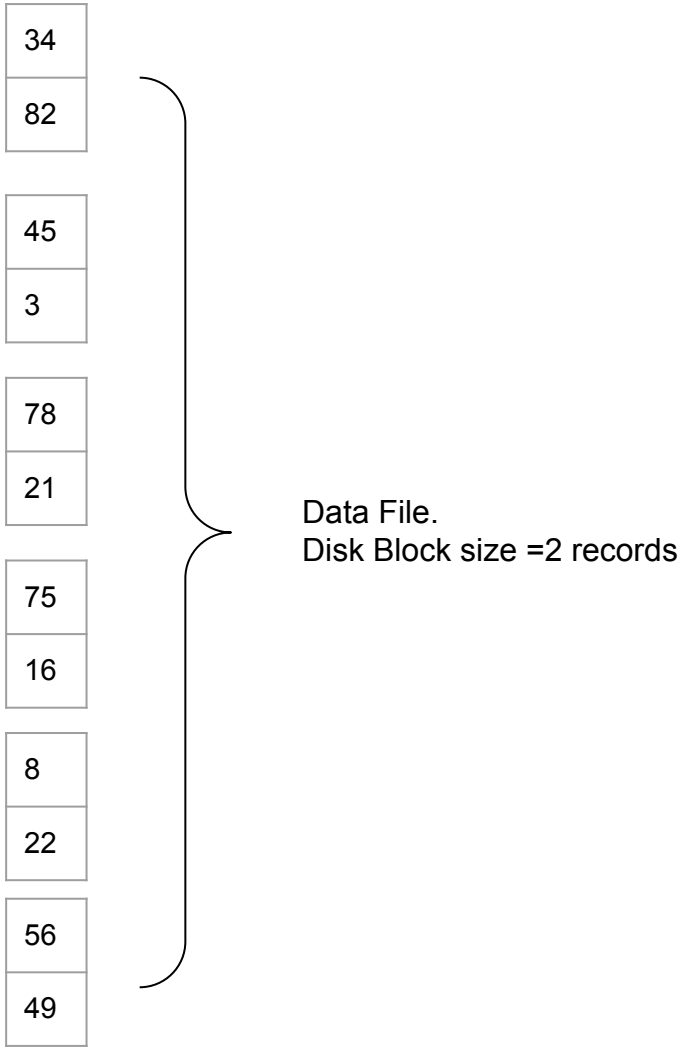
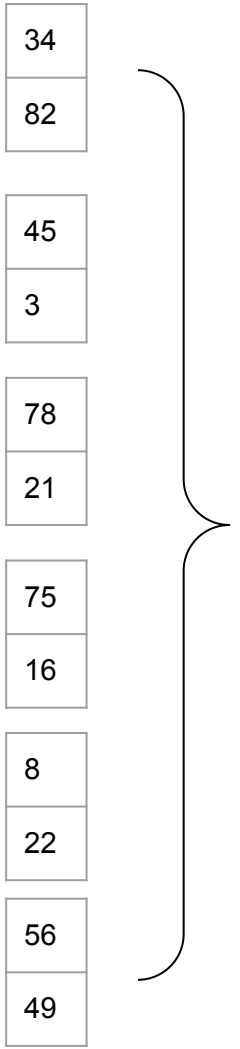
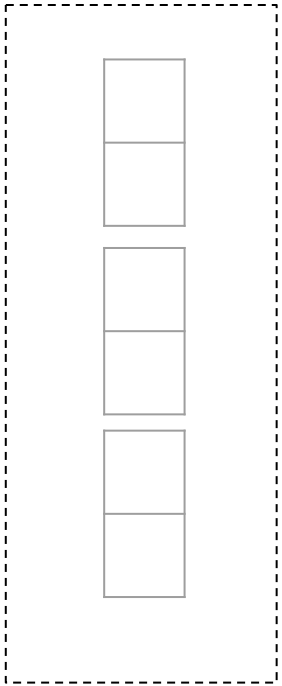
External Sorting with N-way Merge

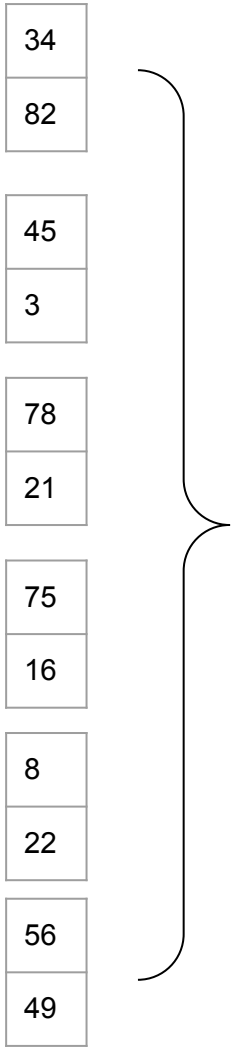

34
82
45
3
78
21
75
16
8
22
56
49
Data File.
Disk Block size =2 records



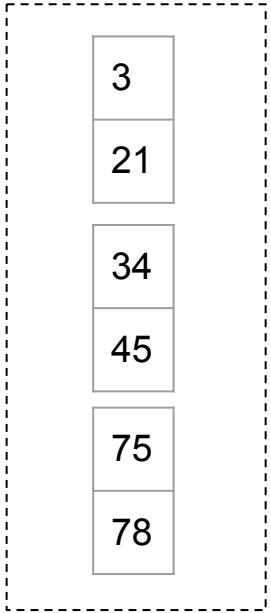
Data File.
Disk Block size =2 records



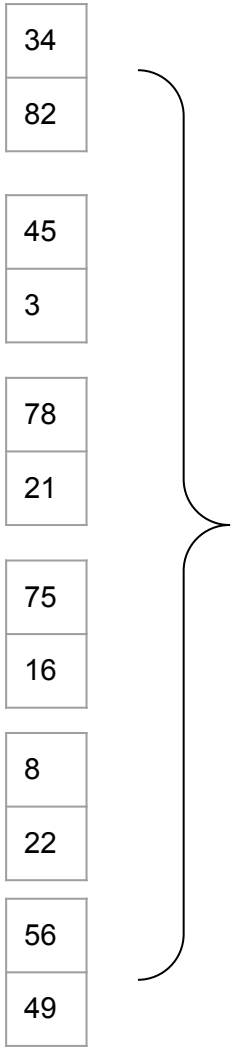
Main Memory available
M=3 (i.e., 3 disk blocks can be stored in Main Memory)



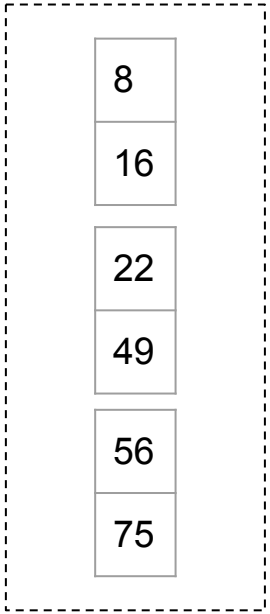
Data File.
Disk Block size =2 records



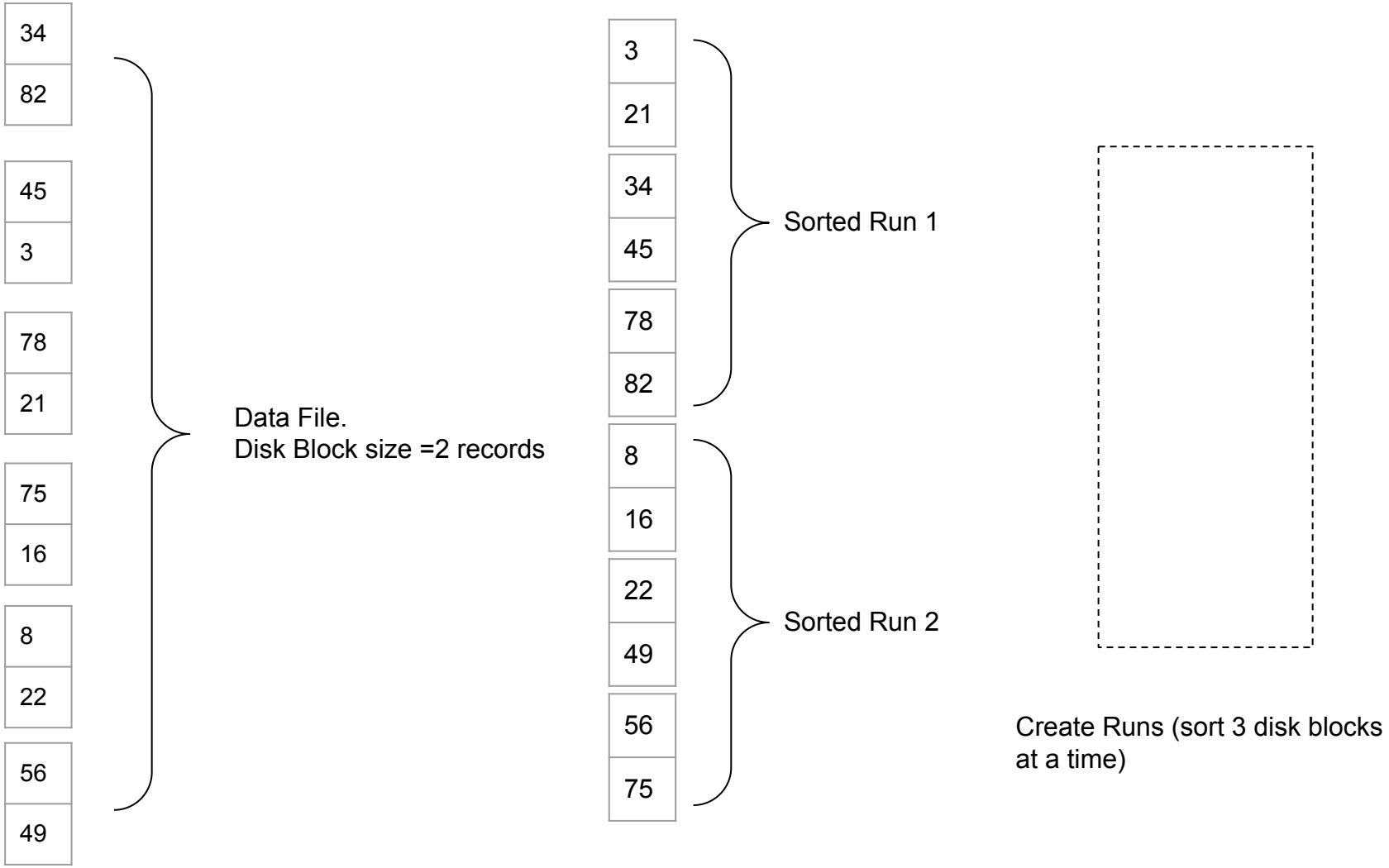
Create Runs (sort 3 disk blocks at a time)



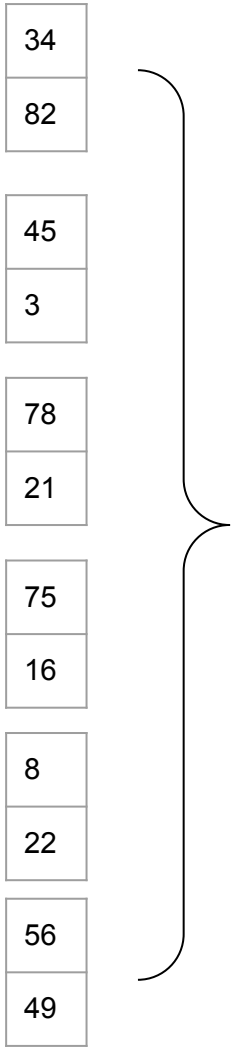
Data File.
Disk Block size =2 records



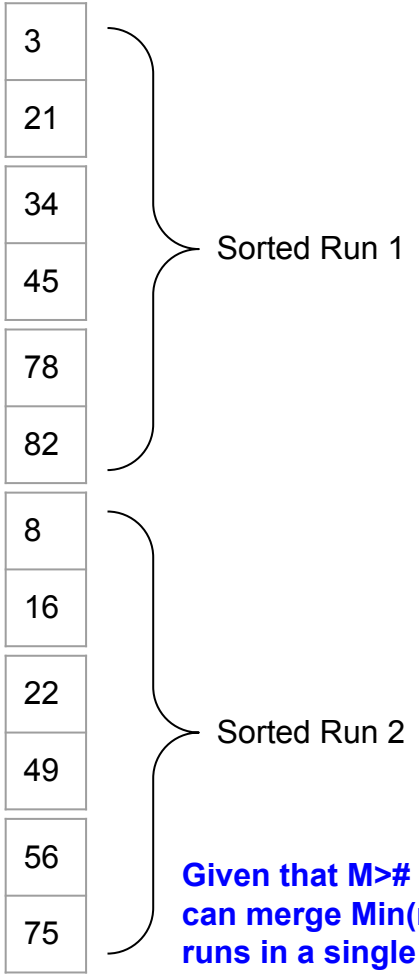
Create Runs (sort 3 disk blocks at a time)


34
82
45
3
78
21
75
16
8
22
56
49
Data File.
Disk Block size =2 records
3
21
34
45
78
82
Sorted Run 1
8
16
22
49
56
75
Sorted Run 2
Create Runs (sort 3 disk blocks at a time)

CASE 1: M > #Runs

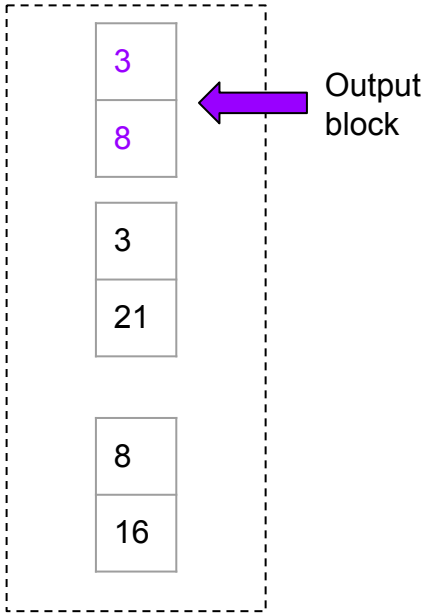


Data File.
Disk Block size =2 records



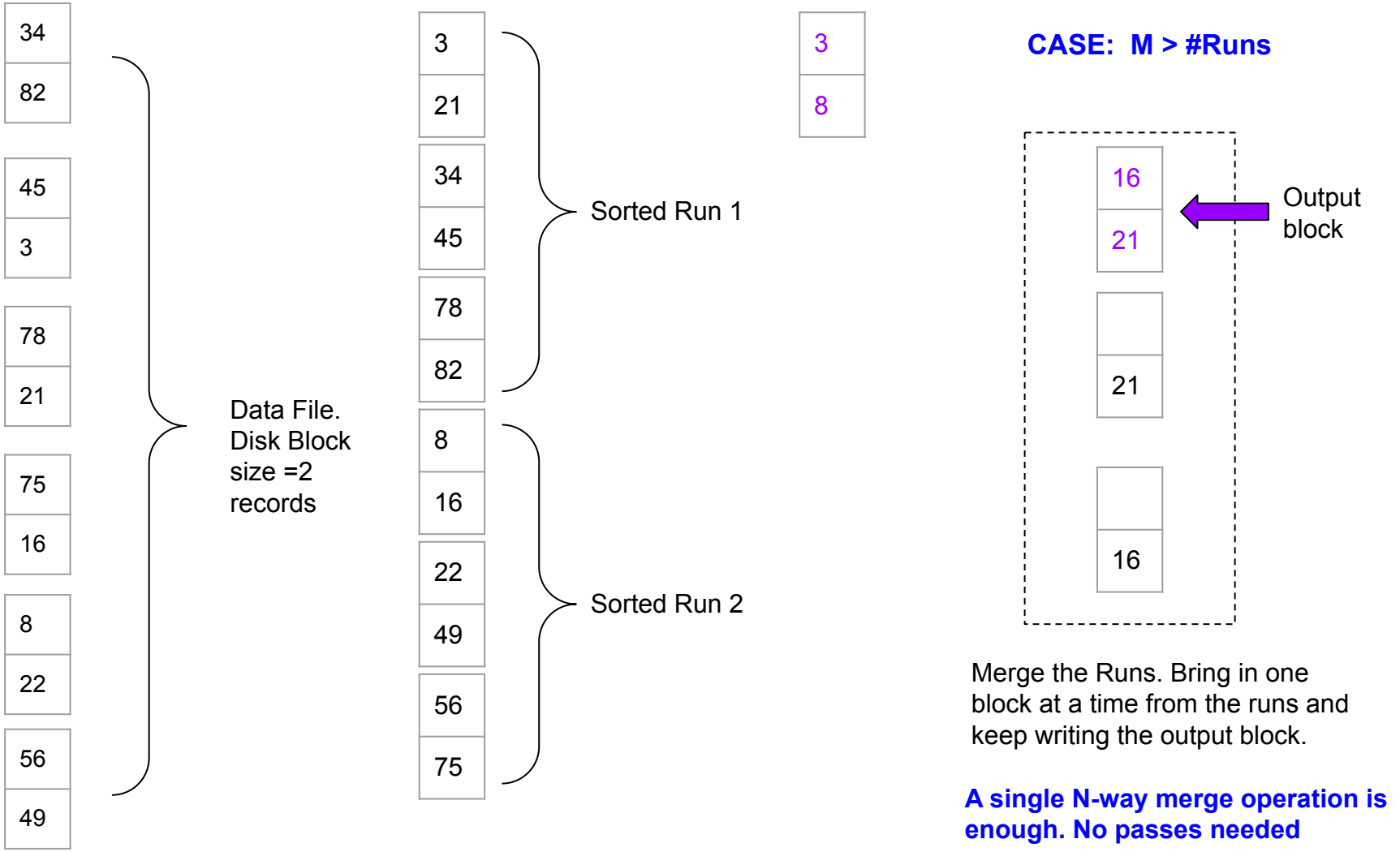
Given that M># runs → we can merge Min(runs, M-1) runs in a single shot. We use N-way merge operation

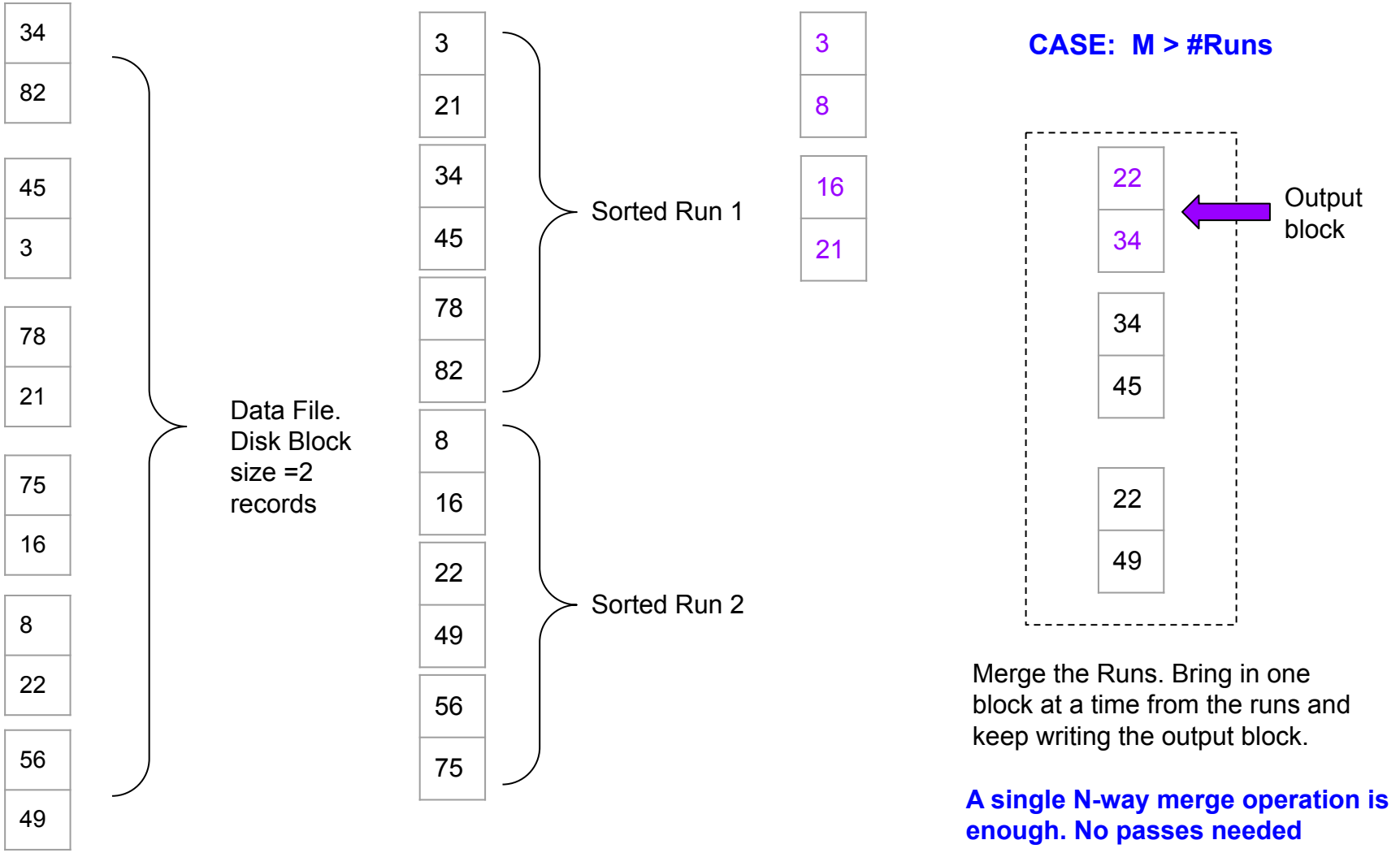
CASE: M > #Runs

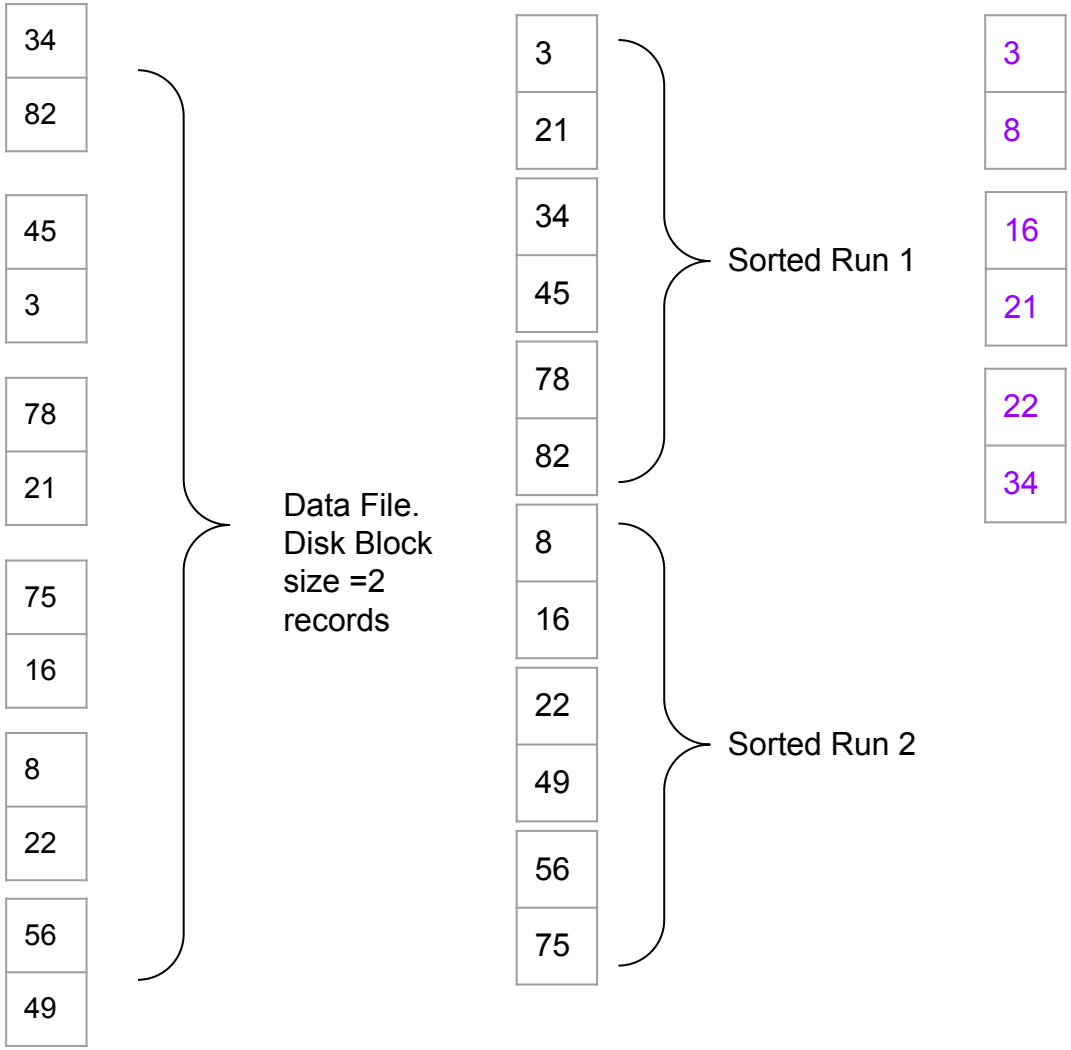


Merge the Runs. Bring in one block at a time from the runs and keep writing the output block.

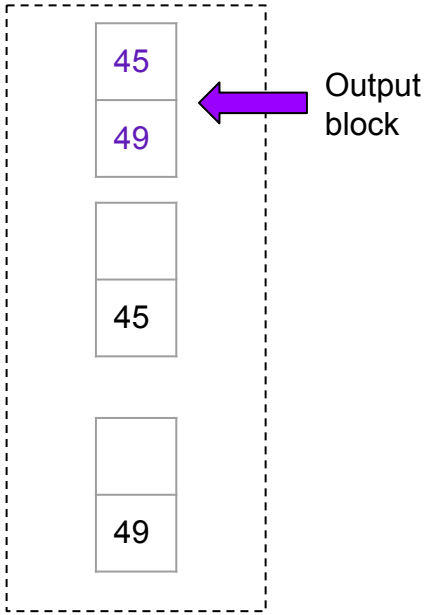
A single N-way merge operation is enough. No passes needed


34
82
45
3
78
21
75
16
8
22
56
49
Data File.
Disk Block
size =2
records
3
21
34
45
78
82
Sorted Run 1
8
16
22
49
56
75
Sorted Run 2
3
8
CASE: M > #Runs
16
21
Output
block
21
16
Merge the Runs. Bring in one block at a time from the runs and keep writing the output block.
A single N-way merge operation is enough. No passes needed


34
82
45
3
78
21
75
16
8
22
56
49
Data File.
Disk Block
size =2
records
3
21
34
45
78
82
Sorted Run 1
8
16
22
49
56
75
Sorted Run 2
3
8
16
21
CASE: M > #Runs
22
34
Output
block
34
45
22
49
Merge the Runs. Bring in one block at a time from the runs and keep writing the output block.
A single N-way merge operation is enough. No passes needed

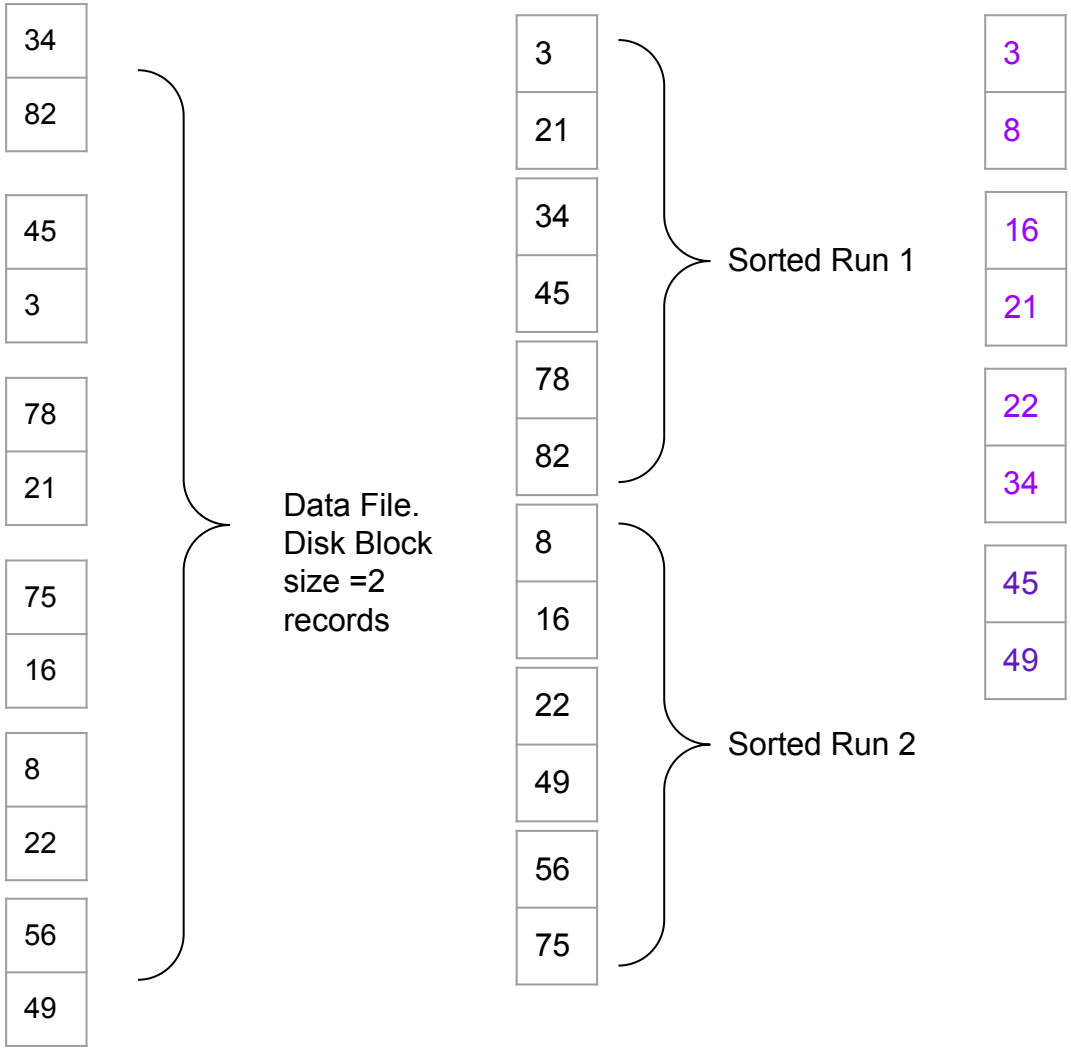


CASE: M > #Runs

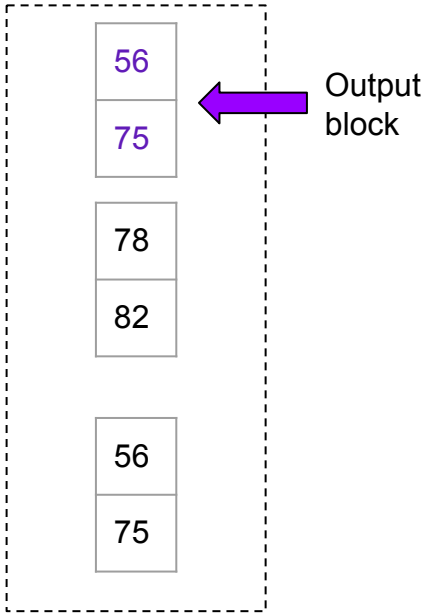


Merge the Runs. Bring in one block at a time from the runs and keep writing the output block.

A single N-way merge operation is enough. No passes needed

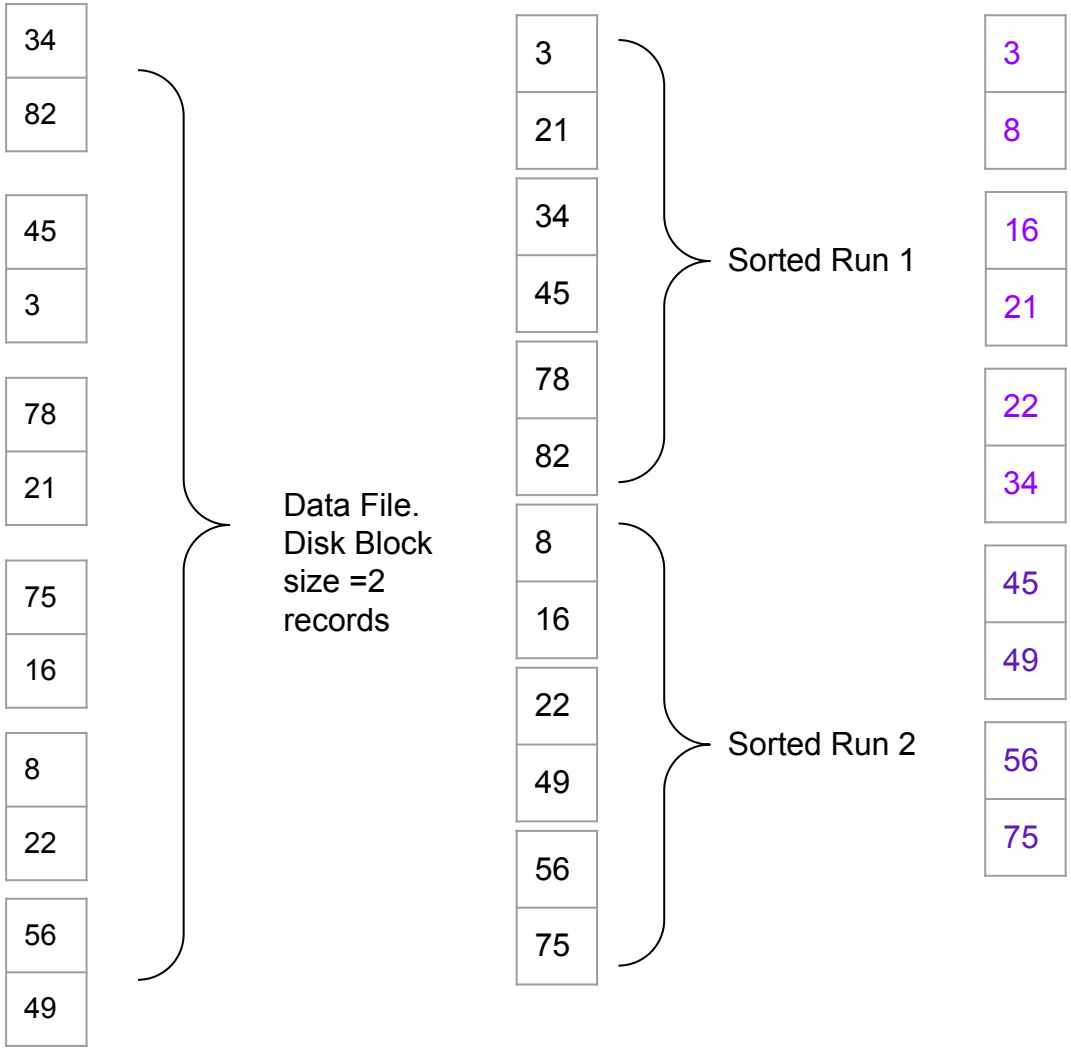


CASE: M > #Runs

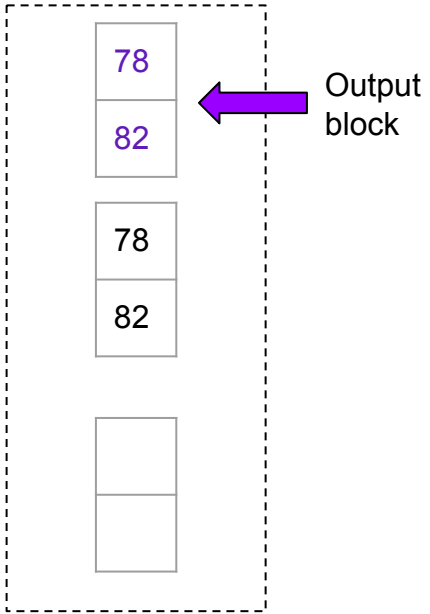


Merge the Runs. Bring in one block at a time from the runs and keep writing the output block.

A single N-way merge operation is enough. No passes needed

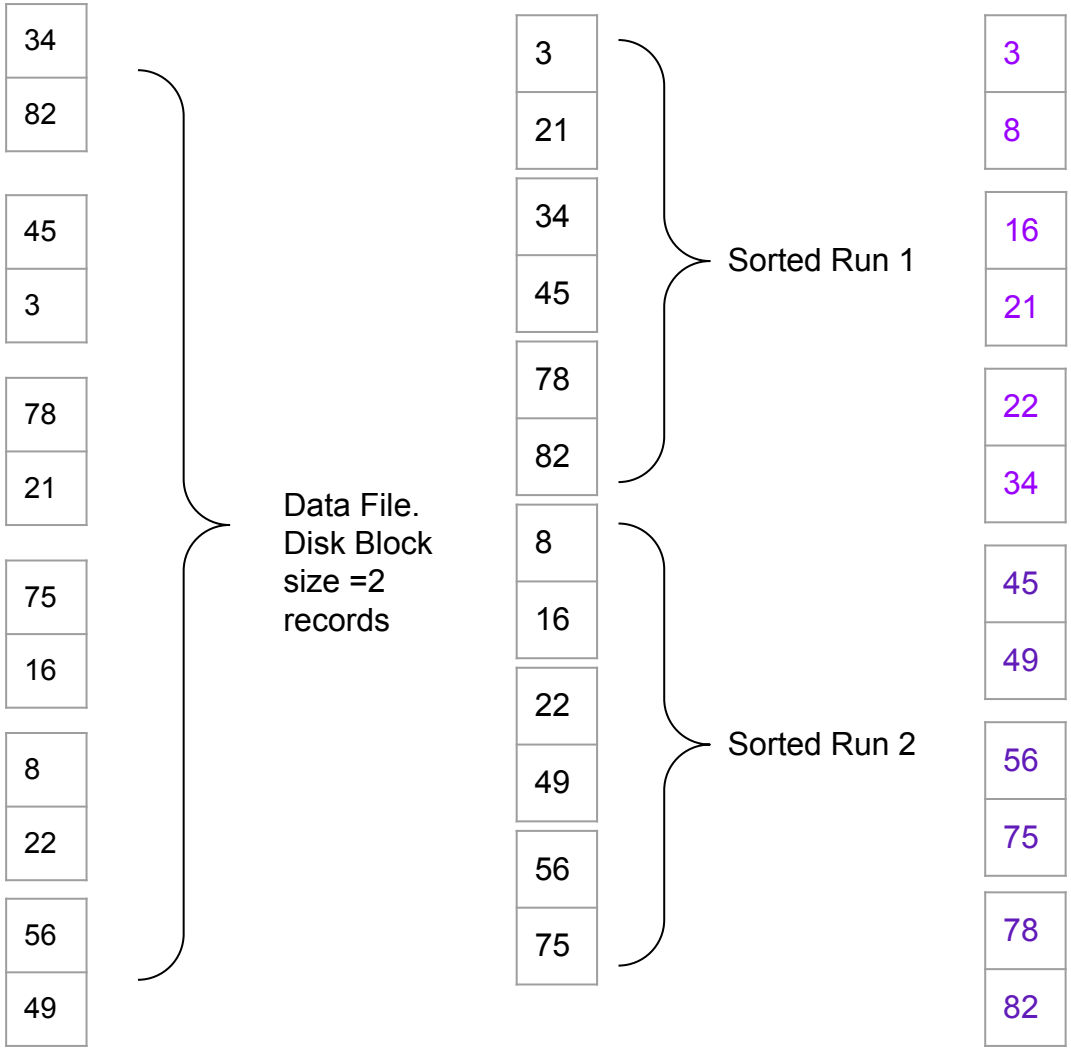


CASE: M > #Runs

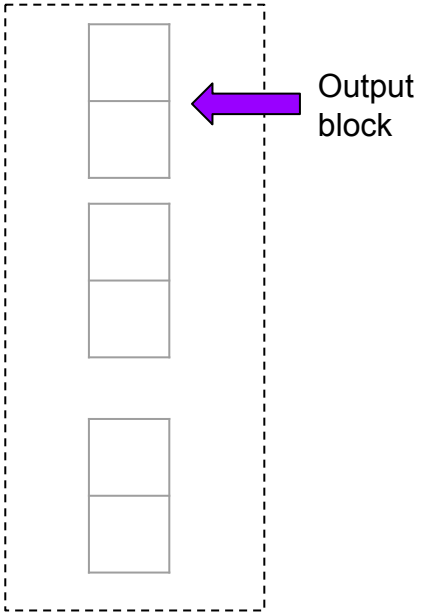


Merge the Runs. Bring in one block at a time from the runs and keep writing the output block.

A single N-way merge operation is enough. No passes needed



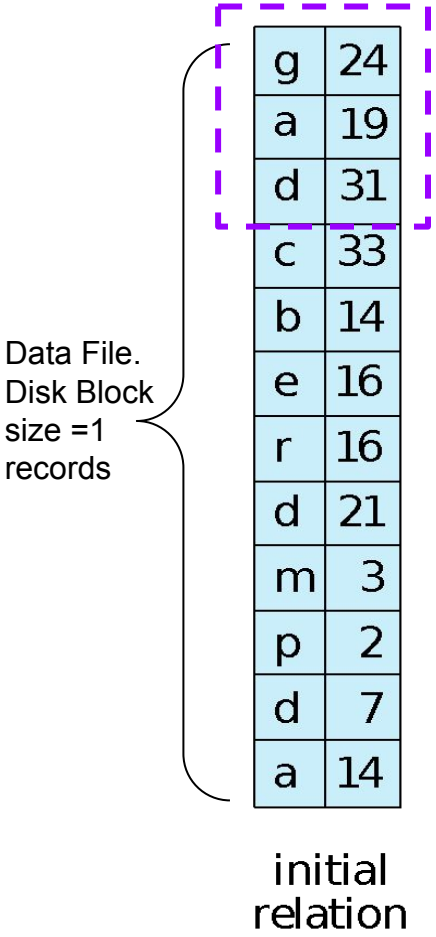
CASE: M > #Runs

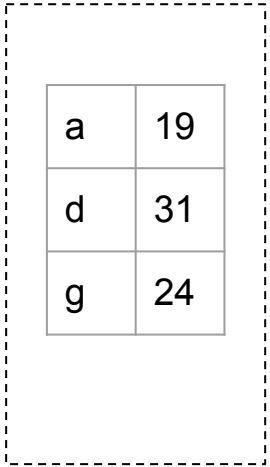


Merge the Runs. Bring in one block at a time from the runs and keep writing the output block.

A single N-way merge operation is enough. No passes needed

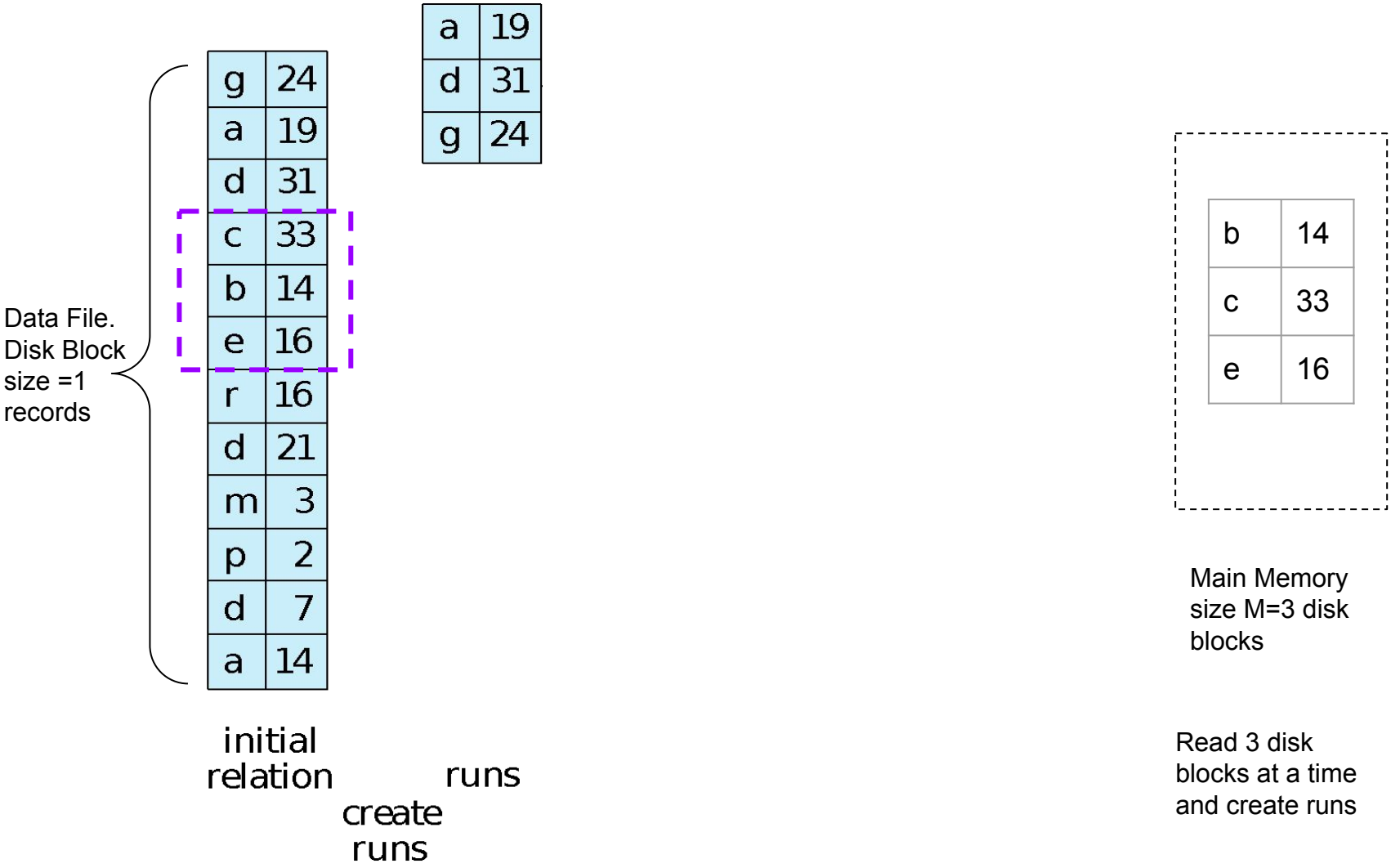
CASE 2: M =< #Runs

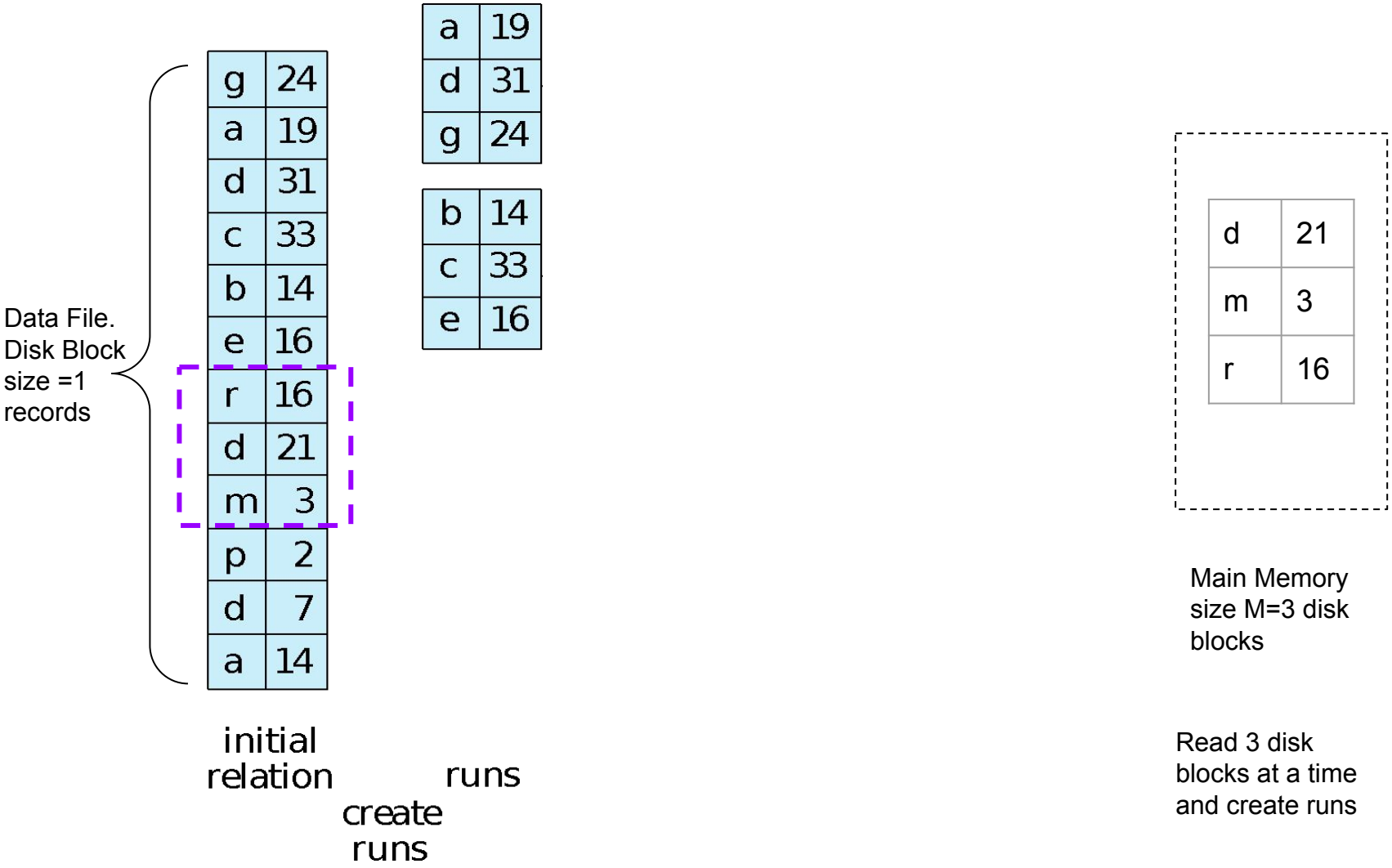


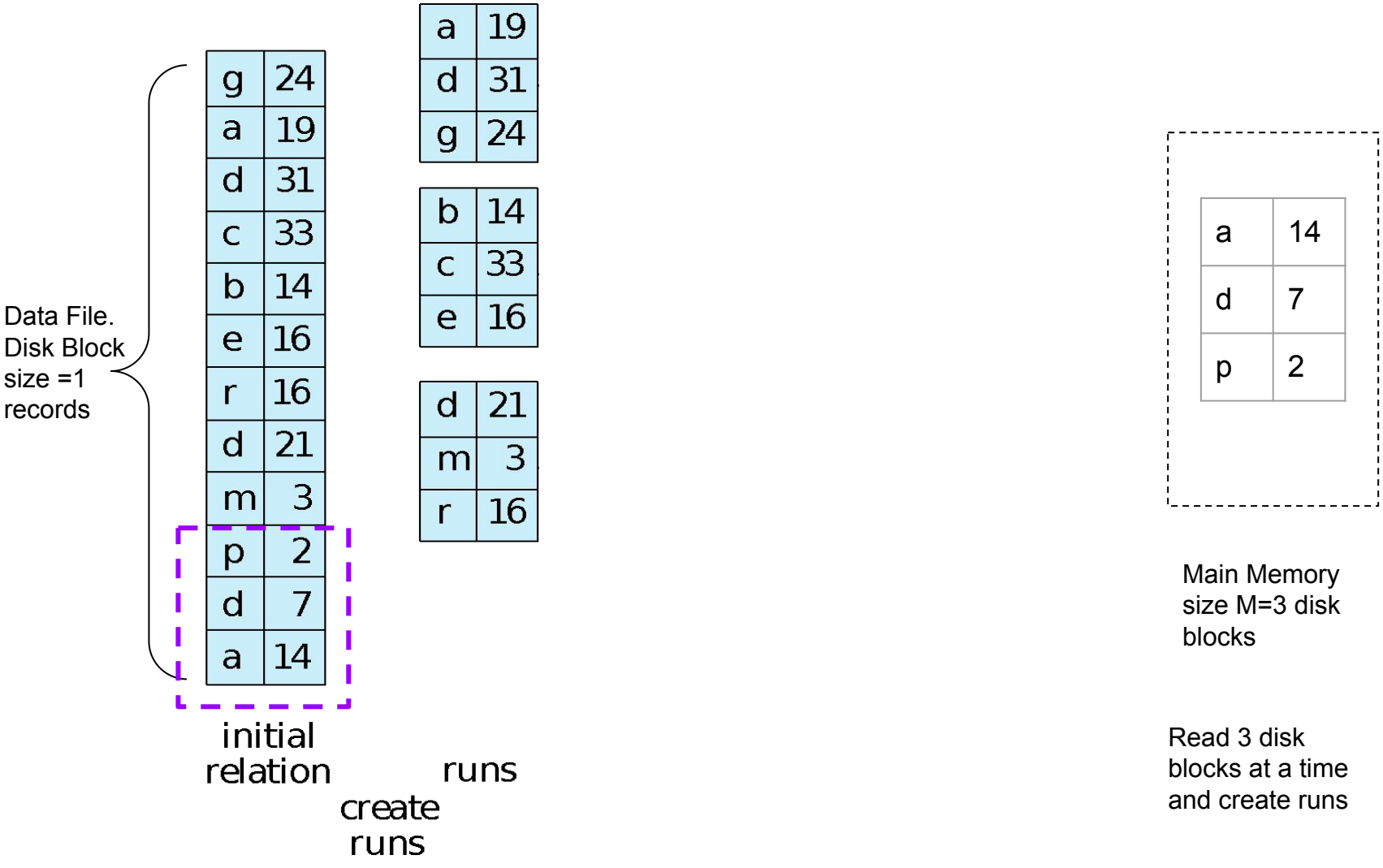


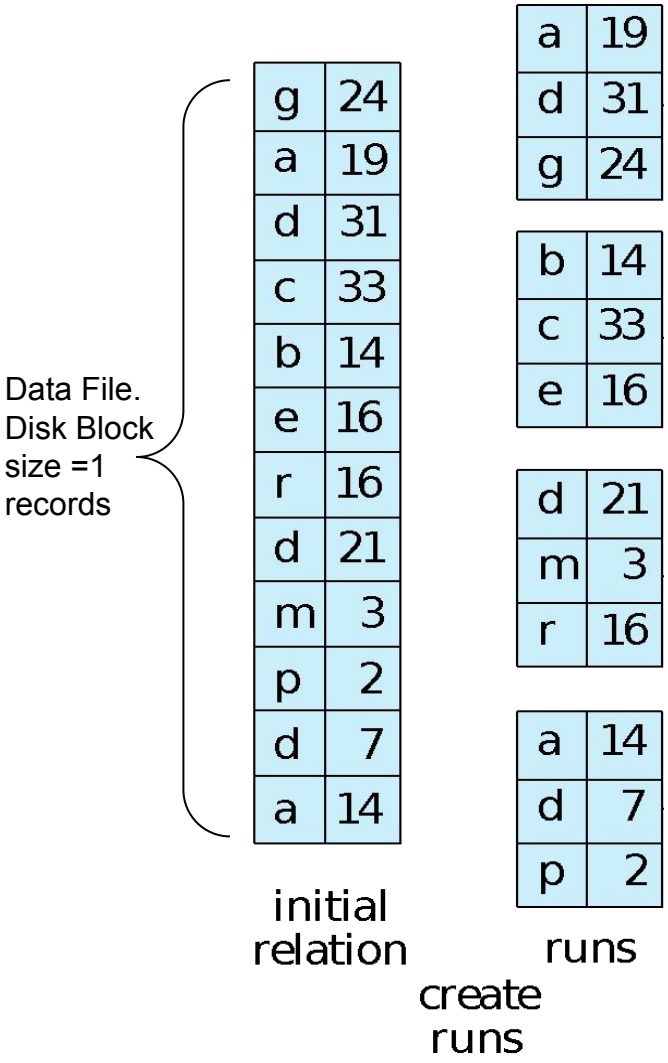
Main Memory
size M=3 disk
blocks

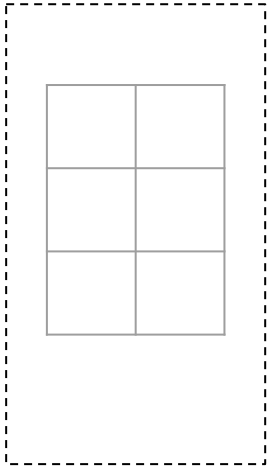
Read 3 disk
blocks at a time
and create runs


Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
runs
b 14
c 33
e 16
Main Memory
size M=3 disk
blocks
Read 3 disk
blocks at a time
and create runs


Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
Main Memory
size M=3 disk
blocks
initial
relation
create
runs
runs
Read 3 disk
blocks at a time
and create runs


Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
runs
a 14
d 7
p 2
Main Memory
size M=3 disk
blocks
Read 3 disk
blocks at a time
and create runs

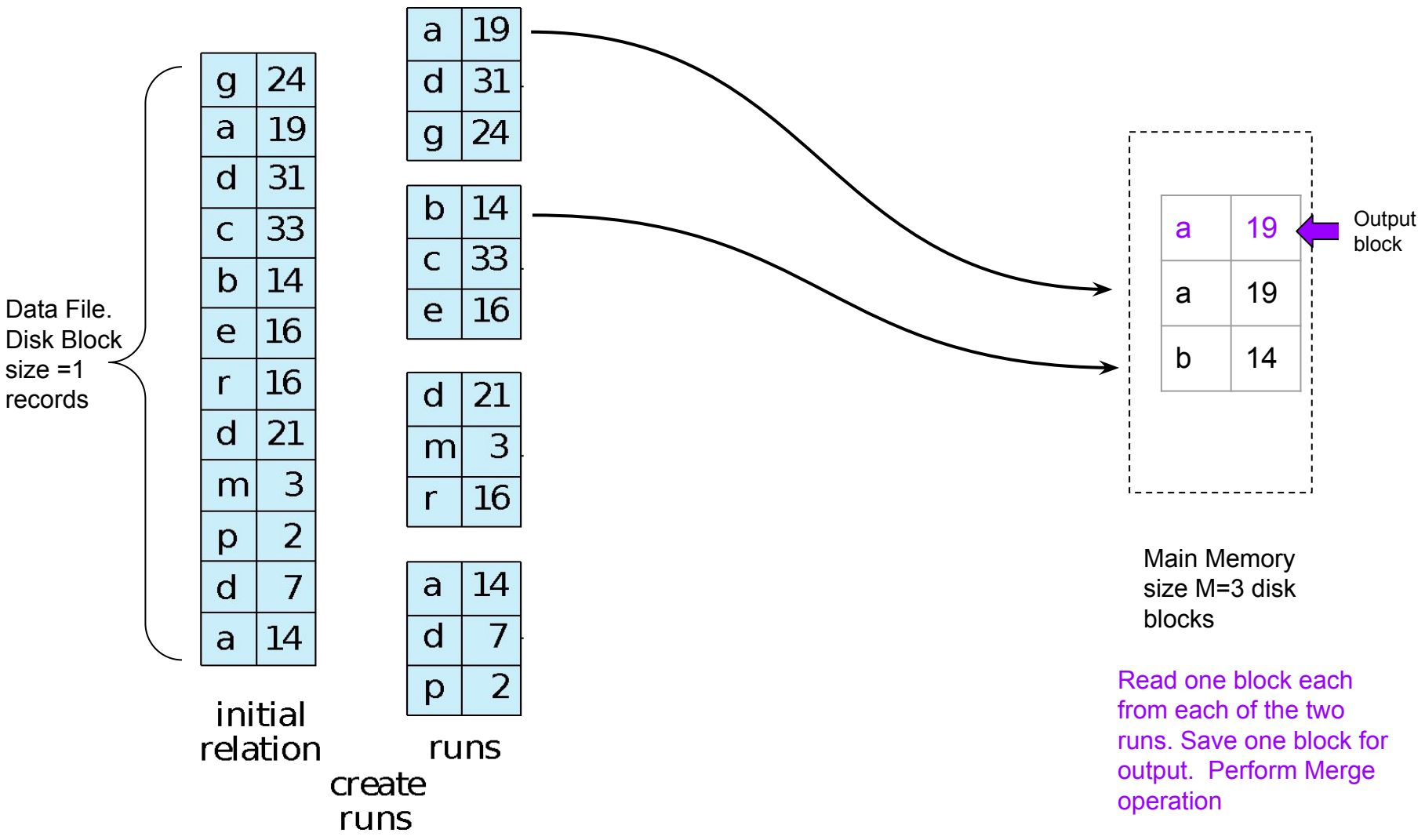


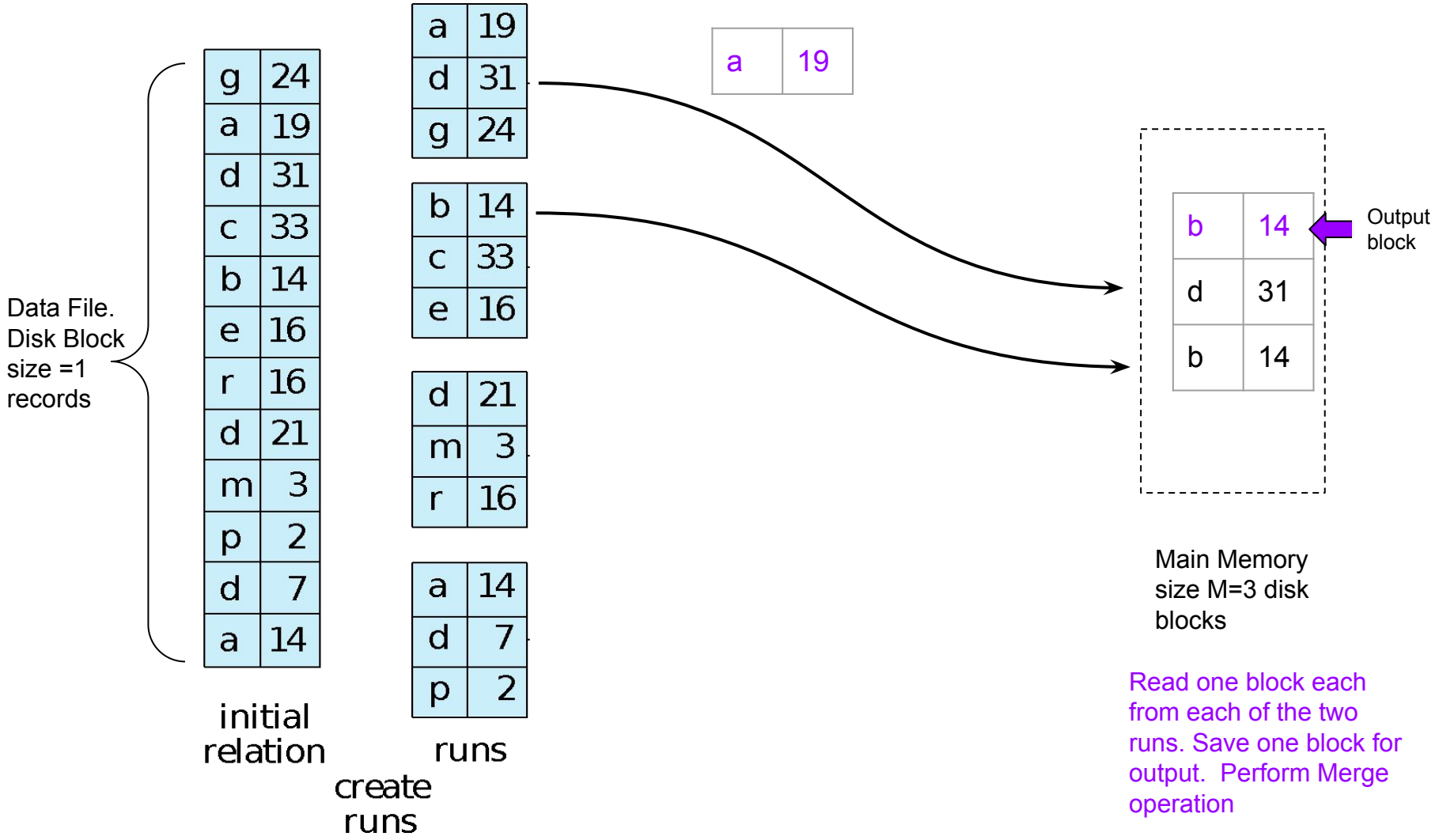


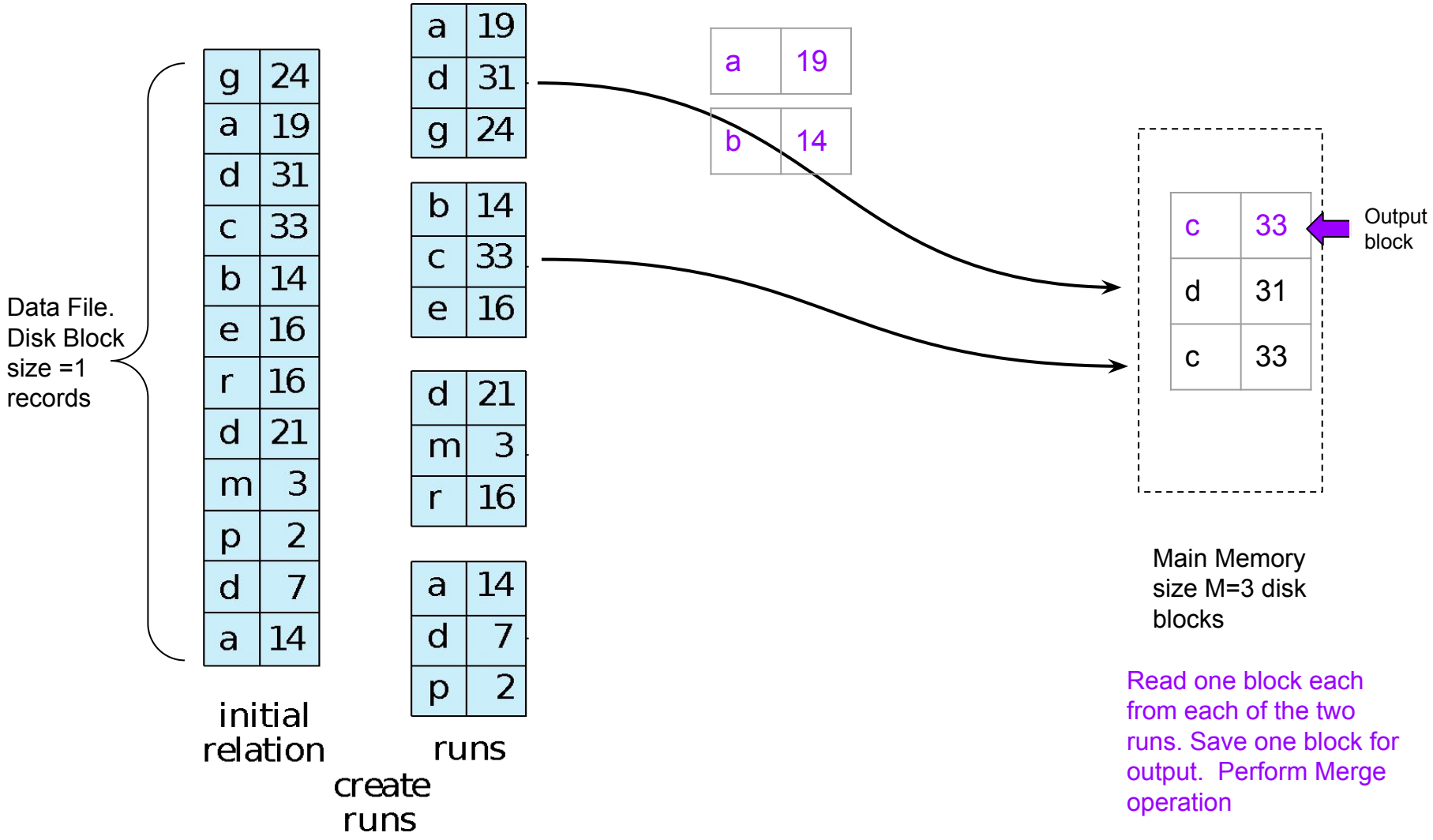
Main Memory
size M=3 disk
blocks

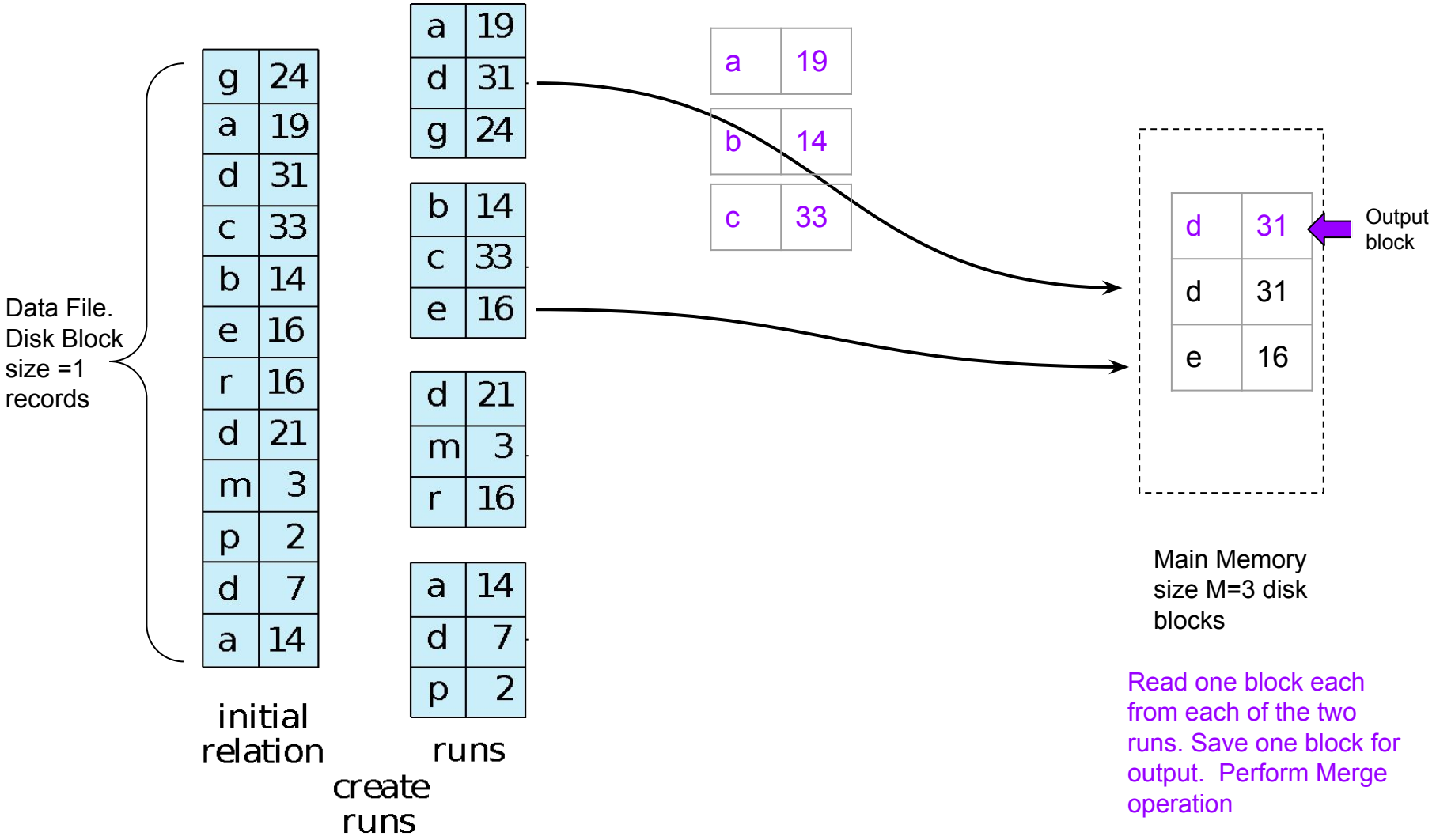
In general, we can merge Min(#runs, M-1) runs at a time. As M=3 in our case, we merge 2 runs at a time.

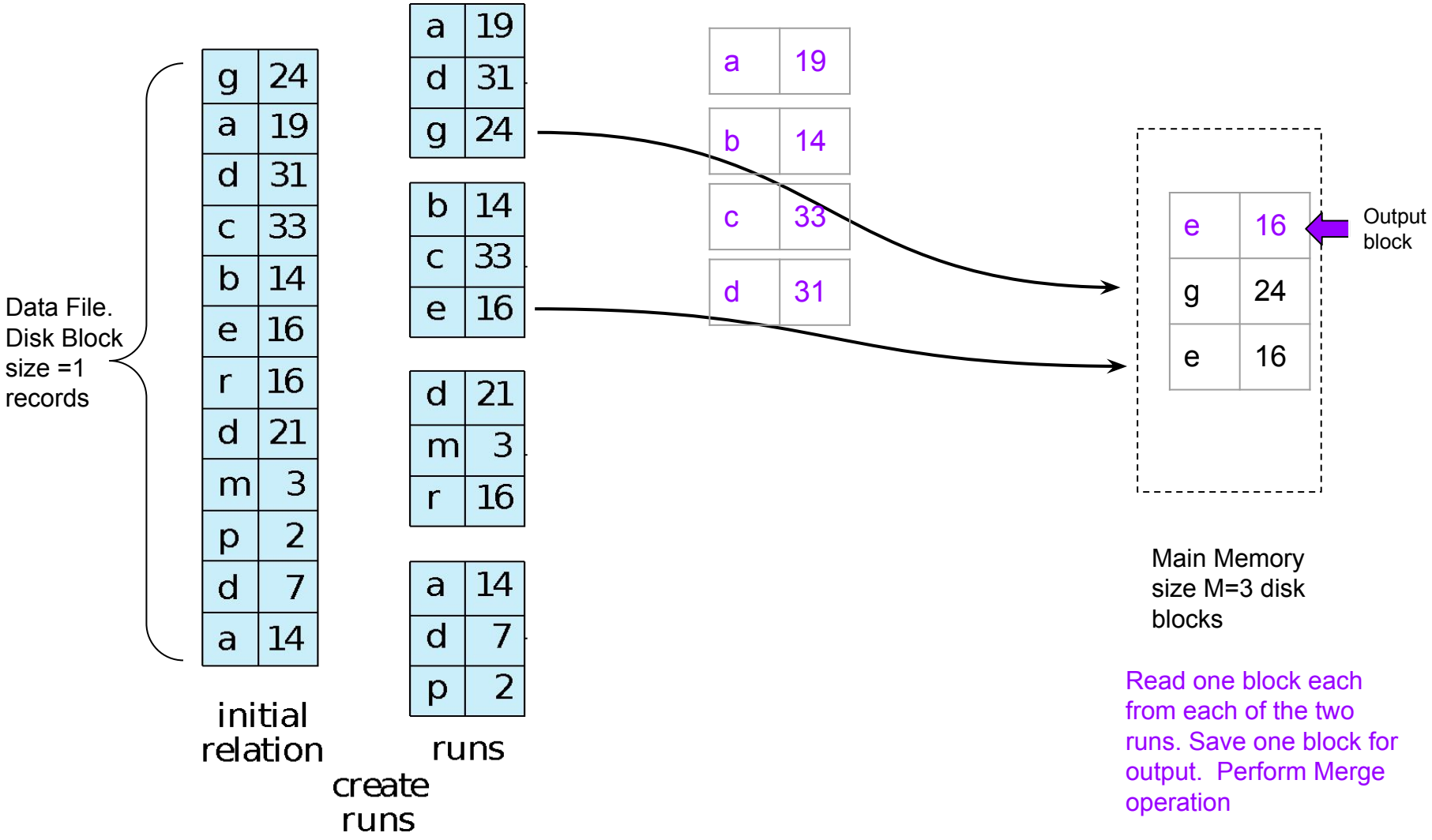
We merge two runs at a time. One block in main memory is saved for output result. Each Run spans across blocks

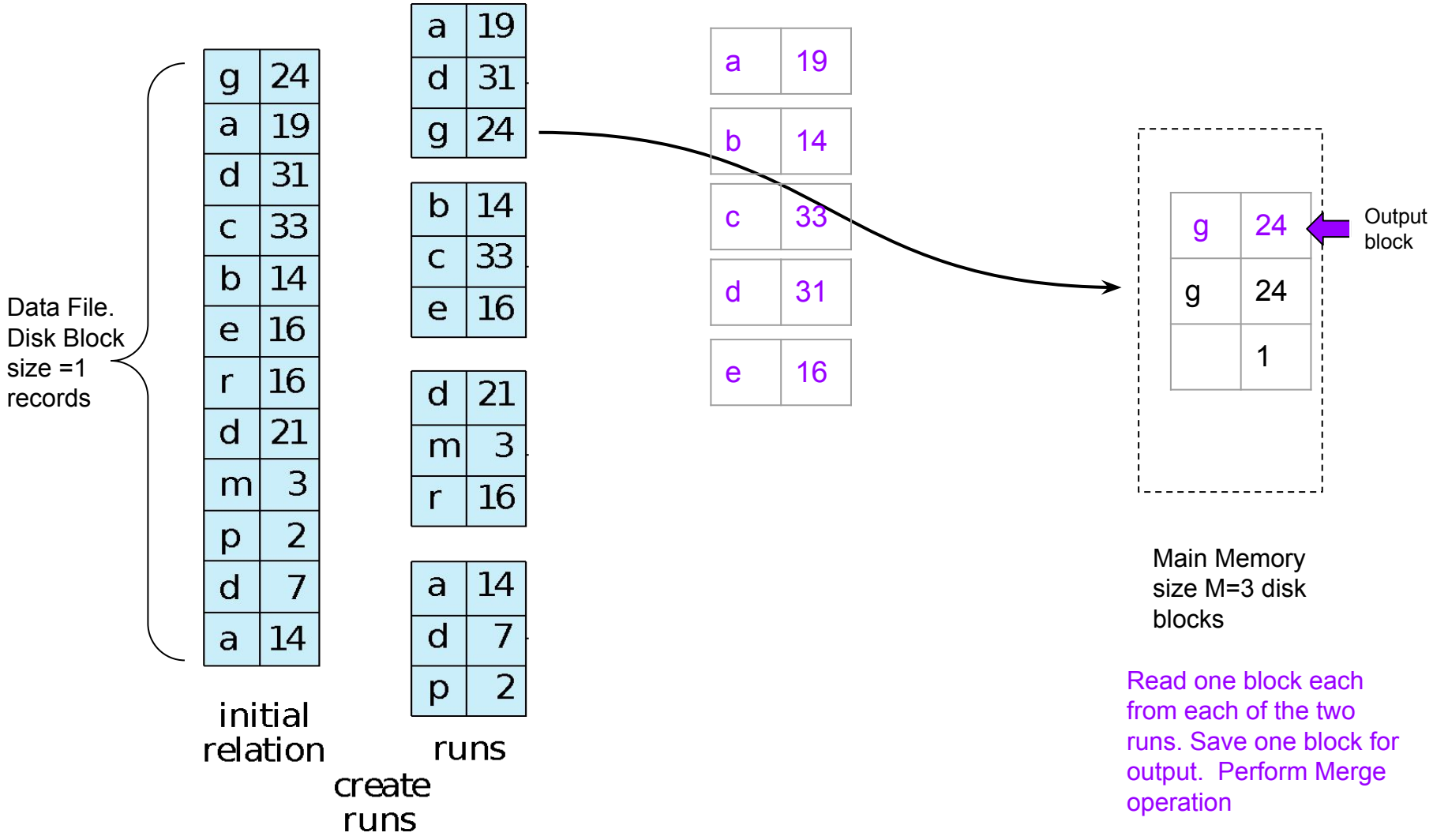

Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
a 14
d 7
p 2
runs
a 19
a 19
b 14
Output
block
Main Memory
size M=3 disk
blocks
Read one block each
from each of the two
runs. Save one block for
output. Perform Merge
operation

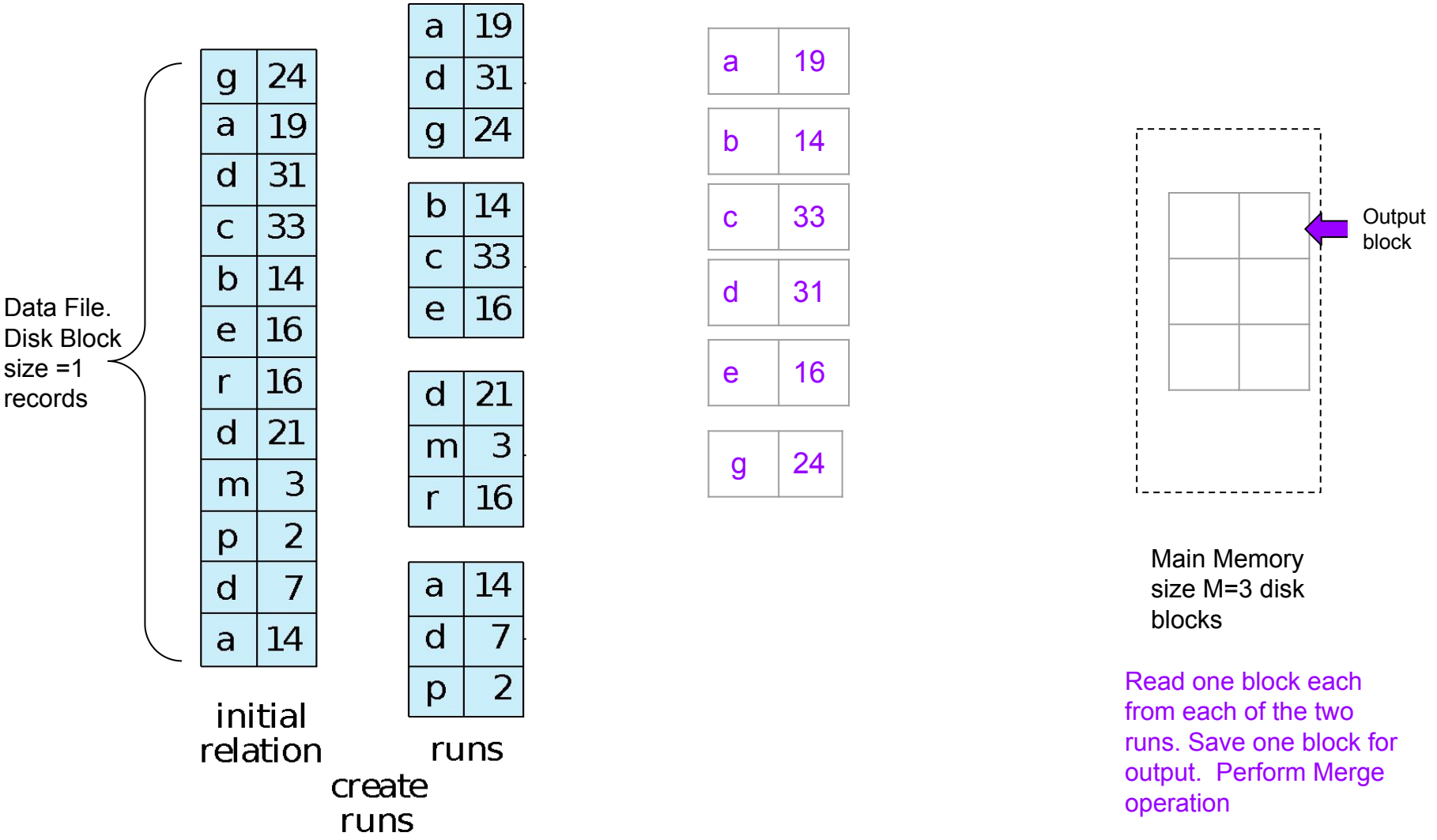

Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
a 14
d 7
p 2
runs
a 19
b 14
d 31
b 14
Output
block
Main Memory
size M=3 disk
blocks
Read one block each
from each of the two
runs. Save one block for
output. Perform Merge
operation

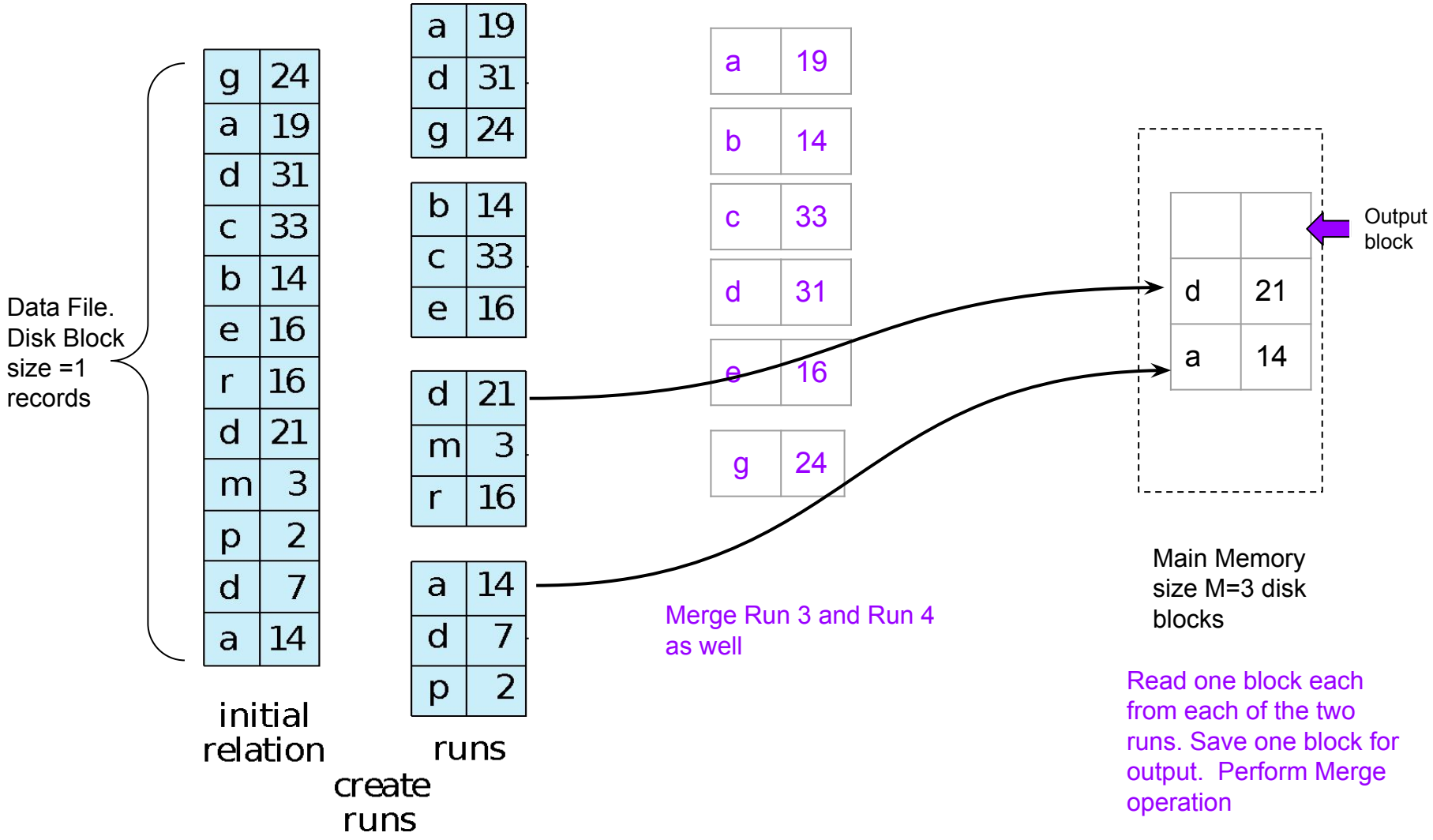

Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
a 14
d 7
p 2
runs
a 19
b 14
c 33
d 31
c 33
Output
block
Main Memory
size M=3 disk
blocks
Read one block each
from each of the two
runs. Save one block for
output. Perform Merge
operation

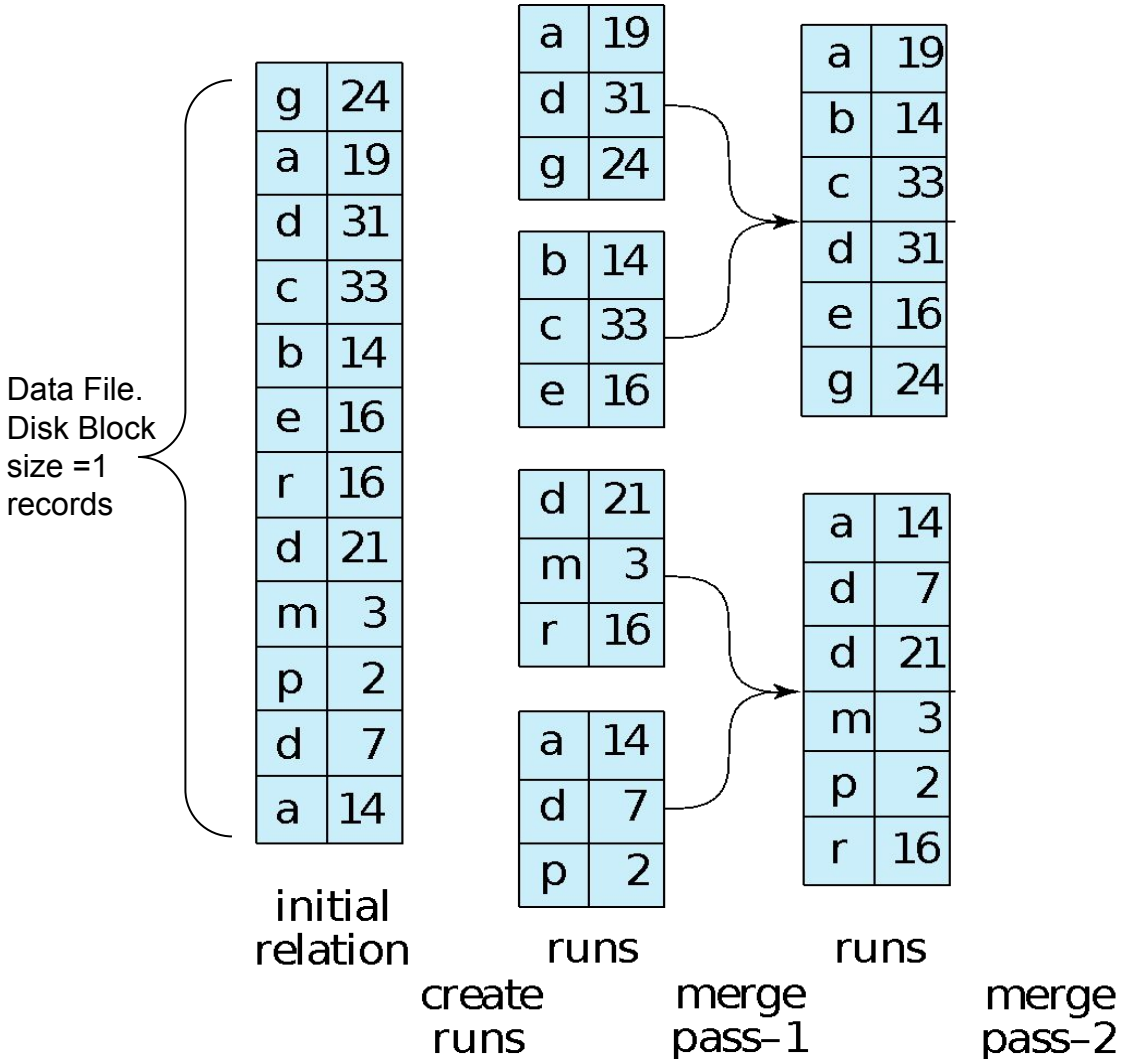

Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
a 14
d 7
p 2
runs
a 19
b 14
c 33
d 31
d 31
e 16
Output
block
Main Memory
size M=3 disk
blocks
Read one block each
from each of the two
runs. Save one block for
output. Perform Merge
operation

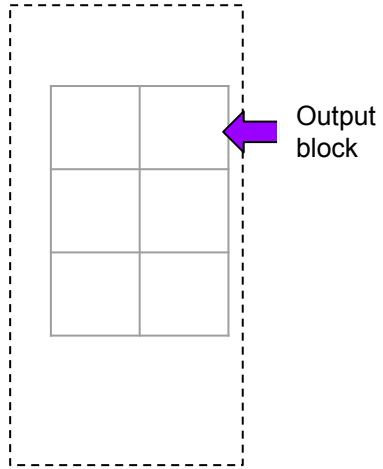

Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
a 14
d 7
p 2
runs
a 19
b 14
c 33
d 31
e 16
g 24
e 16
Output
block
Main Memory
size M=3 disk
blocks
Read one block each
from each of the two
runs. Save one block for
output. Perform Merge
operation


Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
a 14
d 7
p 2
runs
a 19
b 14
c 33
d 31
e 16
g 24
g 24
1
Output
block
Main Memory
size M=3 disk
blocks
Read one block each
from each of the two
runs. Save one block for
output. Perform Merge
operation


Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
a 14
d 7
p 2
runs
a 19
b 14
c 33
d 31
e 16
g 24
Output
block
Main Memory
size M=3 disk
blocks
Read one block each
from each of the two
runs. Save one block for
output. Perform Merge
operation

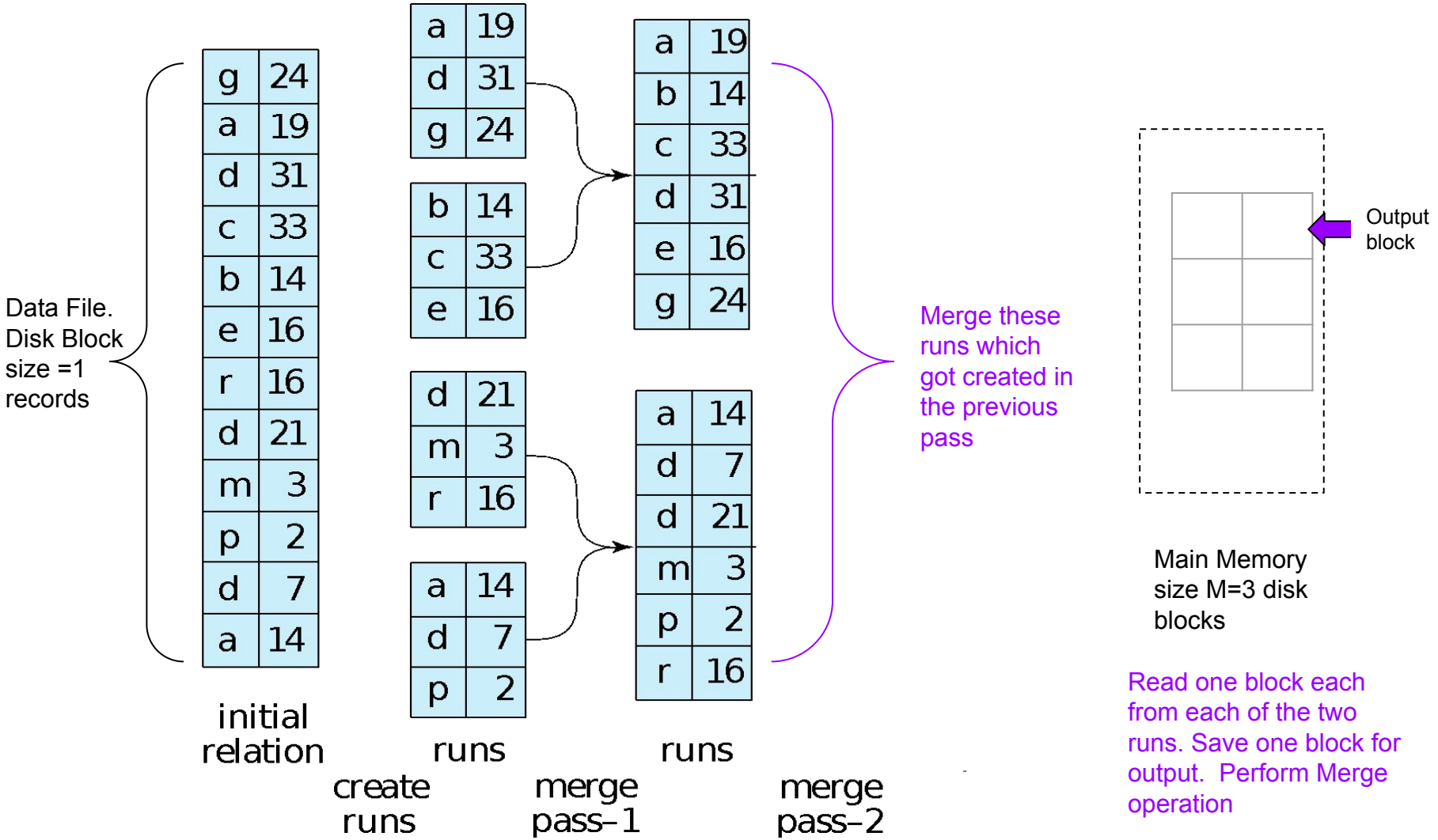

Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
a 14
d 7
p 2
runs
a 19
b 14
c 33
d 31
e 16
g 24
Merge Run 3 and Run 4
as well
Output
block
d 21
a 14
Main Memory
size M=3 disk
blocks
Read one block each
from each of the two
runs. Save one block for
output. Perform Merge
operation

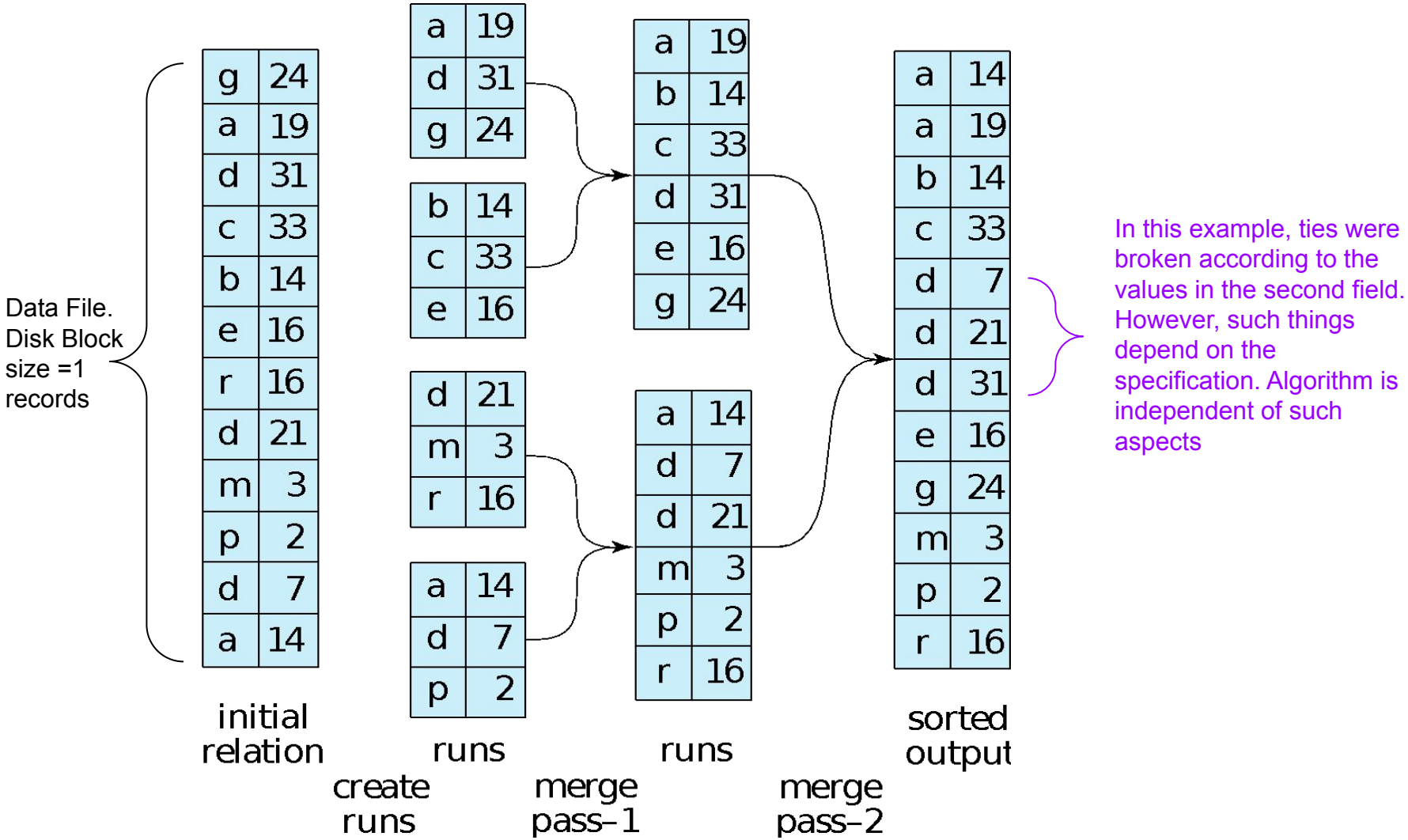




Main Memory
size M=3 disk
blocks

Read one block each from each of the two runs. Save one block for output. Perform Merge operation


Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
create
runs
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
a 14
d 7
p 2
runs
merge
pass–1
a 19
b 14
c 33
d 31
e 16
g 24
a 14
d 7
d 21
m 3
p 2
r 16
runs
merge
pass–2
Merge these
runs which
got created in
the previous
pass
Output
block
Main Memory
size M=3 disk
blocks
Read one block each
from each of the two
runs. Save one block for
output. Perform Merge
operation


Data File.
Disk Block
size =1
records
g 24
a 19
d 31
c 33
b 14
e 16
r 16
d 21
m 3
p 2
d 7
a 14
initial
relation
a 19
d 31
g 24
b 14
c 33
e 16
d 21
m 3
r 16
a 14
d 7
p 2
runs
create
runs
a 19
b 14
c 33
d 31
e 16
g 24
a 14
d 7
d 21
m 3
p 2
r 16
runs
merge
pass–1
a 14
a 19
b 14
c 33
d 7
d 21
d 31
e 16
g 24
m 3
p 2
r 16
sorted
output
merge
pass–2
In this example, ties were broken according to the values in the second field. However, such things depend on the specification. Algorithm is independent of such aspects