

External Sorting

Frequently DBMS needs to sort data

Often not enough memory to do this in RAM so need to have disk based techniques

When does DBMS SORT DATA

- when users wants data in some order
(Ex) increasing age
- bulk loading tree indexes
- eliminating duplicate copies in a collection of records
- computing joins.

Typical Way to do Sorting Based on a bottom up version of Merge Sort

Recall: 2-way Merge Sort

Suppose we want to sort

Run of length 1

4	3	7	9	2	6	5	8
---	---	---	---	---	---	---	---

Run of length 2

3	4	7	9	2	6	4	5
---	---	---	---	---	---	---	---

pass 1 Merge Runs of length 1 to produce runs of length 2

Each merge $O(n)$ time

3	4	7	9	2	4	5	6
---	---	---	---	---	---	---	---

pass 2 Merge Runs of length 2 to produce runs of length 4

Need to merge $\log_2 n$ times

2	3	4	5	6	7	9
---	---	---	---	---	---	---

pass 3

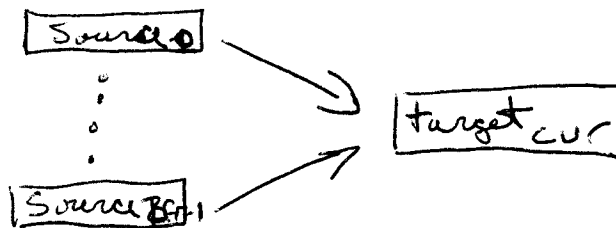
Merge Runs of length 4 to produce runs of length 8

External Merge Sort

Idea: Want to change $O(n \log_2 n)$
 to $O(n \log_{bfr} n)$
 # of blocks \uparrow # of records/block \uparrow

Have $2 \times bfr$ many files
 In a given phase have bfr many source files
 and bfr many target files (Initialized by divided input files into bfr many files)

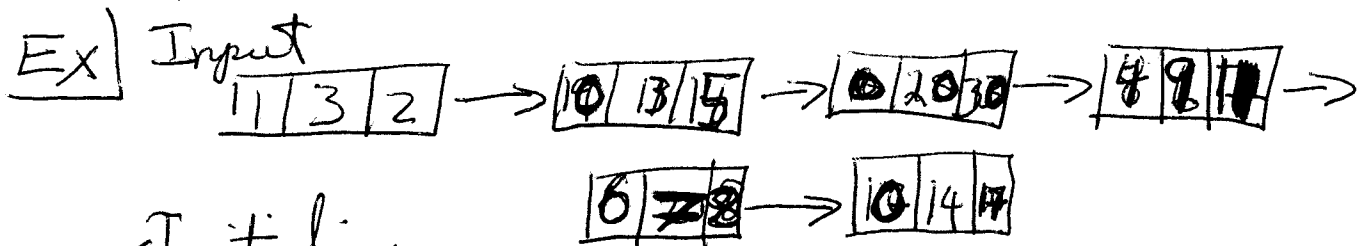
~~$O(n \log_2 n)$~~ We cycle through each target file to merge to in the following way



once have merged together
 number of current blocks in source file
 $runs \times bfr$

many blocks let $cur = (cur + 1 \% bfr)$

At end of a pass roles of source and target files are ~~inter~~ interchanged.



Initialize
 → split into 3 source files sort blocks

Source 1

1	2	3
---	---	---

 →

4	9	11
---	---	----

Source 2

10	13	15
----	----	----

 →

6	7	8
---	---	---

Source

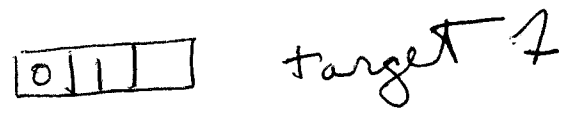
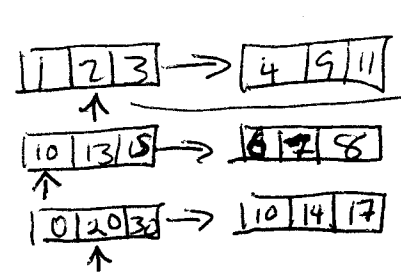
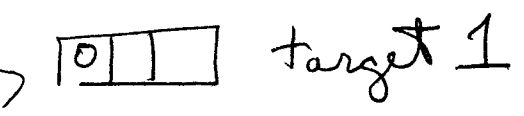
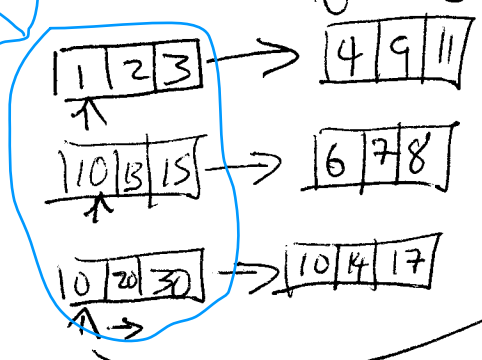
0	20	30
---	----	----

 →

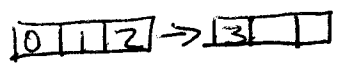
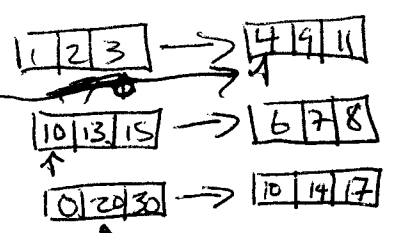
10	14	17
----	----	----

Pass 1
Merge into
Runs of length 3 blocks

Only these
in buffer
Read 1st
block of
each
source
file
compare
1st records

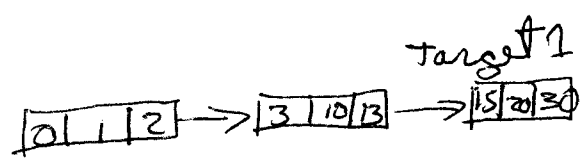
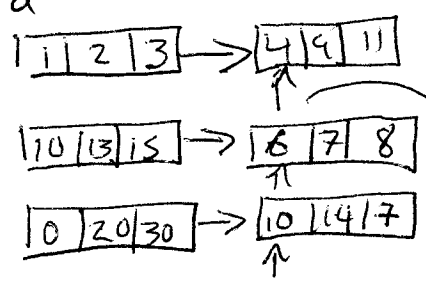


Skip
some
steps

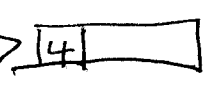


note
initial
run
length 1
since this
5 is at
end of run
won't move till
run is over

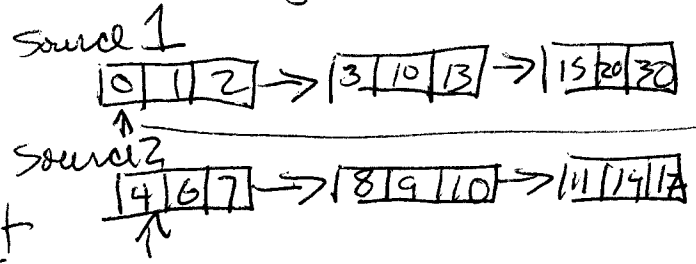
Skip more



target 2



After pass 1 targets become sources



begin next
run using
target 1
target 2

process begins
again
after pass 2
data will be
sorted.

note didn't
have enough
data to produce
3 files