Mini Project

Building a Search Engine

Phase - II

Information Retrieval and Extraction CSE 4731

Abhishek Singh

Project Task

- Data: Wikipedia English Dump ~ 46 GB
  - Data link - ftp://10.4.17.131/Datasets/IRE_Monsoon_2017/WikiSearch/
  - enwiki-latest-pages-articles-multistream.xml.bz2 (for Phase II)
- Search time < 1sec (200-500ms) (Java) ; (< 5 sec; Python)
- Index size ~ 10 GB (less than ¼ of data size)
- Support for field queries
- External tools and libraries like Lucene, WikiXMLj, elasticsearch, redis, etc not allowed.

System capabilities

- Given a query / Field query output top 10 results(title of wiki document) sorted by relevancy of document with respect to give query.

- Relevant results should come within expected time limit.
  - < 1s for Java, C++
  - < 5s for Python

Phase II

- Inverted index creation on whole wiki dump (~ 46 GB)
- Implement Ranking mechanism
- End to End search system

Scalable Inverted Index Creation

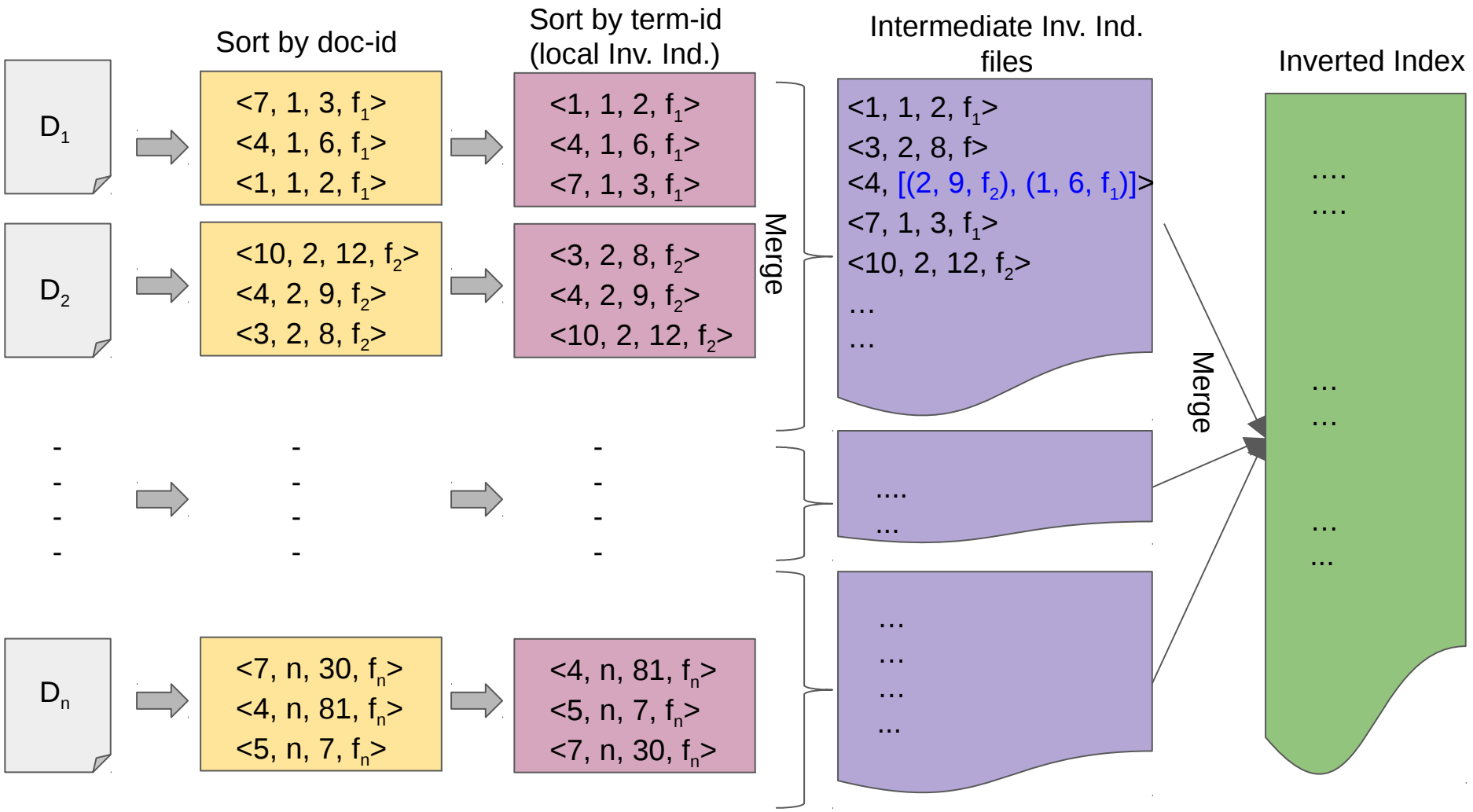
- Main challenge is to build a huge index with limited memory.

- Sort-based Method
  - Build local inverted index
  - Merge local inverted index
  - Obtain large inverted index

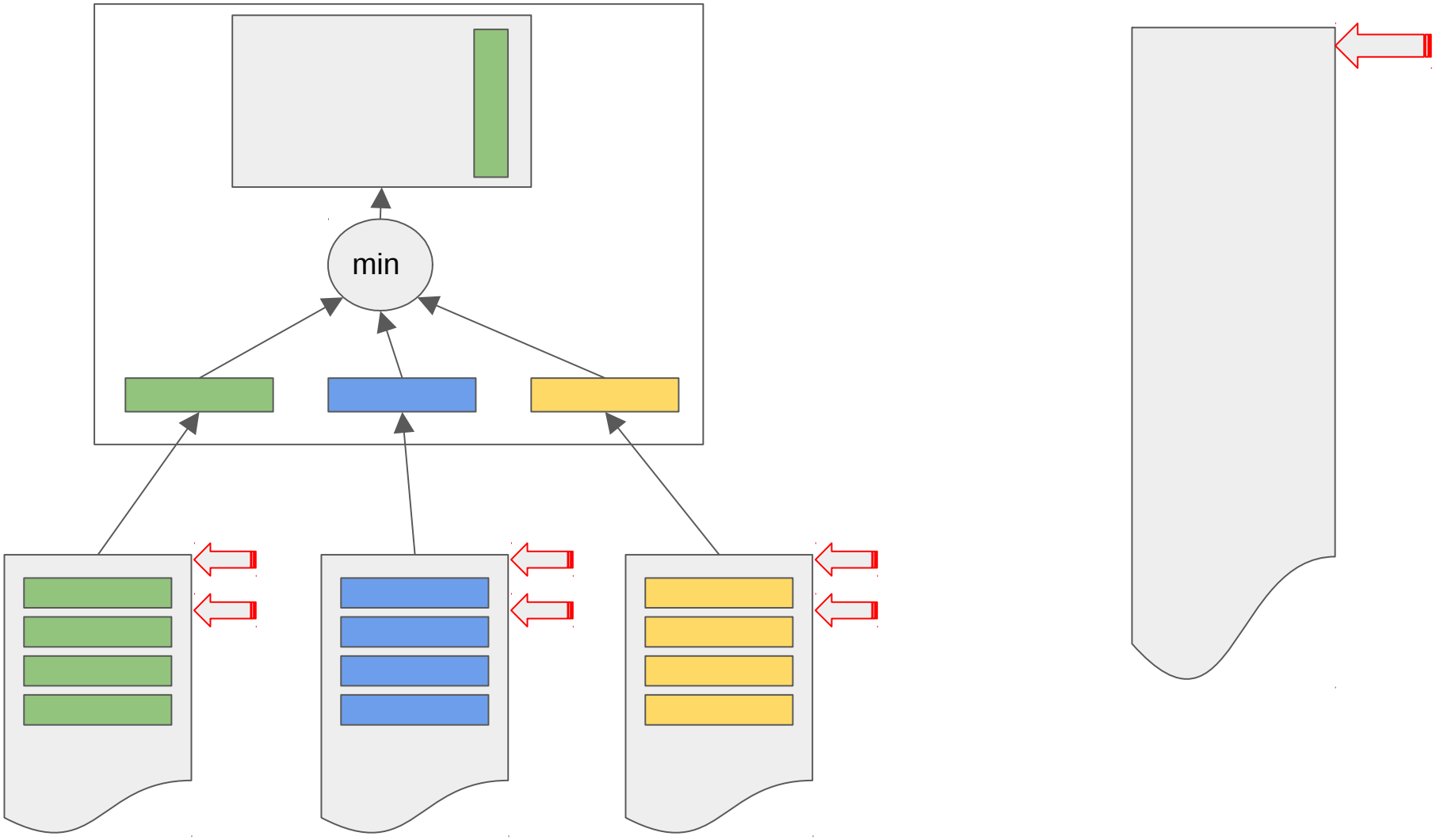
What information is needed about a term?

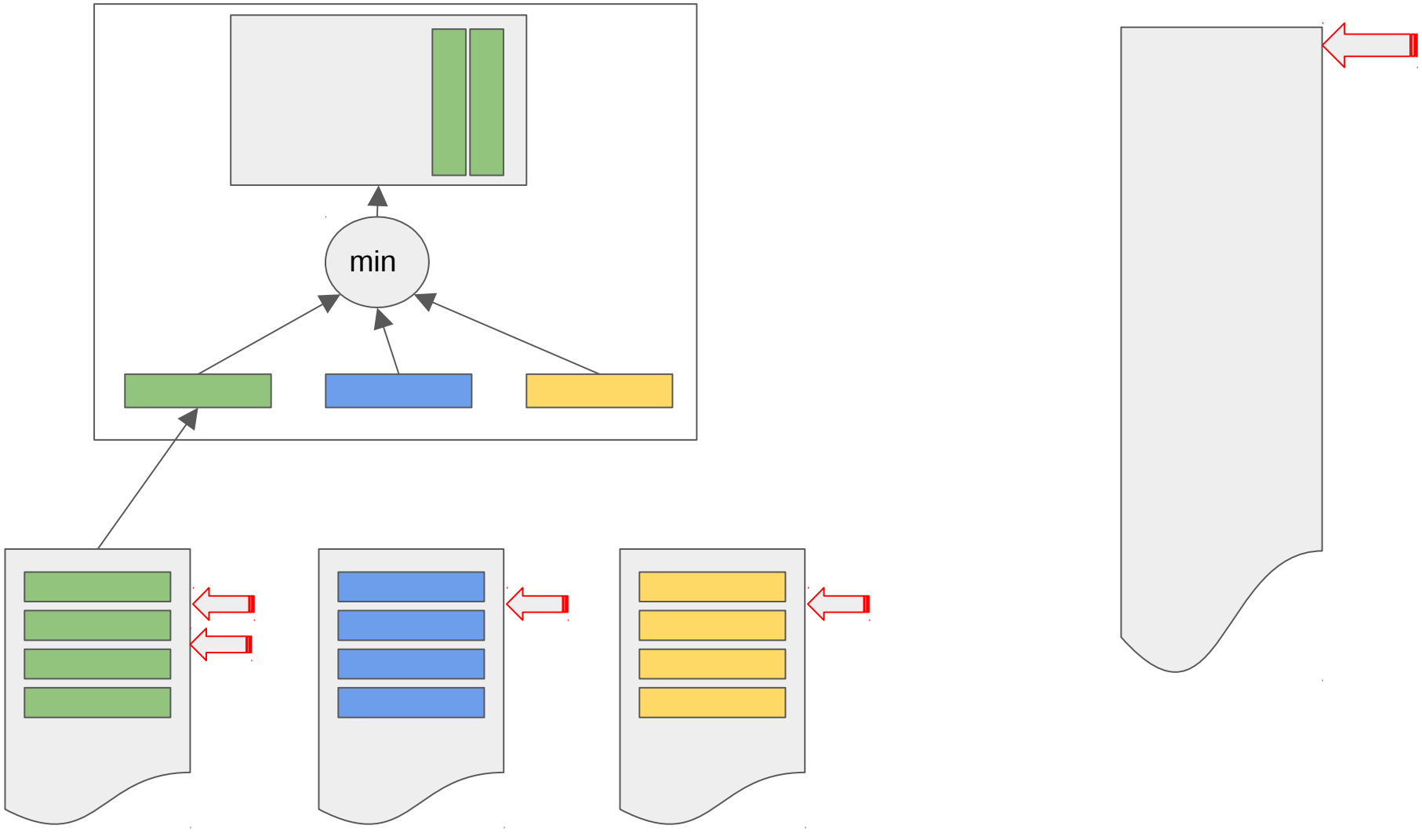
- <term id/word, doc id, term freq, field info>

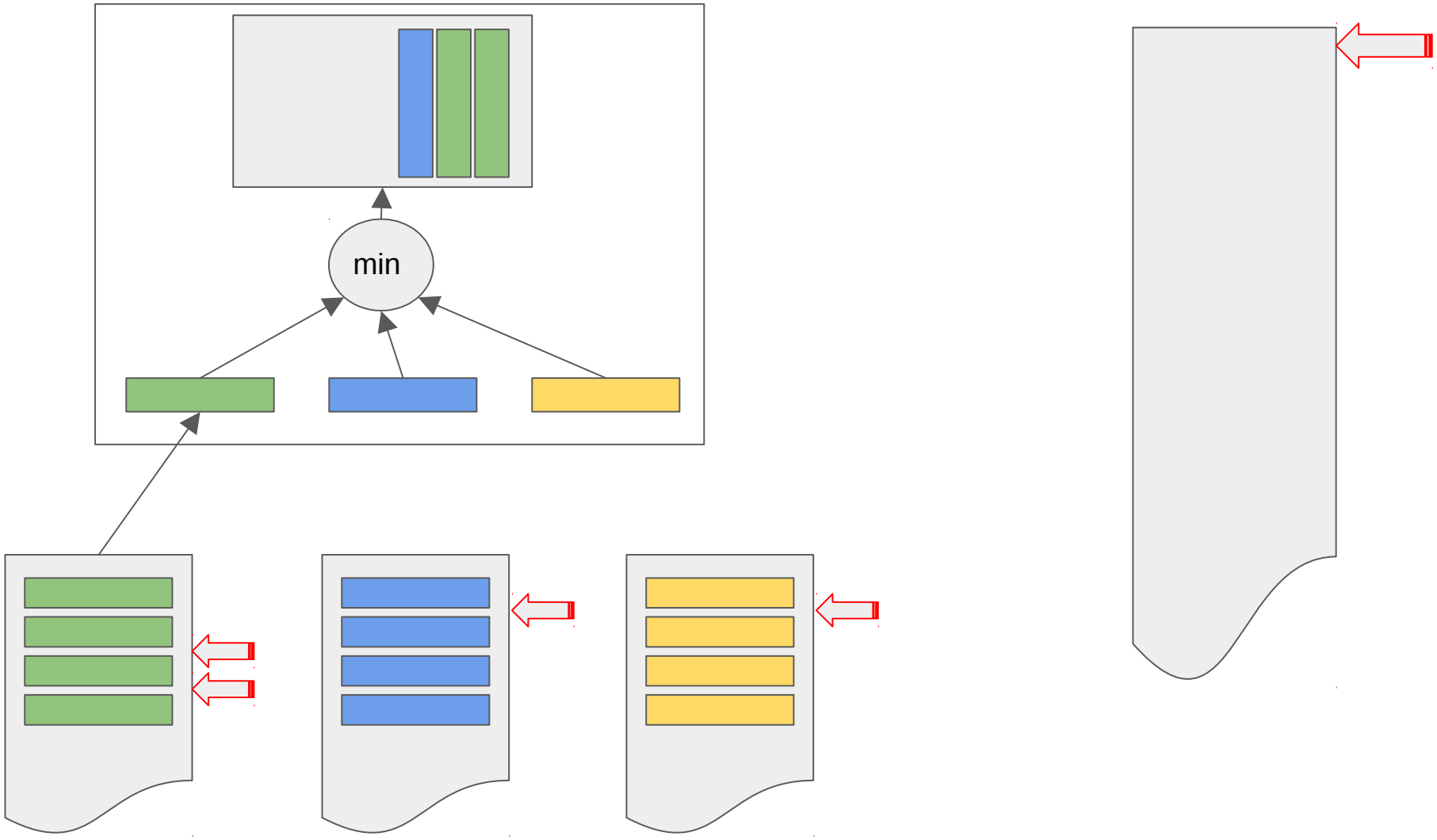
  Eg: <2, 12, 354, t:1b:4:c2>

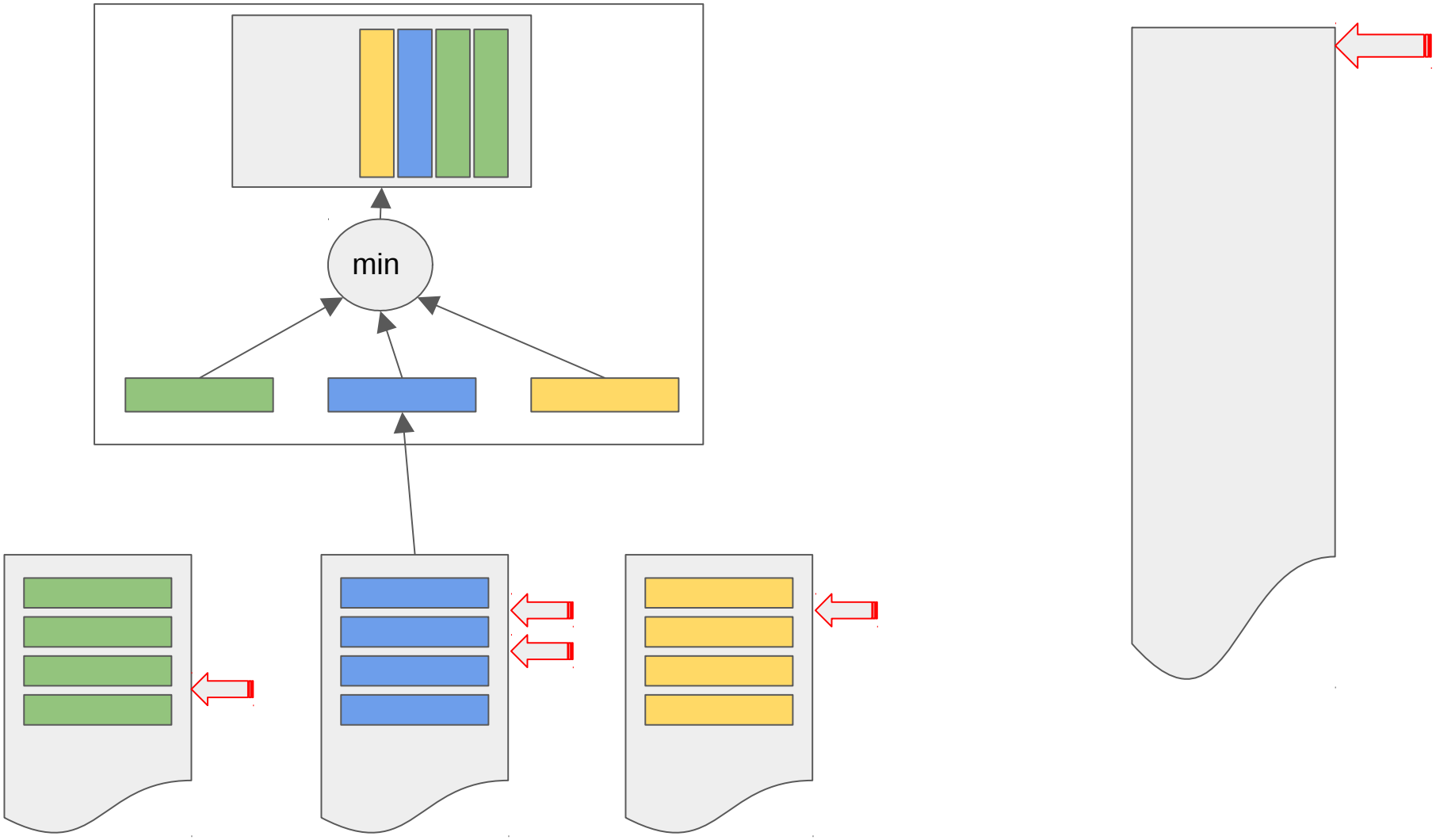

Sort by doc-id
Sort by term-id
(local Inv. Ind.)
Intermediate Inv. Ind.
files
Inverted Index
D1
D2
Dn
<7, 1, 3, f1>
<4, 1, 6, f1>
<1, 1, 2, f1>
<10, 2, 12, f2>
<4, 2, 9, f2>
<3, 2, 8, f2>
<7, n, 30, fn>
<4, n, 81, fn>
<5, n, 7, fn>
<1, 1, 2, f1>
<4, 1, 6, f1>
<7, 1, 3, f1>
<3, 2, 8, f2>
<4, 2, 9, f2>
<10, 2, 12, f2>
<4, n, 81, fn>
<5, n, 7, fn>
<7, n, 30, fn>
Merge
<1, 1, 2, f1>
<3, 2, 8, f>
<4, [(2, 9, f2), (1, 6, f1)]>
<7, 1, 3, f1>
<10, 2, 12, f2>
Merge

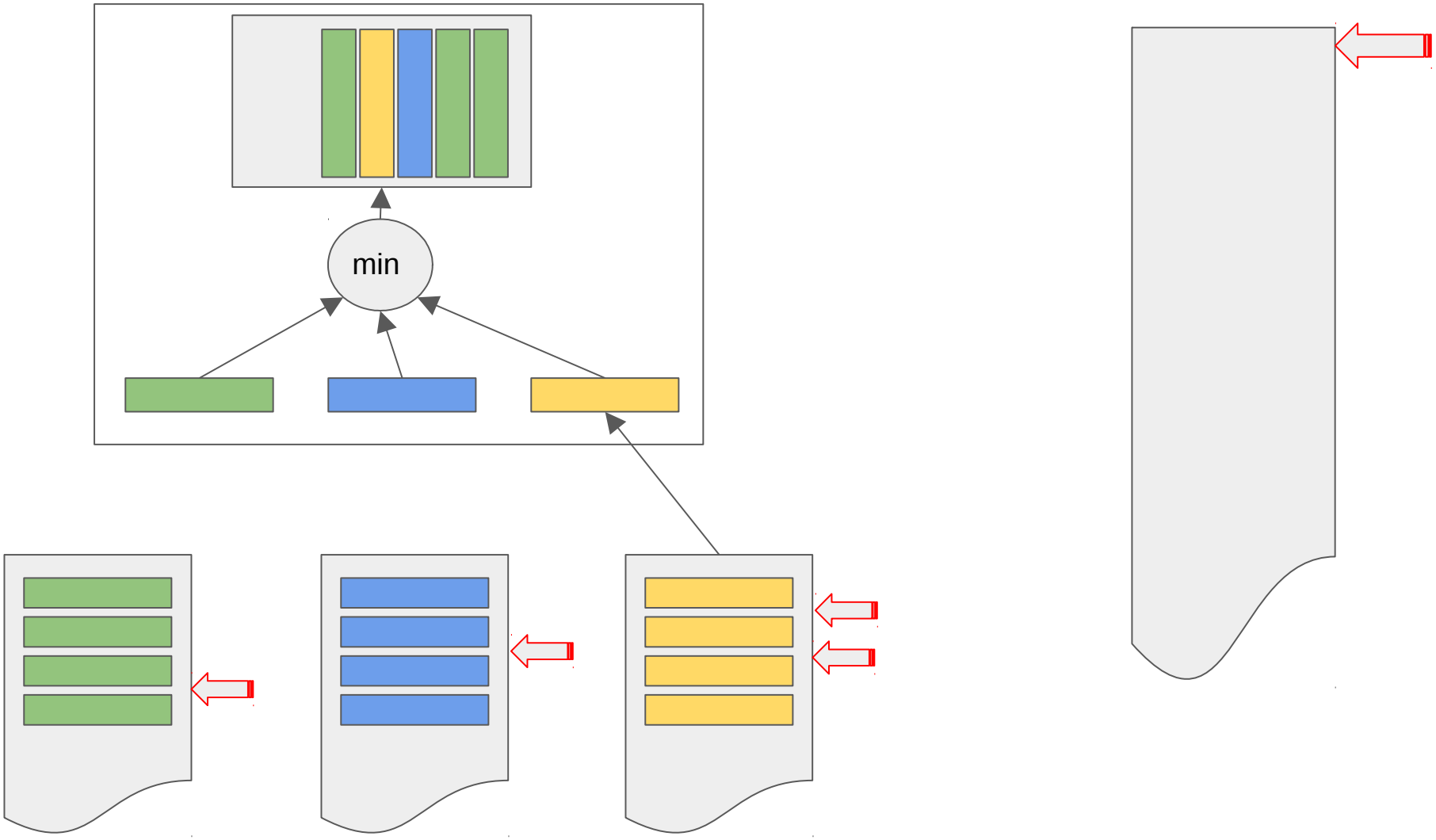
Merging sorted files

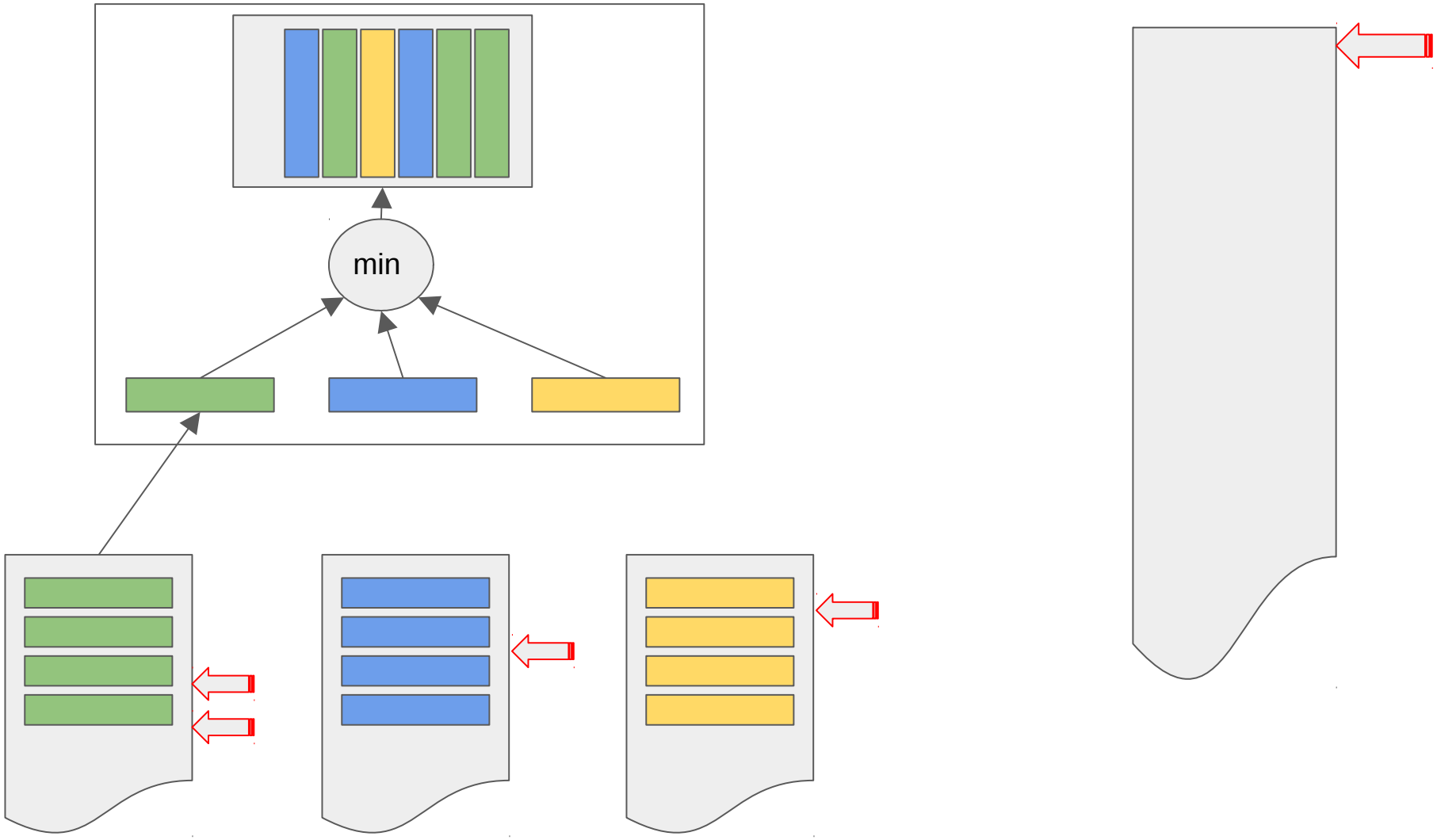

min

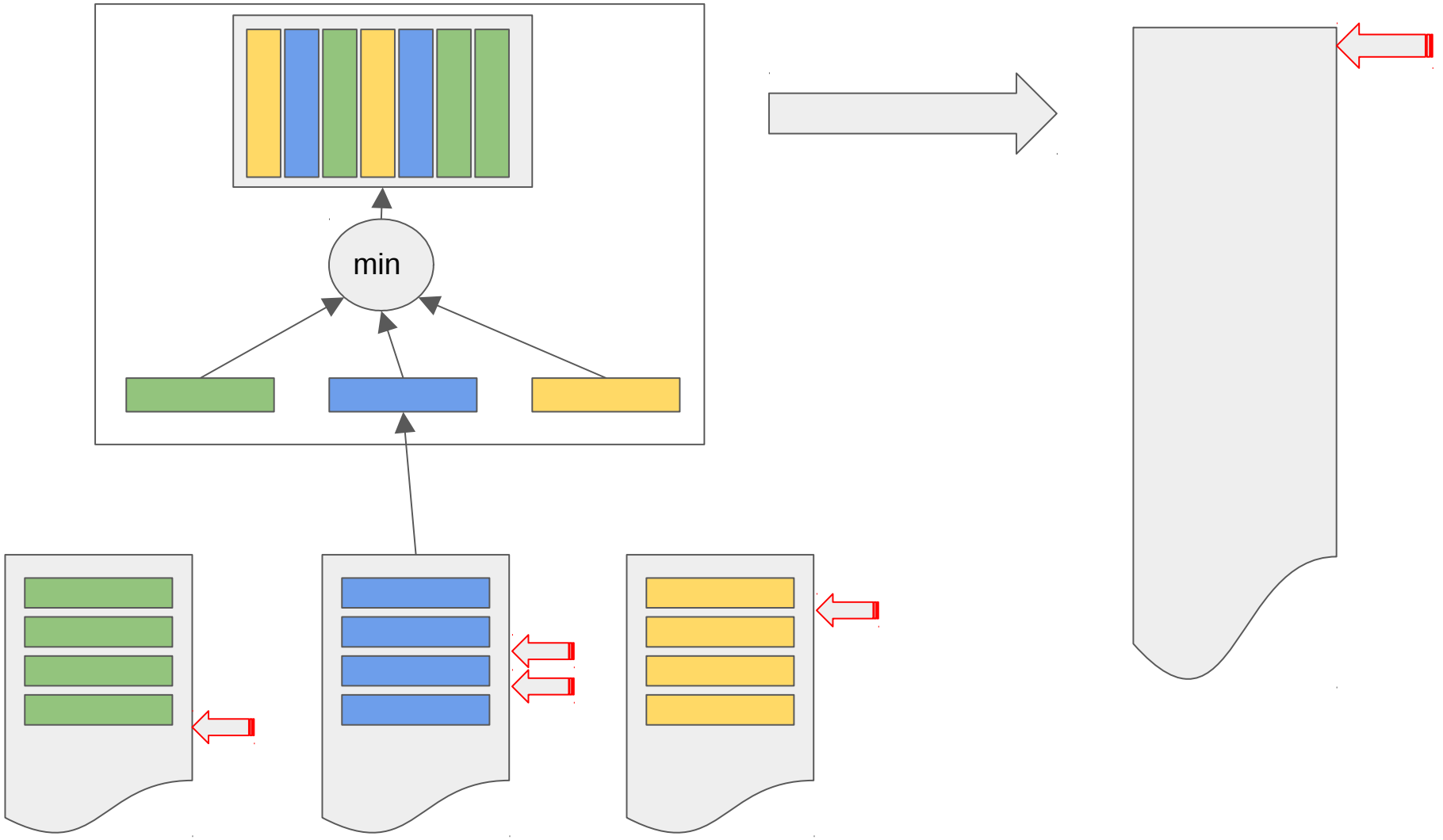

min

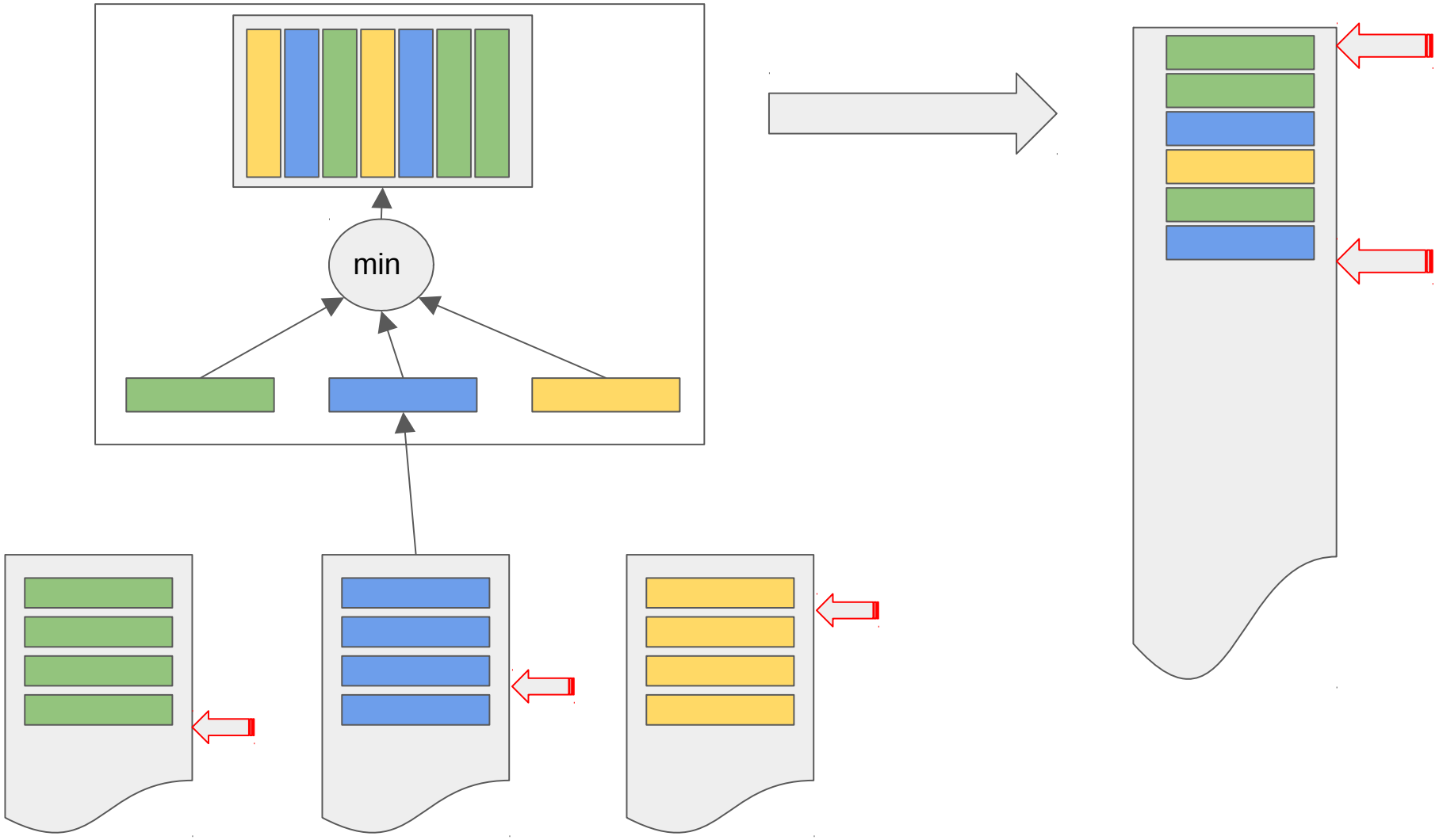

min


min


min

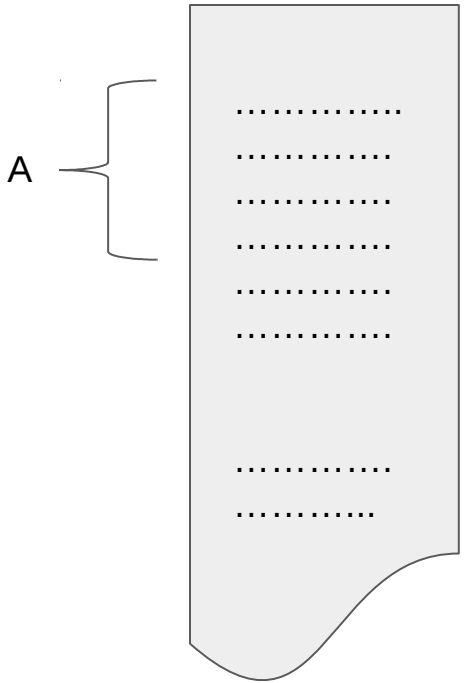

min


min

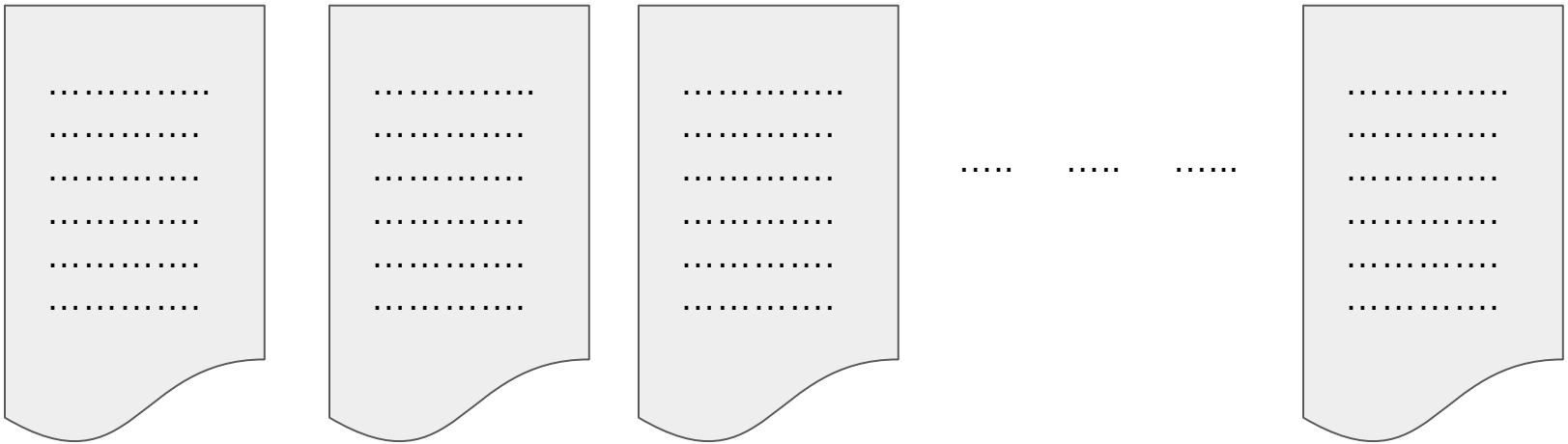

min

Levels of Indexing...

First letter of Term	Range (line no.)
A	1-1000
B	1001-3050
C	3051 - 4800
….	...
Z	10000-10080



Levels of Indexing..



Mapping

A-C	D-F	G-I	X-Z
0.txt	1.txt	2.txt	8.txt

Thank You!