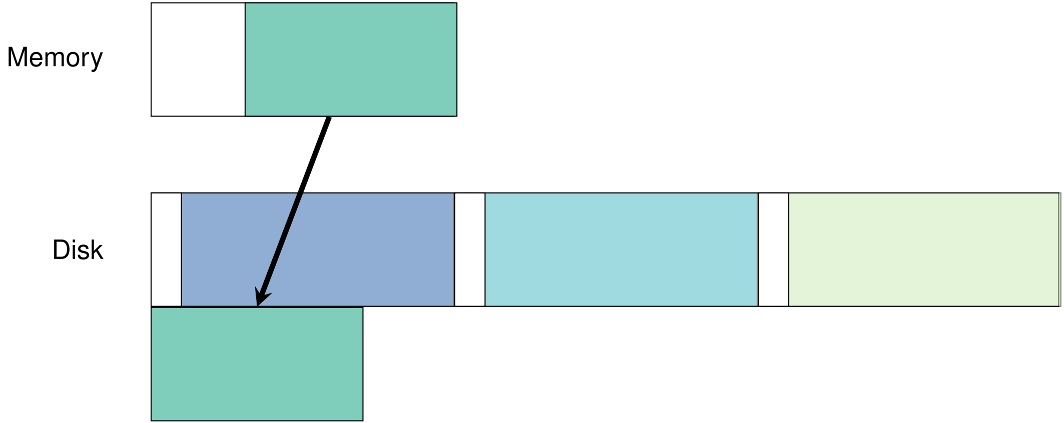
k-way merge



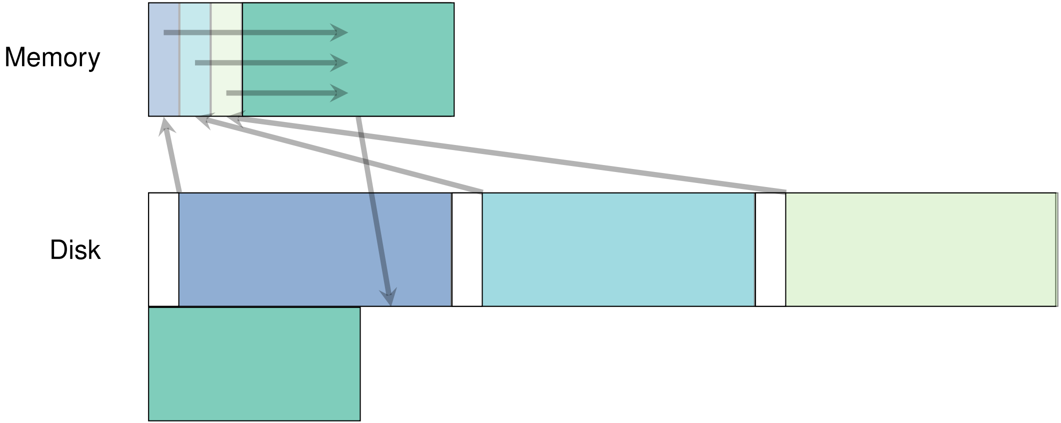
k-way merge



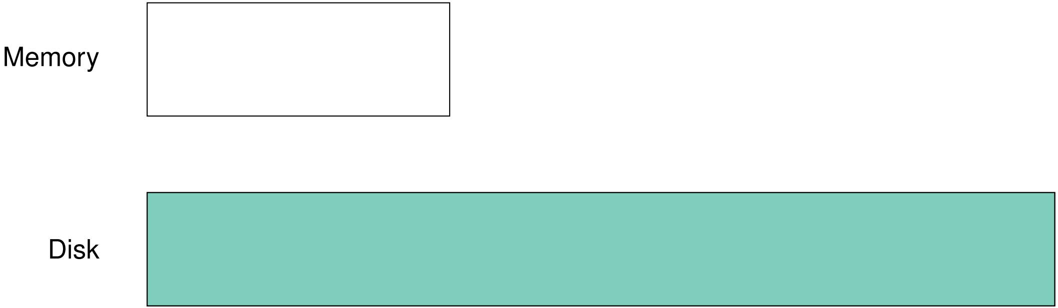
k-way merge



k-way merge



k-way merge



Longer runs

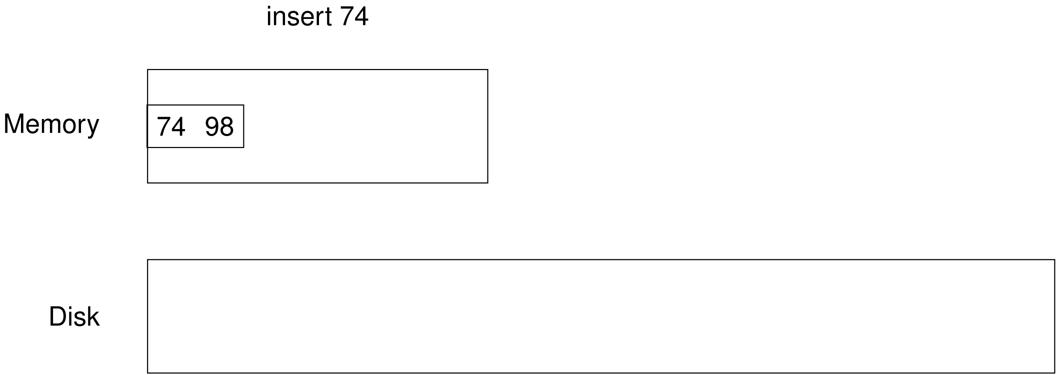
- Minimal disk reads
- *k*-way merge might not fit memory
- Fall back to regular merge
- Produce longer runs with replacement selection

Replacement selection

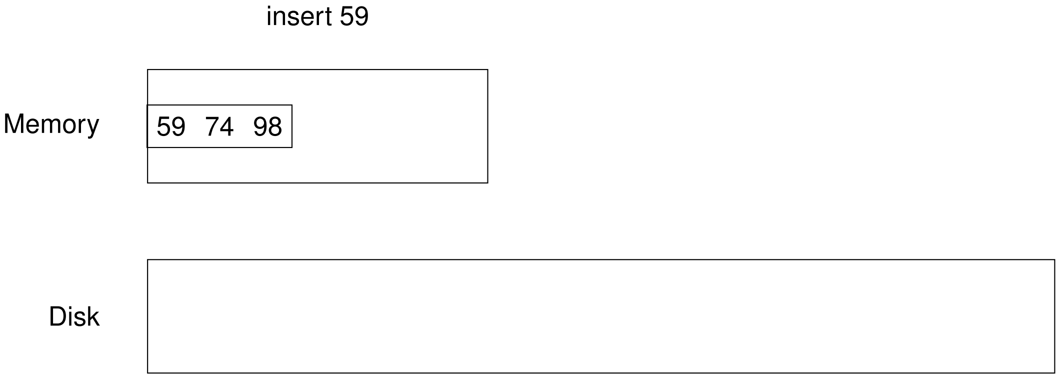
Replacement selection

insert 98

Replacement selection



Replacement selection



Replacement selection

insert 76

Memory	59 74 76 98
Disk	

Replacement selection

insert 38

Memory

38 59 74 76 98

Disk

Replacement selection

insert 66

Memory	38 59 66 74 76 98
Disk	

Replacement selection

insert 10

Memory

10	38	59	66	74	76	98

Disk

Replacement selection

insert 34

Memory

10 38 59 66 74 76 98

Disk

Replacement selection

Memory	34 38 59 66 74 76 98
Disk	10

Replacement selection

insert 47

Memory

34 38 59 66 74 76 98

Disk

10

Replacement selection

Memory	38 47 59 66 74 76 98
Disk	10 34

Replacement selection

insert 53

Memory

38	47	59	66	74	76	98

Disk

10	34

Replacement selection

Memory

47 53 59 66 74 76 98

Disk

10 34 38

Replacement selection

insert 86

Memory

47	53	59	66	74	76	98

Disk

10	34	38

Replacement selection

Memory

53	59	66	74	76	86	98

Disk

10	34	38	47

Replacement selection

insert 84

Memory

53 59 66 74 76 86 98

Disk

10 34 38 47

Replacement selection

Memory	59 66 74 76 84 86 98
Disk	10 34 38 47 53

Replacement selection

insert 19

Memory	59 66 74 76 84 86 98
Disk	10 34 38 47 53

Replacement selection

Memory	66	74	76	84	86	98	19

Disk	10	34	38	47	53	59

Replacement selection

insert 30

Memory	66	74	76	84	86	98	19
Disk	10	34	38	47	53	59	

Replacement selection

Memory	74	76	84	86	98	19	30

Disk	10	34	38	47	53	59	66

Replacement selection

insert 63

Memory	74	76	84	86	98	19	30
Disk	10	34	38	47	53	59	66

Replacement selection

Memory | 76 84 86 98 | 19 30 63

Disk | 10 34 38 47 53 59 66 74

Replacement selection

insert 83

Memory

76	84	86	98	19	30	63

Disk

10	34	38	47	53	59	66	74

Replacement selection

Memory

83	84	86	98	19	30	63

Disk

10	34	38	47	53	59	66	74	76

Replacement selection

insert 37

Memory							
	83	84	86	98	19	30	63

Disk									
	10	34	38	47	53	59	66	74	76

Replacement selection

Memory	84	86	98	19	30	37	63			
Disk	10	34	38	47	53	59	66	74	76	83

Replacement selection

insert 65

Memory	84	86	98	19	30	37	63

Disk	10	34	38	47	53	59	66	74	76	83

Replacement selection

Memory	86 98	19	30	37	63	65					
Disk	10	34	38	47	53	59	66	74	76	83	84

Replacement selection

insert 29

Memory	86 98	19 30 37 63 65
Disk	10 34 38 47 53 59 66 74 76 83 84	

Replacement selection

Memory							
	98	19	29	30	37	63	65

Disk													
	10	34	38	47	53	59	66	74	76	83	84	86	95

Replacement selection

insert 45

Memory

98	19	29	30	37	63	65

Disk

10	34	38	47	53	59	66	74	76	83	84	86	95

Replacement selection

Memory

19	29	30	37	45	63	65

Disk

10	34	38	47	53	59	66	74	76	83	84	86	95	98