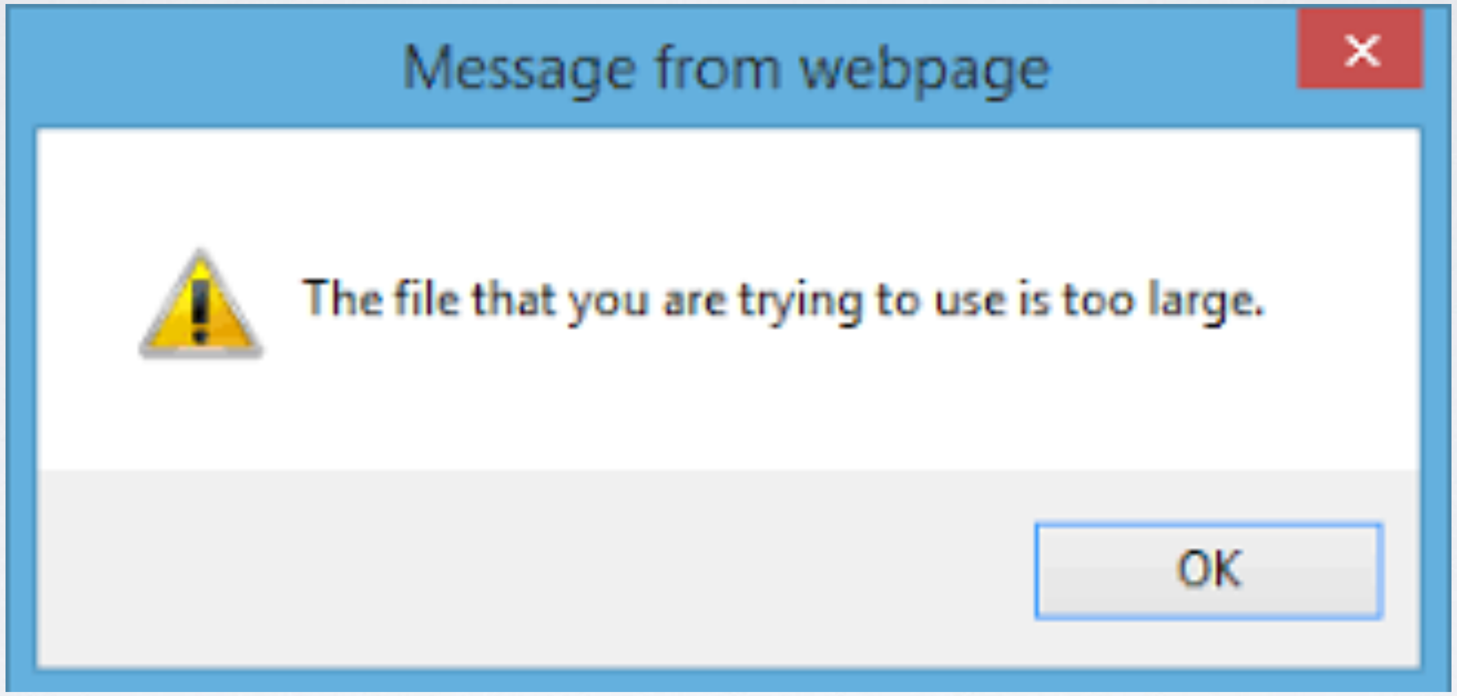
EXTERNAL MEMORY SORTING



John Geissberger Jr.
Rushita Patel

QUESTIONS

- Question #1: External memory sorting algorithm analysis focuses on what aspect?
- Question #2: When was the External Memory Model Proposed?
- Question #3: Name one application where External Memory Sort Used?

JOHN GEISSBERGER JR.

- Knoxville TN.

- Olympia Washington.

JOHN GEISSBERGER JR.

Hometown: Knoxville TN.
Masters student in CS

Data Science.

Hip Hop Music.

Weight lifting.

Tutor.

RUSHITA PATEL

- Knoxville, TN

RUSHITA PATEL

- Hometown : Surat, Gujarat ,India
- Masters Student in CS
- Traveling
- Cooking
- Music depends on mood

OUTLINE

- Overview- Definitions, Applications
- History of External Sorting
- Algorithms
- Applications
- Implementation
- Open Issues
- References
- Discussion

OVERVIEW

Exponential growth of Data.

"Degrees" of Big Data -

One grain of rice = 8 bits = one byte

2 Containerships full of rice = 2^40 bytes = Terabyte

Enough Rice to cover Manhattan = 2^50 = Petabyte

Enough Rice to cover the west coast = 2^60 = Exabyte

Enough rice to fill the pacific ocean = 2^70 = Zettabyte

In 2008 Google was handling 20 Petabytes worth of information per day

Now we can safely assume, by applying Moore's law, that Google is producing atleast 200 Petabyte's of data per day

OVERVIEW

Sorting

- Fundamental routine computers perform

- Common sorting algorithms

Radix Sort, Merge Sort, Counting Sort, Quick Sort

Underlying assumption- RAM model of computation

- All memory is equally expensive to access
- All reasonable instructions take unit time

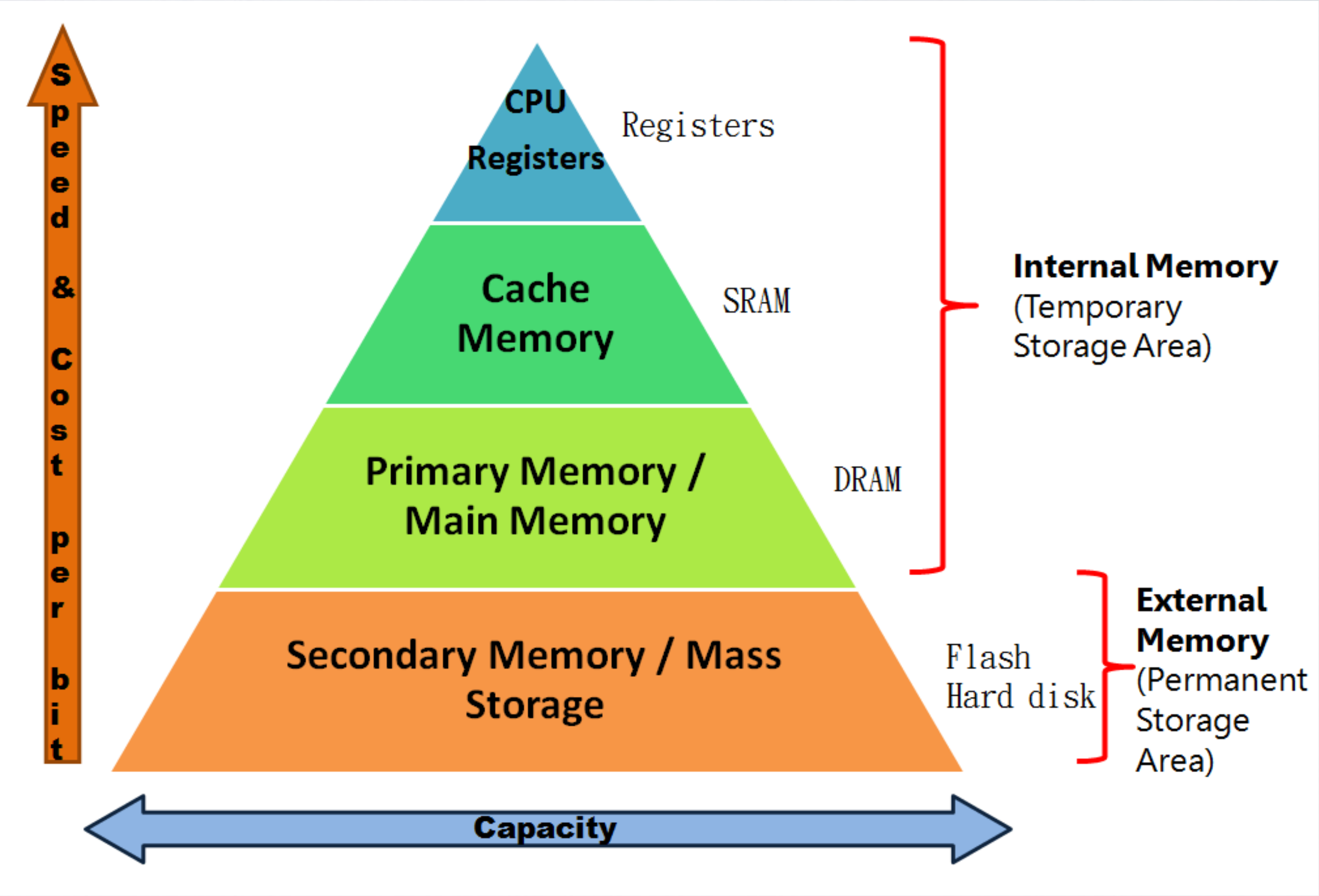
OVERVIEW

External Memory Model and Idealized Cache Model

-Takes into account memory hierarchy

-Not all memory access is equivalent

Access times to disk are typically as much as 100,000-1,000,000 times longer



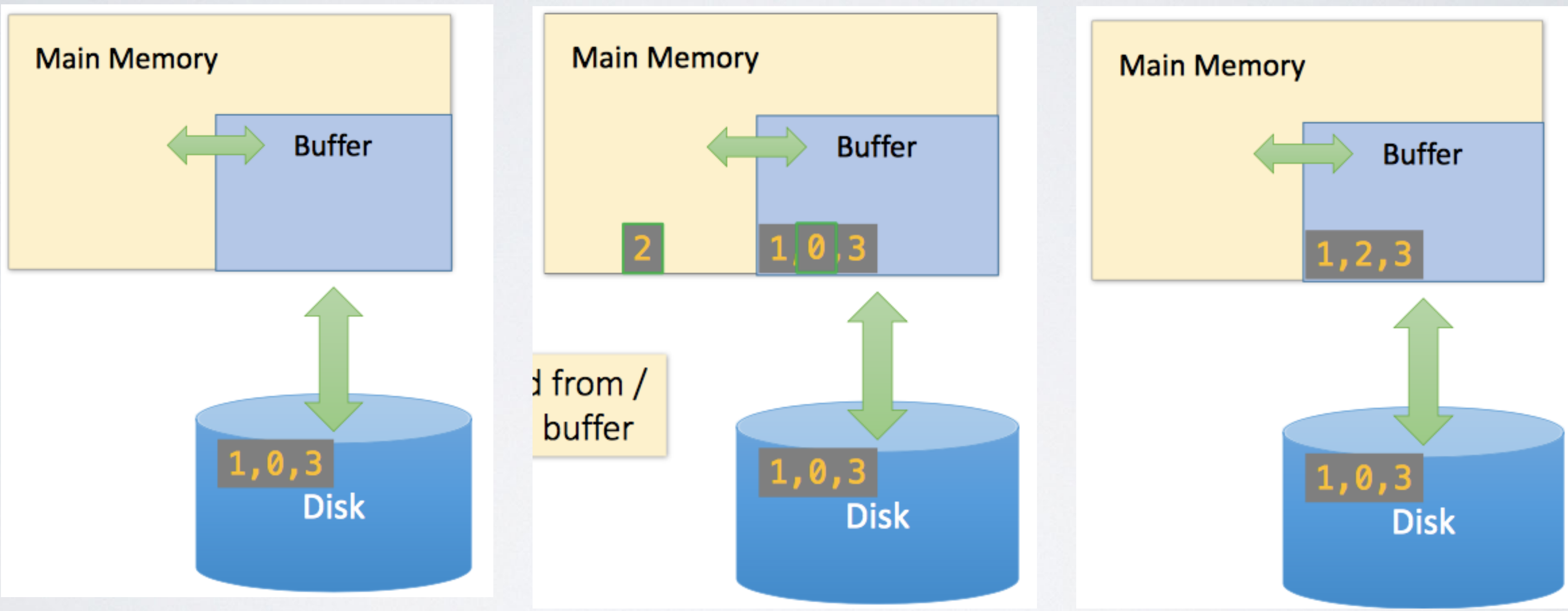
A student living in Baltimore can communicate with their parents by email, internal memory access, within 5 seconds. Or that student could deliver the message in person, which would take about a month if the student averages 20 miles per day.

OVERVIEW

- Algorithm Analysis - The number of memory transfers required.
- Def: Memory transfer operations, which read a block from disk to cache, or write a block from cache to disk.
- Both the cache and disks are divided into blocks.
- A memory transfer will be completed by reading M records at a time, sorting the records internally, and then write the sorted records onto a secondary device. This process is called a Run.
- Components:
  B = Size of disk block.
  M = number of items that fits in memory

OVERVIEW

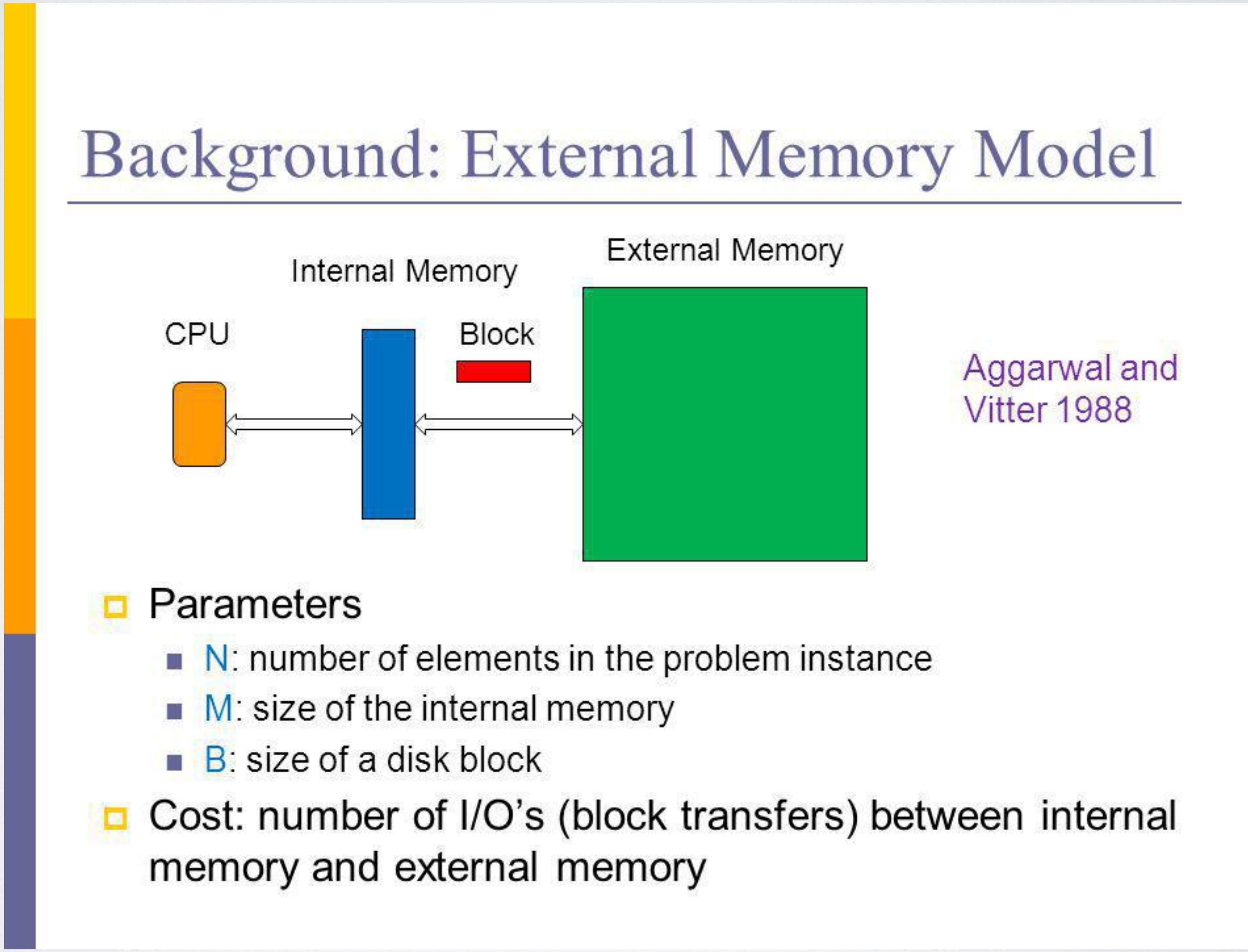
- Def: A <u>buffer</u> is a region of physical memory used to store temporary data.



OVERVIEW

External Memory Model and Idealized Cache Model

- Algorithm Analysis - The number of memory transfers required.



HISTORY

Aggarwal and Vitter

- Introduced External Memory Model in 1988

- Jeffrey Scott Vitter -

P.H.D. - Comp Sci Stanford

Made contributions in-

Huffman coding, image compression, machine learning, databases.

HISTORY

Idealized Cache Model-

Charles E. Leiserson -

Ph.D. - Computer Sci Carnegie Mellon

developed idealized Cache Model

contributed to VLSI theory

Professor in computer science at MIT

co-authored Introduction to Algorithms

ALGORITHMS

Outline:

- Basic example of how External Memory Sort works
- Two-Way Sorting
- B-Way Sorting
- K-Way Merge Sort

ALGORITHMS

Walk through Basic example

How would we merge two large sorted files with limited memory?

To find an element that is no larger than all elements in two lists, one only needs to compare minimum elements from each list.

If:

$$A_1 \leq A_2 \leq \cdots \leq A_N$$
$$B_1 \leq B_2 \leq \cdots \leq B_M$$

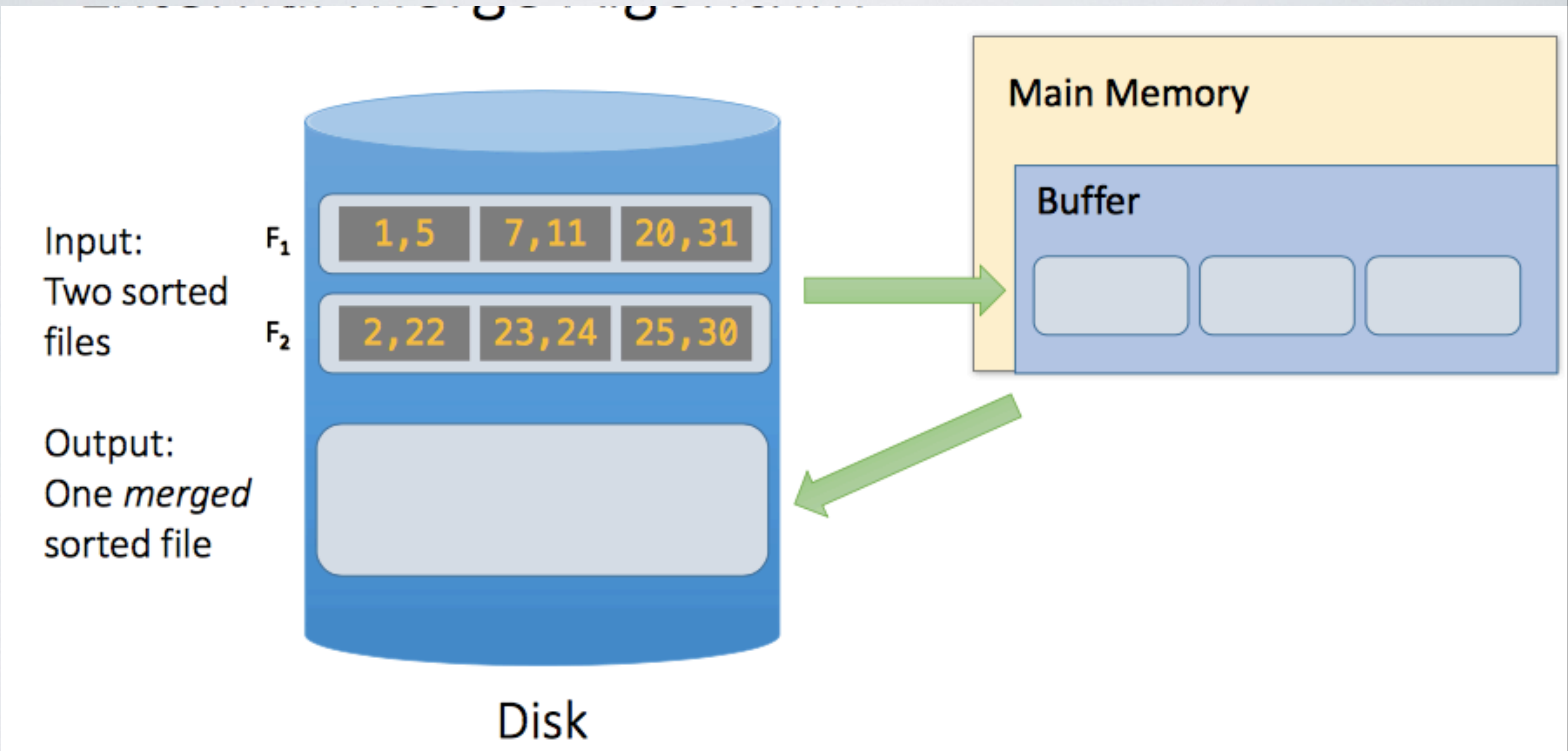
Then:

$$Min(A_1, B_1) \leq A_i$$
$$Min(A_1, B_1) \leq B_j$$

for i=1....N and j=1....M

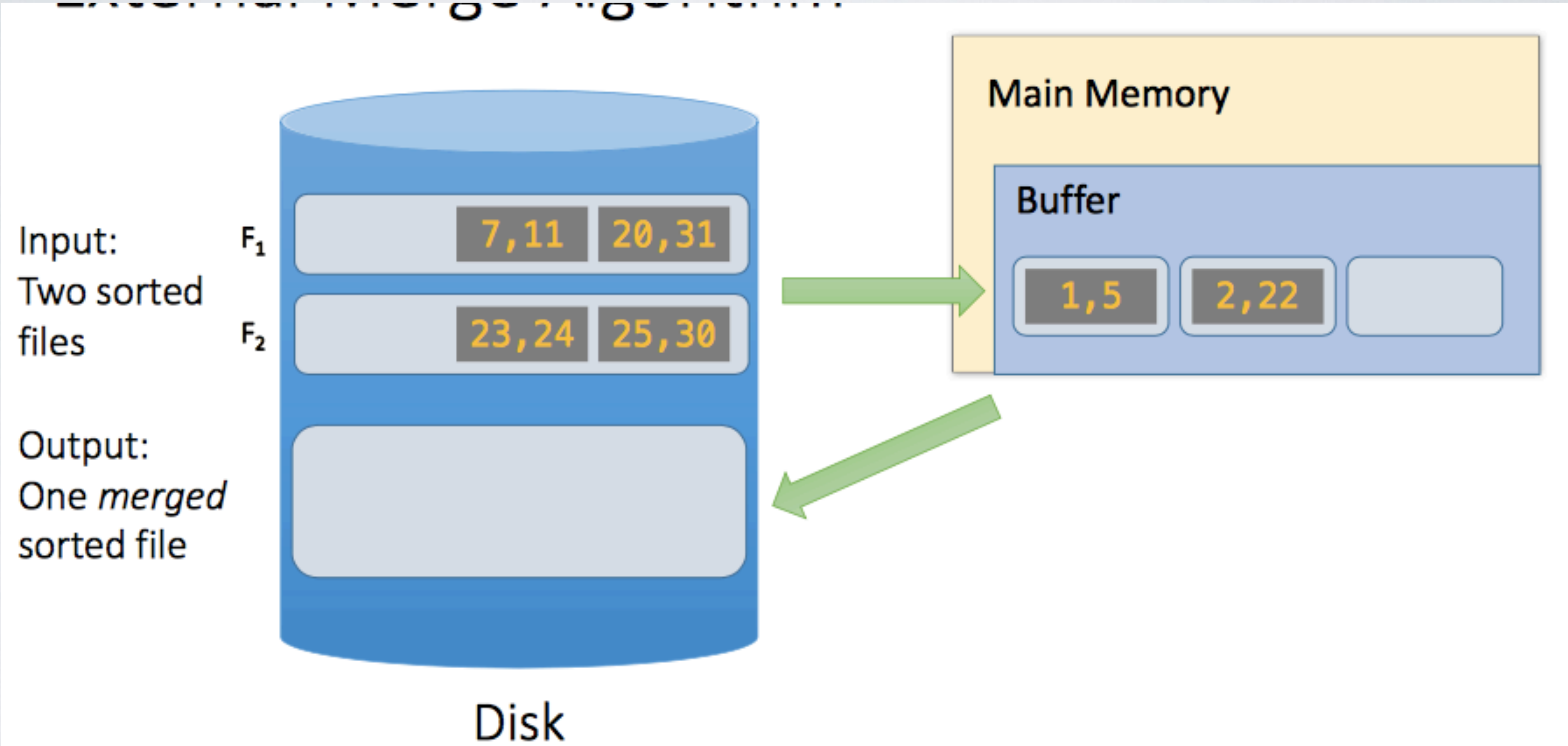
ALGORITHMS

Merge two large sorted files with limited memory



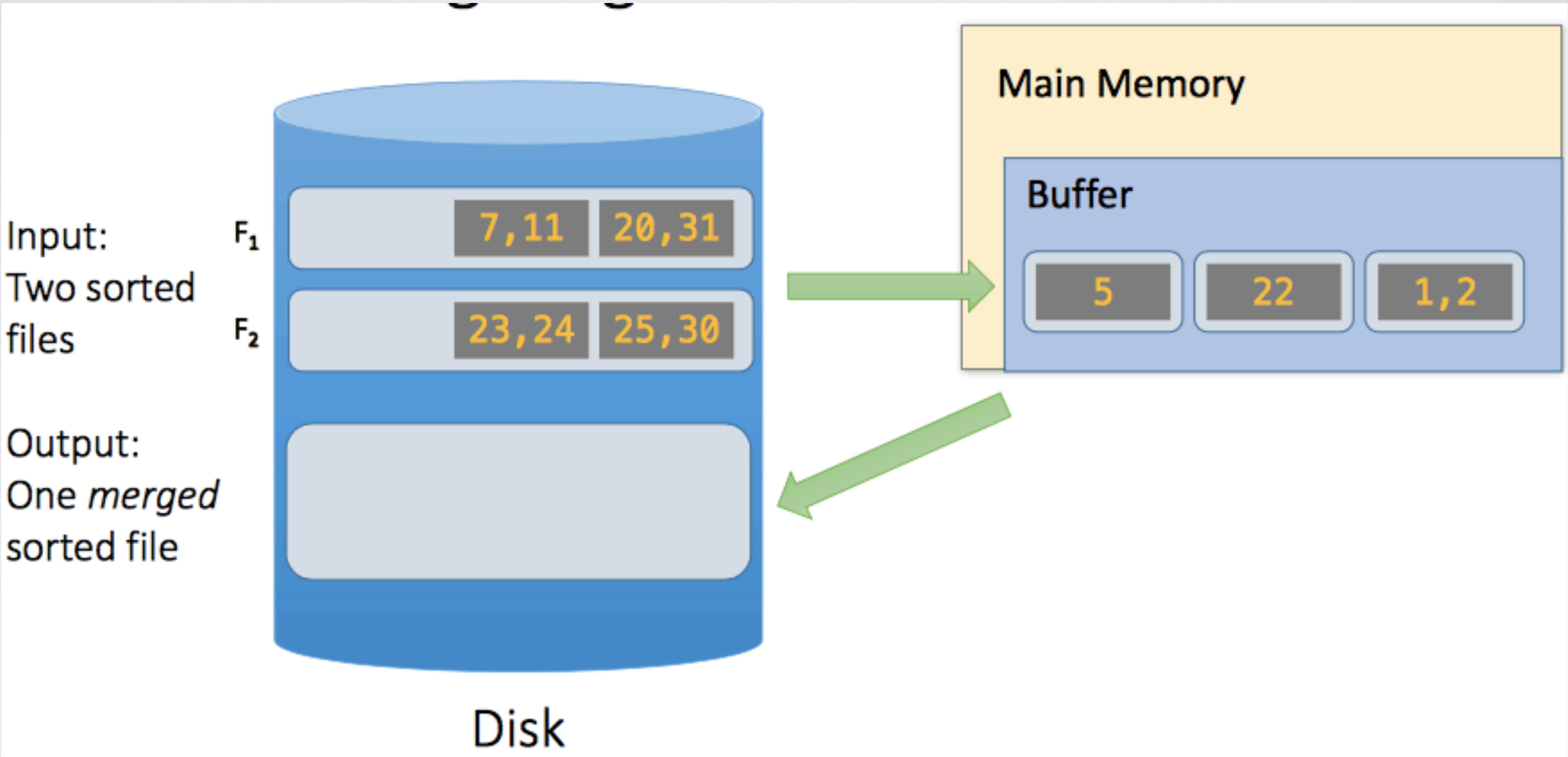
ALGORITHMS

Merge two large sorted files with limited memory-cont



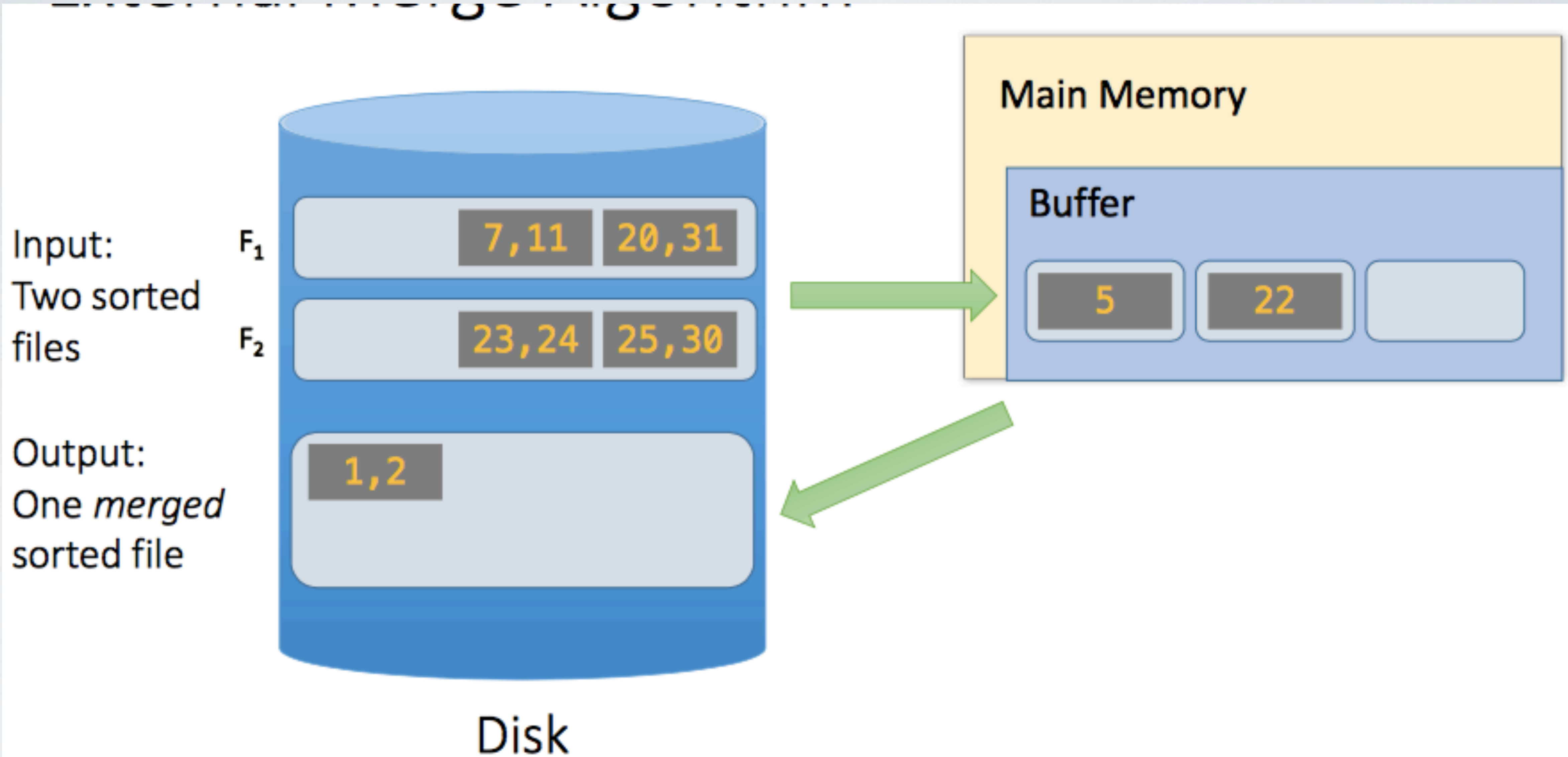
ALGORITHMS

Merge two large sorted files with limited memory-cont



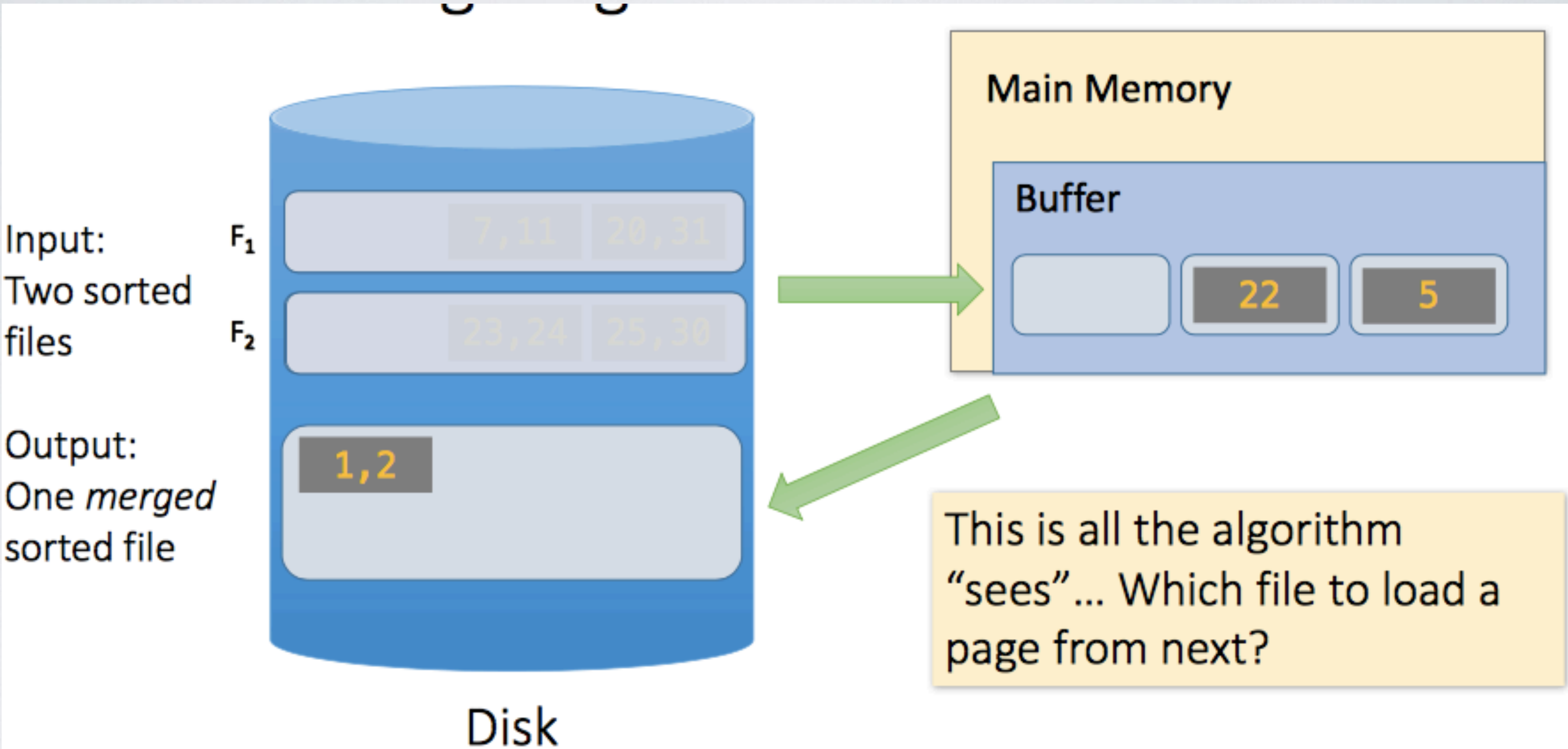
ALGORITHMS

Merge two large sorted files with limited memory-cont



ALGORITHMS

Merge two large sorted files with limited memory-cont

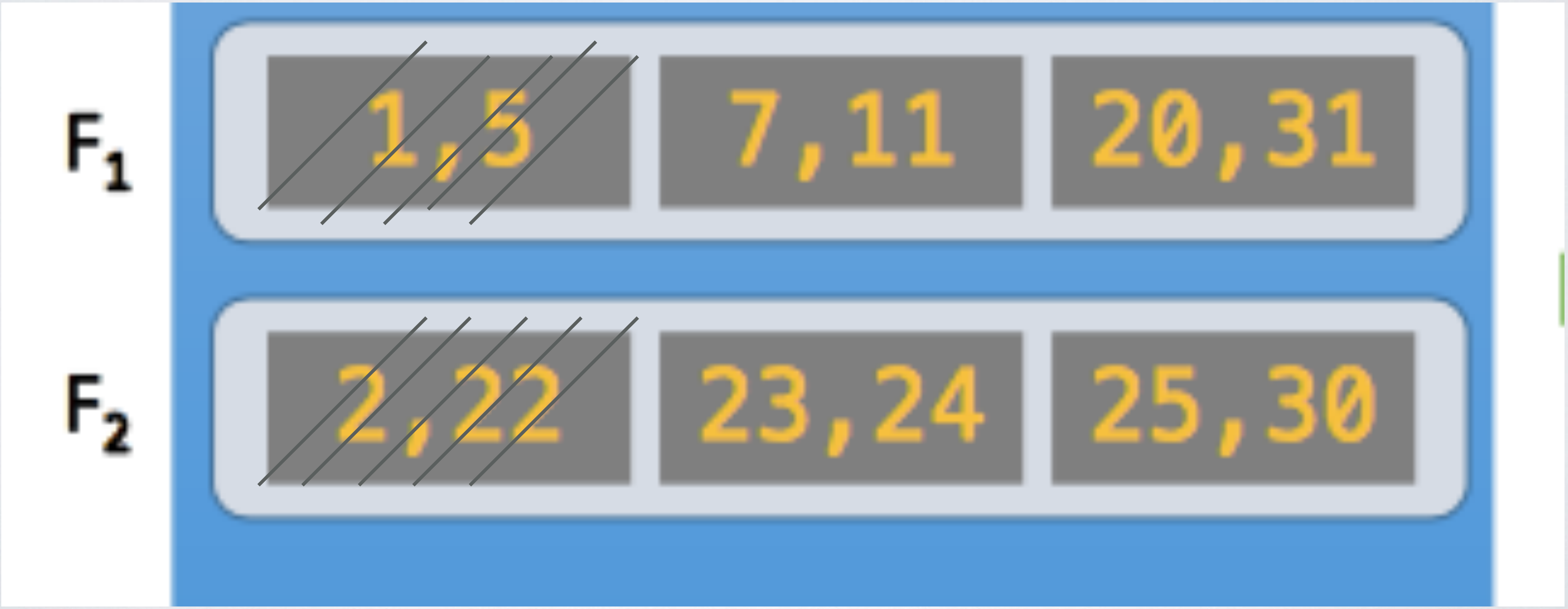


ALGORITHMS

Merge two large sorted files with limited memory-cont

Which File should the algorithm load next?

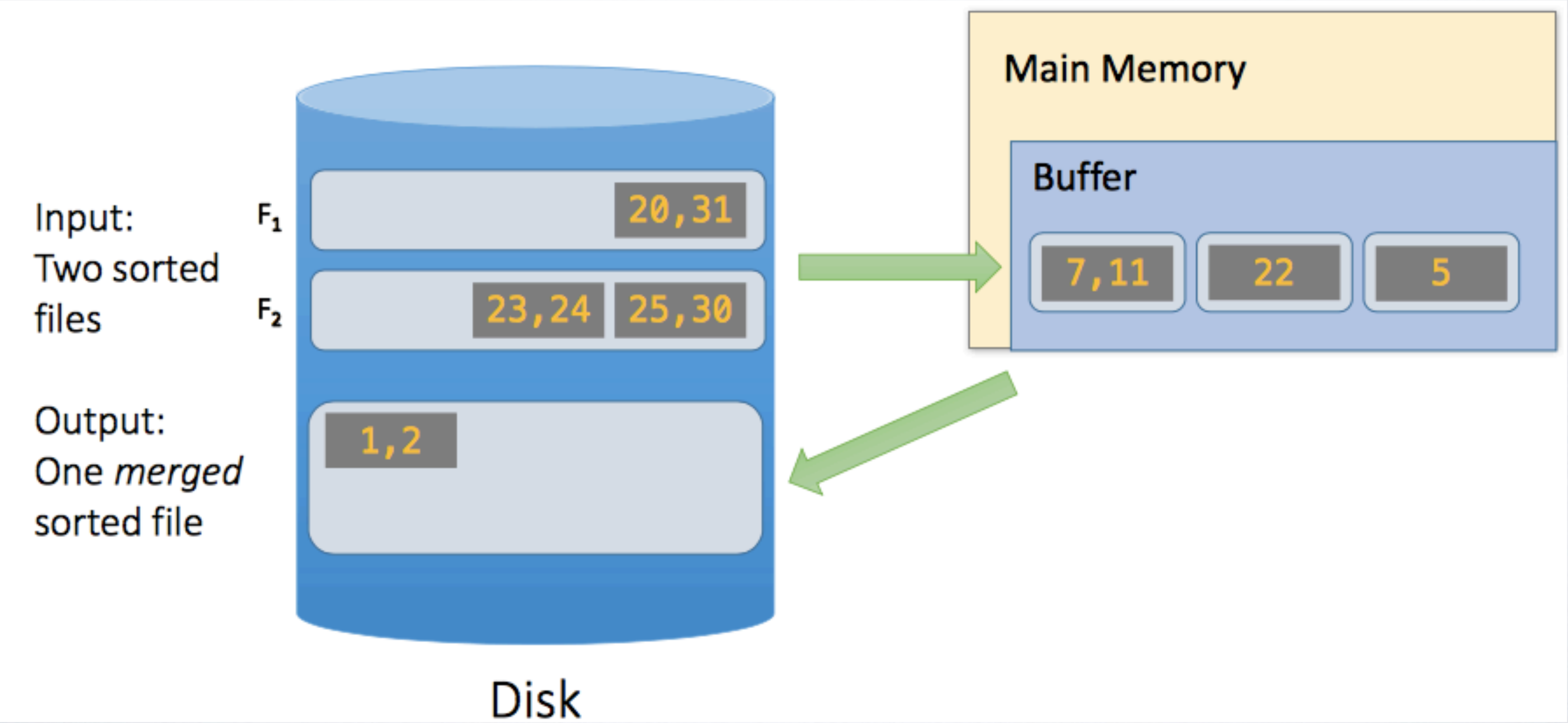
We have already loaded - {1,5} & {2,22}



ALGORITHMS

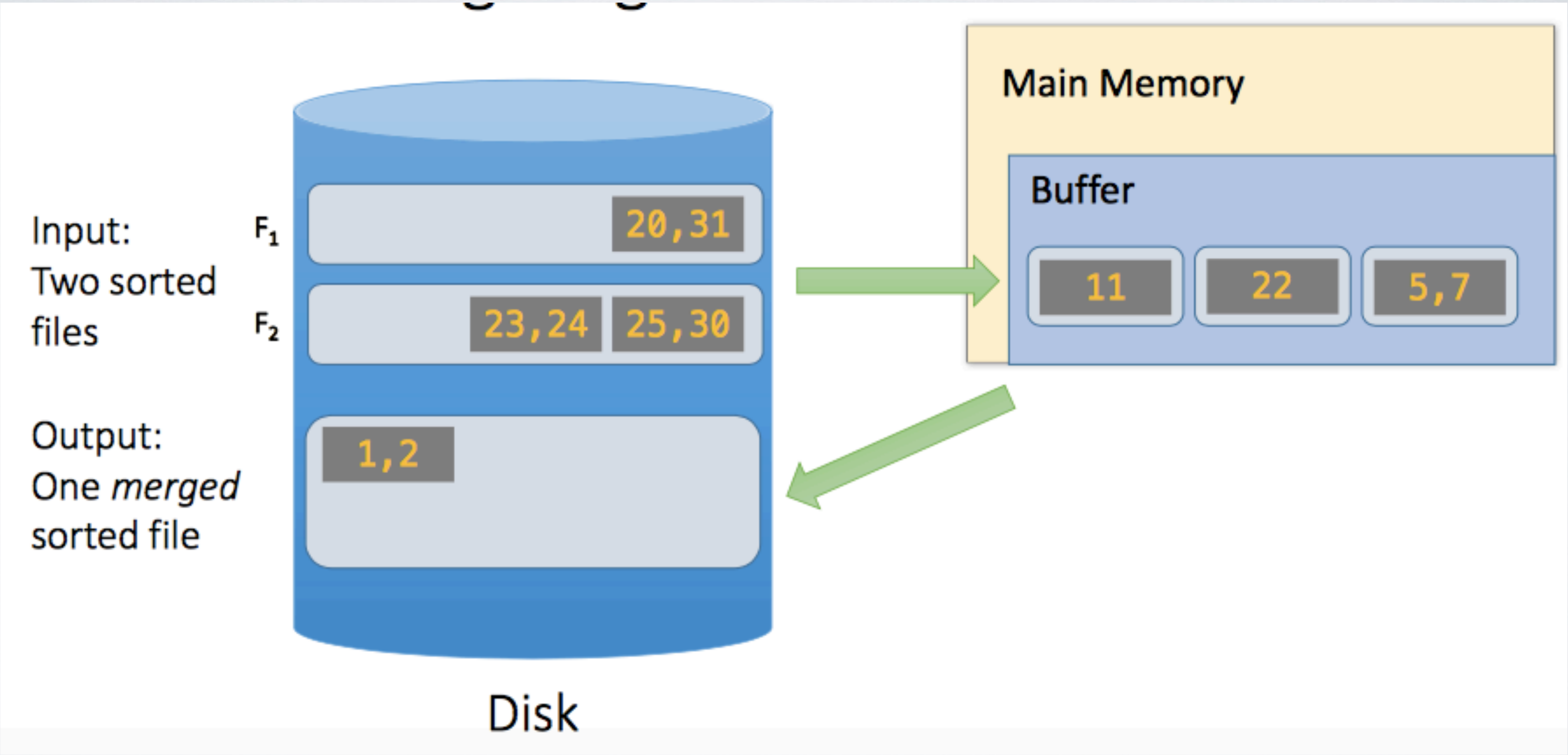
Merge two large sorted files with limited memory-cont

If you chose F_1 You were right!



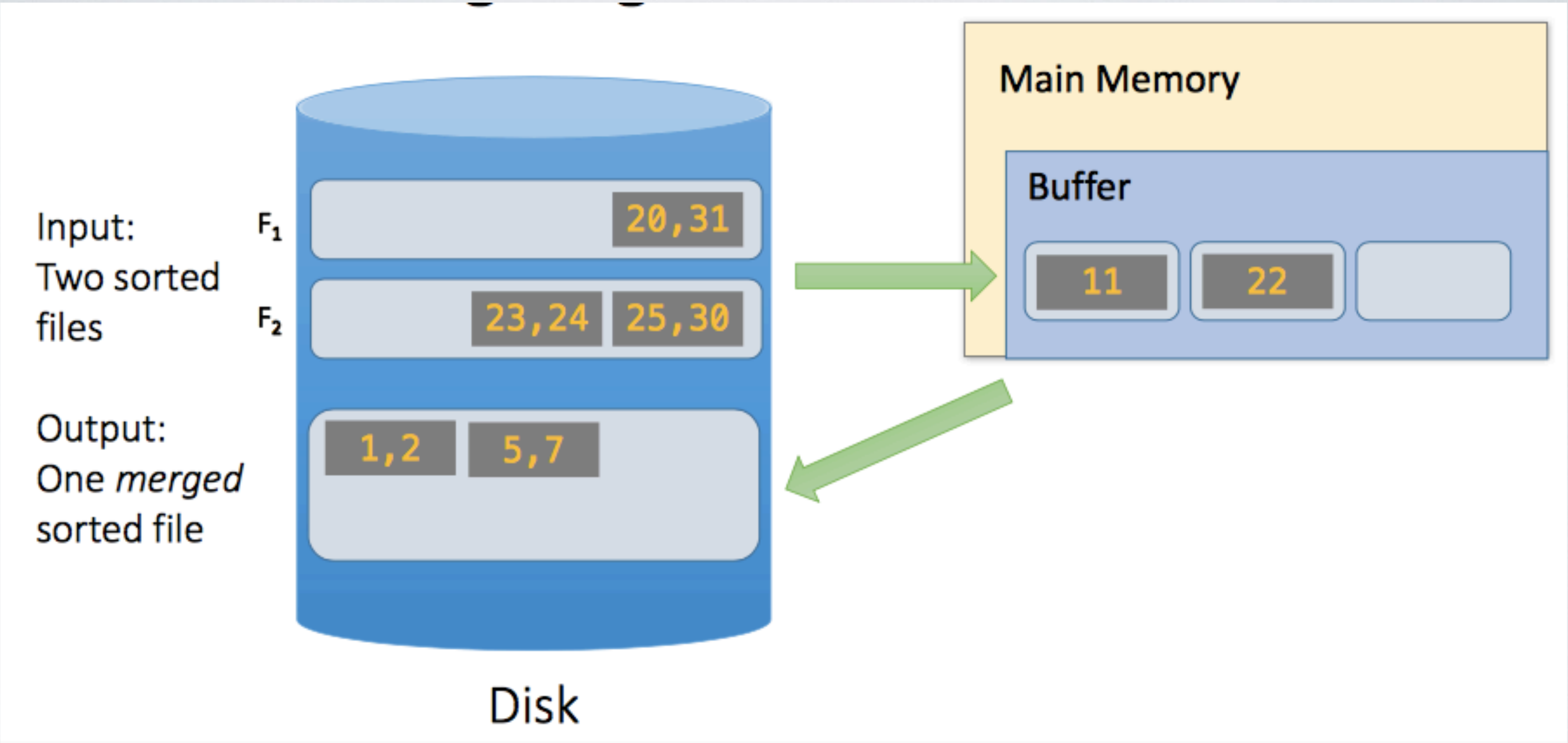
ALGORITHMS

Merge two large sorted files with limited memory-cont



ALGORITHMS

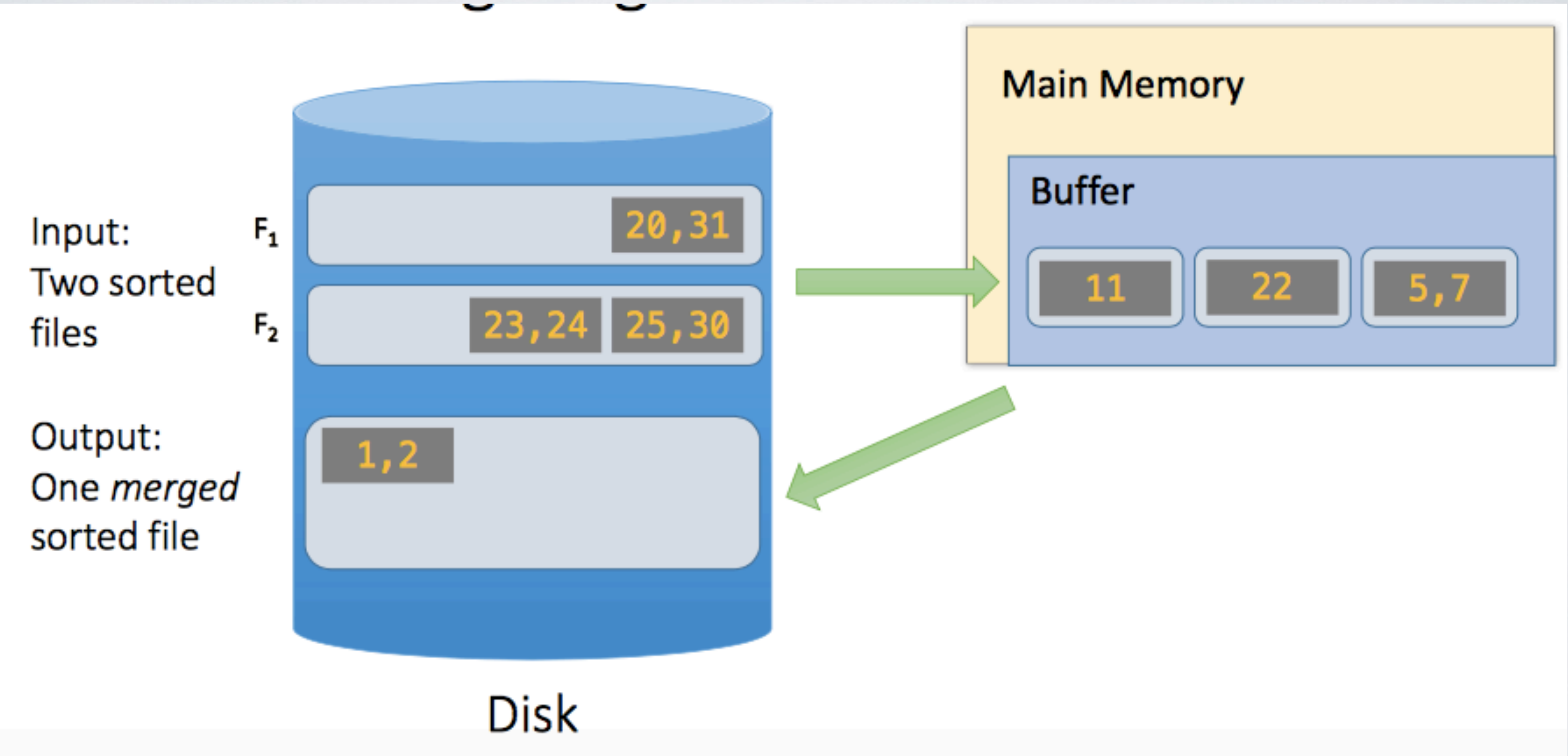
Merge two large sorted files with limited memory-cont



Process will continue as shown until we have merged both sorted lists into one.

ALGORITHMS

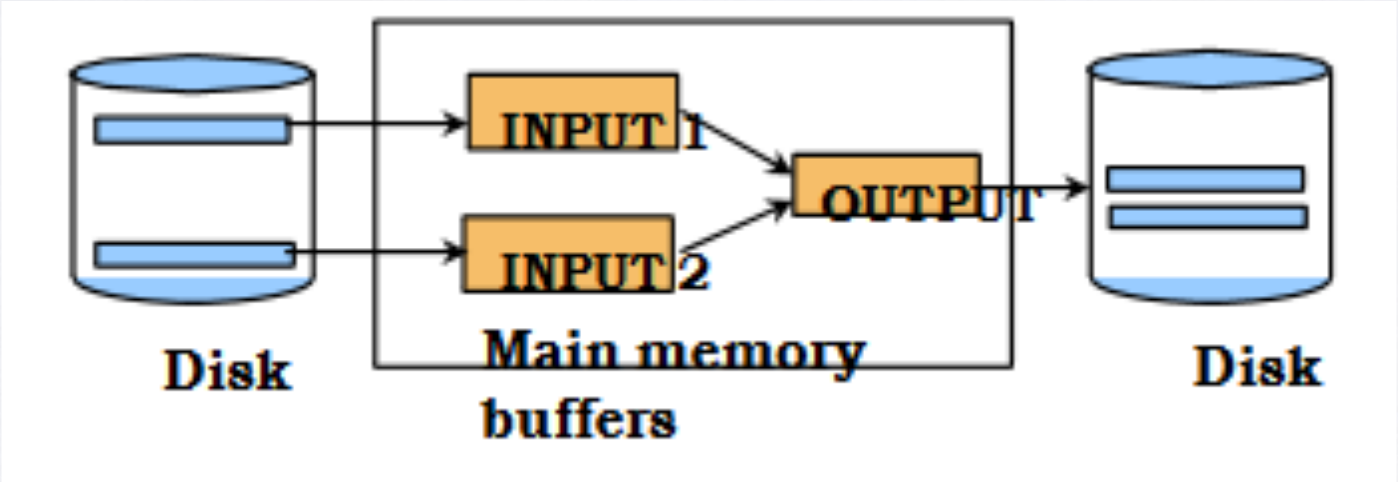
Merge two large sorted files with limited memory-cont



ALGORITHMS
Two-Way Sorting Example

Assumption:

- Only 3 buffers are available

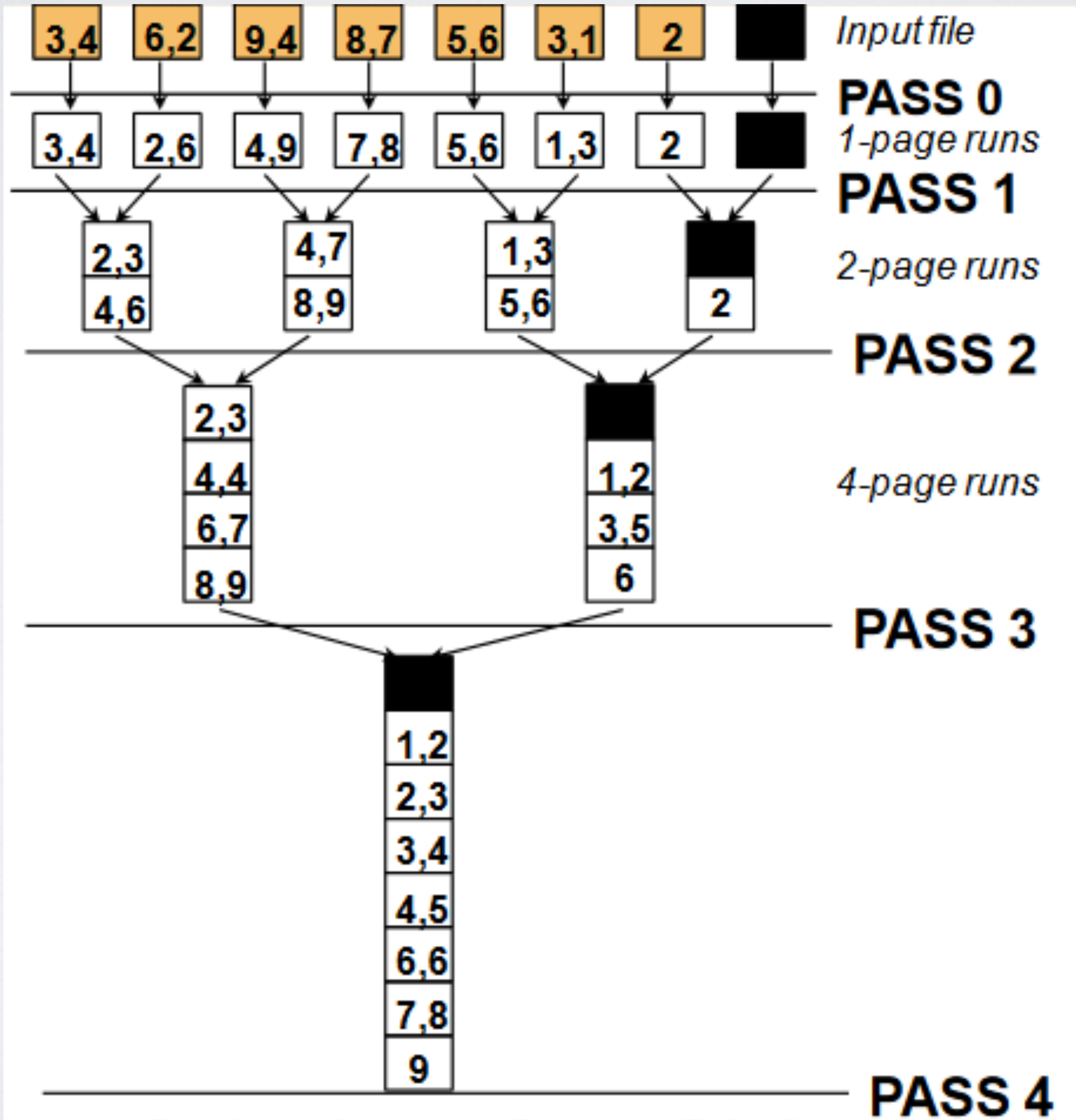


ALGORITHMS
Two-Way Sorting Example

Pass 0:

- Read, sort and write
- Only 1 buffer page is used

ALGORITHMS
Two-Way Sorting Example



ALGORITHMS

Two-Way Sorting Example

- Pass 1,2,... :Merge hierarchically using 3 buffers
  1. Load 2 runs at a time into buffers B[i],b[j]
  2. Initialize i=j=0
  3. Compare elements B[i] and B[j] , move smallest element to output buffer
  4. As output buffer gets full, append to disk and clear the RAM
  5. Repeat above steps until all runs are traverse

ALGORITHMS
ANALYSIS OF TWO-WAY SORTING

- For each pass we read and write N Pages
- Number of passes = LogN + 1
- Total I/O cost = 2N (No of Passes)

  = 2N(LogN+1)

ALGORITHMS
B-WAY MERGE SORT

Assumption:

- B buffers are available

Pass 0:

- Sort N pages using B buffers
- It will generate N/B runs

Pass1,2,..

- Perform (B-1) way merge of runs
- Use B-1 buffers for input and 1 for output

ALGORITHMS
ANALYSIS OF B-WAY SORTING

- No of passes : $1+[\mathrm{Log}_{B-1}(N/B)]$

- I/O cost : $2N(1+[\mathrm{Log}_{B-1}(N/B)])$

K-WAY MERGE SORT

Run Information Phase

- If we have a Memory size M
- Divide input with N elements file into k blocks such that block fits into a main memory
- Create a temp files for each block and save data into the file
- Sort each temp files using merge sort

K-WAY MERGE SORT

Merging Phase

- Merge temp files into one output file
    - Find smallest element from all temp files
    - Remove that element from temp file and copy that to main output file
    - Repeat this till temp files get empty

APPLICATIONS

- GIS - Geographical information systems
- Database systems
- Computational biology
- Computer Graphics and virtual reality systems

APPLICATIONS

NASA'S EOS -(Earth Observing System) Project GIS System

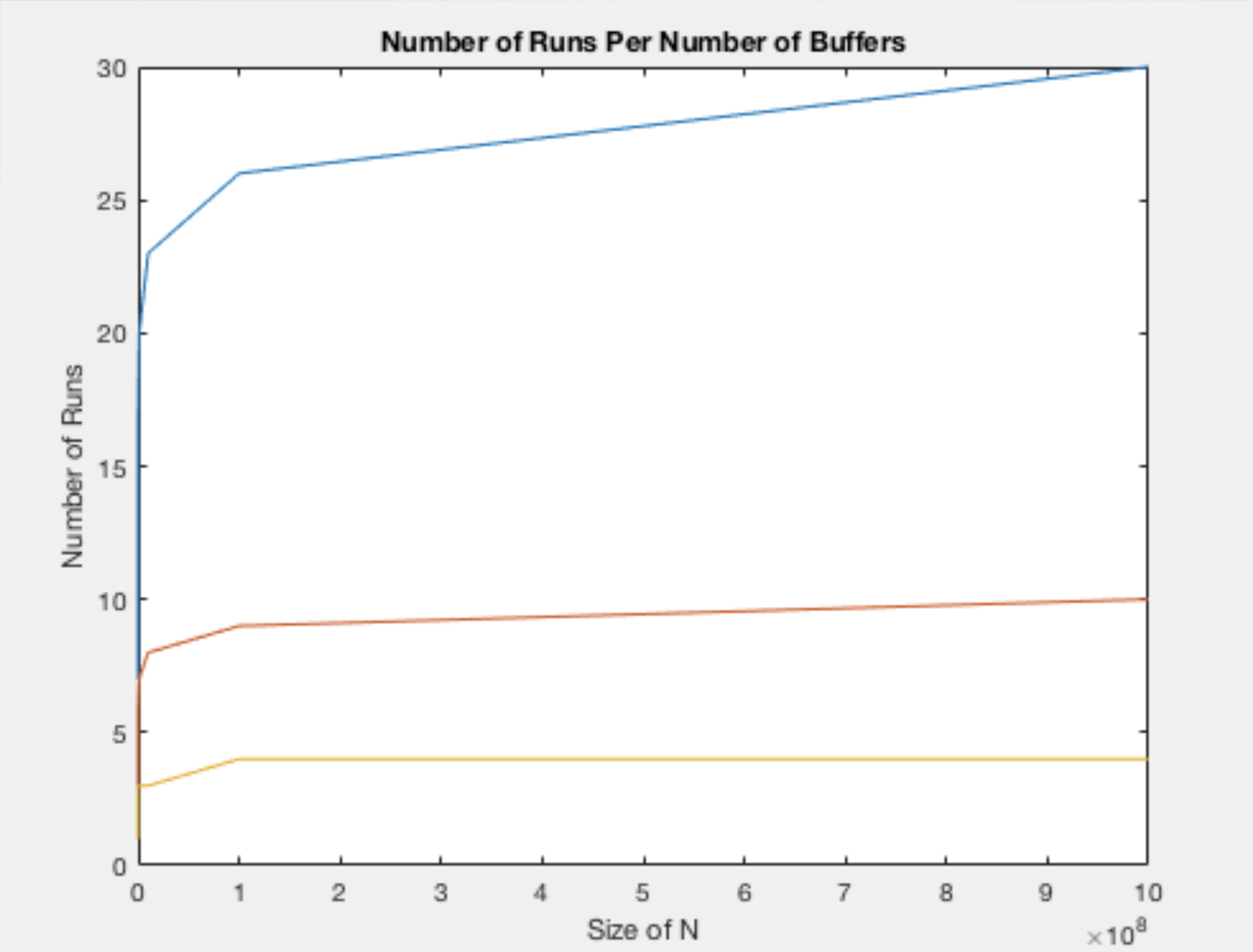
- Polar orbiting satellites observes, land surface, biosphere, atmosphere, and oceans

Manipulates Petabytes of spatial data.

IMPLEMENTATIONS K-WAY MERGE SORT

```
CreateInitialRuns()
{
    Split i/p file in no of runs;
    Apply merge sort on each file;
}
MergeFiles()
{
    Select smallest element from file
    add that to output file till last element
}
```

IMPLEMENTATIONS



Blue = 3 Buffers

Orange= 9 Buffers

Yellow = 257 Buffers

OPEN ISSUES

Asymptotic efficiencies - In different areas of applications.

Example- GIS - General Line Segmentation Intersection Problem -Can it be solved in,

$$O(n \, \text{Logm}(n+t)) \, ?$$

Example - Can one triangulate a simple polygon in a linear number of I/O transfers?

QUESTIONS

- Question #1: External memory sorting analysis focuses on what aspect?
- Question #2: When was the External Memory Model Proposed.
- Question #3: Name one application where External Memory Sort Used?

REFERENCES

- http://jeffe.cs.illinois.edu/teaching/473/01-search+sort.pdf
- http://vargas-solar.com/big-linked-data-keystone/wp-content/uploads/sites/37/2016/07/I-Big-Data-Analytics-Trends.pdf
- https://www.guidesify.com/much-data-google-handle/
- http://www.vlsifacts.com/classification-of-semiconductor-memories-and-computer-memories/
- http://slideplayer.com/slide/2411095/
- https://www.slideshare.net/milindhg/project-report-milindgokhale
- http://www.cs.au.dk/~large/Papers/thesis.pdf

DIAGRAM REFERENCES

- https://web.stanford.edu/class/cs145/lectures/lecture-12-13/Lecture_12-13_EMS_Indexes.pdf

- https://www.cs.ucy.ac.cy/~dzeina/courses/epl446/lectures/09.pdf

- http://slideplayer.com/slide/6826122/

THANK YOU!