Iterators, Relational Operators and Joins

R&G Chapters 12 & 14

Recall from Last Lecture

SQL Query

```
SELECT S.name
FROM Reserves R, Sailors S
WHERE R.sid = S.sid
AND R.bid = 100
AND S.rating > 5
```

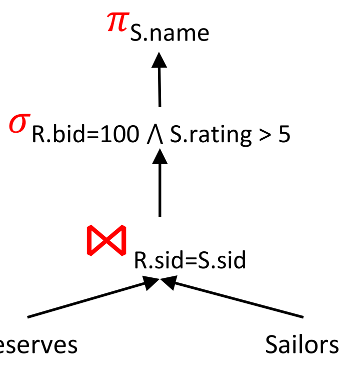
Query Parser & Optimizer

Relational Algebra

$$\pi_{S.name}(\sigma_{bid=100 \land rating>5}(\text{Reserves} \bowtie_{R.sid=S.sid} \text{Sailors}))$$

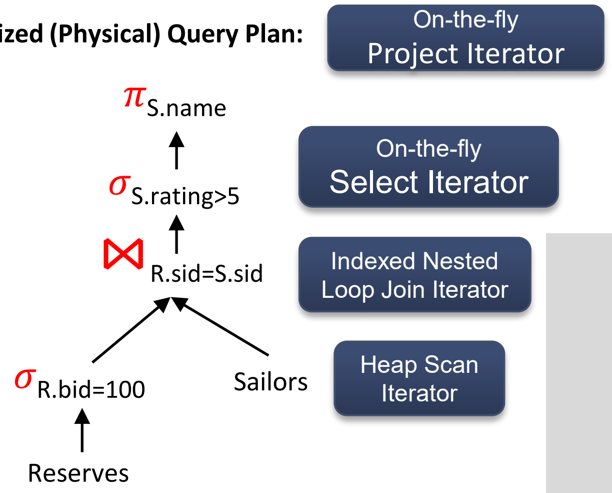
Equivalent to…

(Logical) Query Plan:



But actually will produce…

Optimized (Physical) Query Plan:



Operator Code

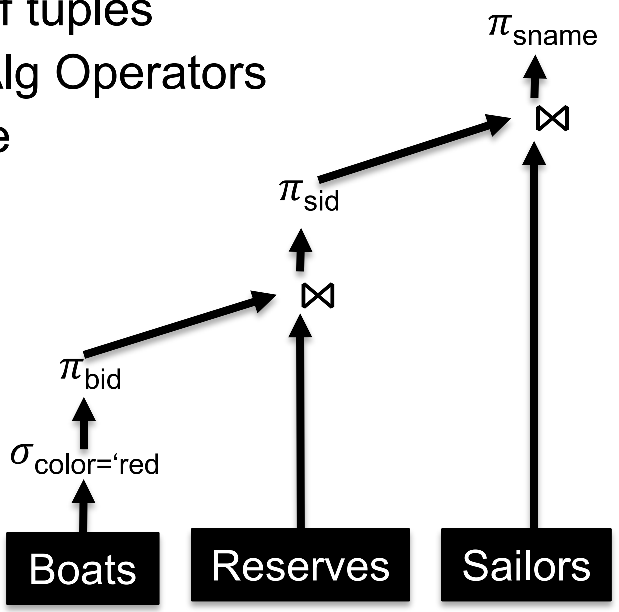
B+-Tree Indexed Scan Iterator

Relational Operators and Query Plans

$\pi_{sname}(\pi_{sid}(\pi_{bid}(\sigma_{color='red'}(Boats)) \bowtie Res) \bowtie Sailors)$

- Query plan
  - Edges encode “flow” of tuples
  - Vertices = Relational Alg Operators
  - Source vertices = table access operators …

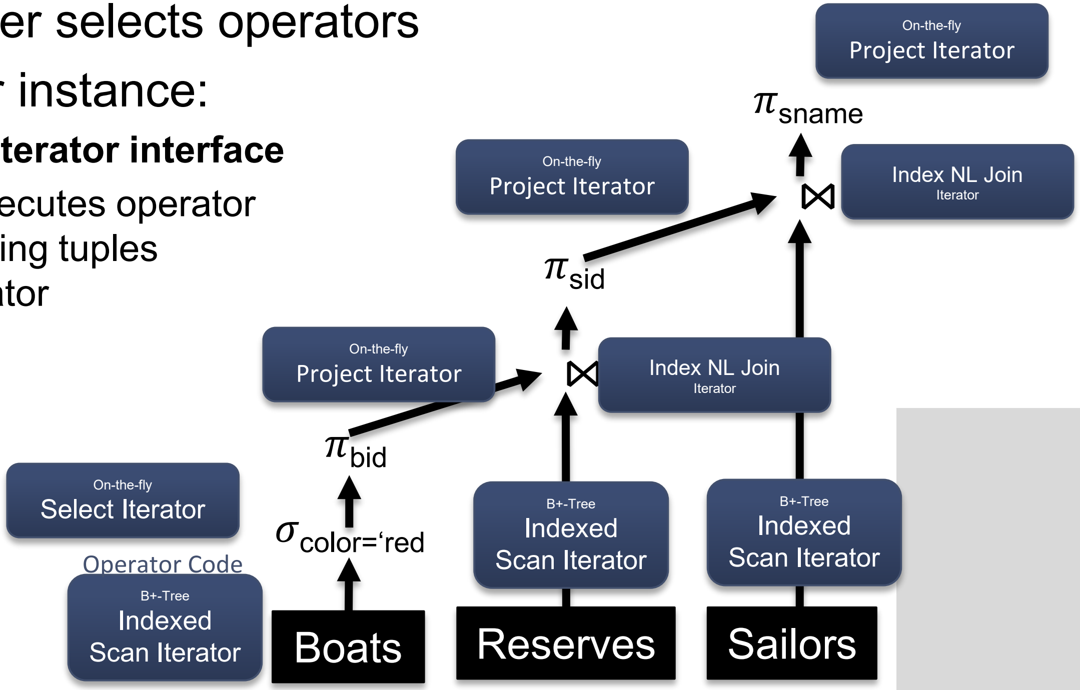
- Also called dataflow graph



Query Executor Instantiates Operators

$$\pi_{sname}(\pi_{sid}(\pi_{bid}(\sigma_{color='red'}(Boats)) \bowtie Res) \bowtie Sailors)$$

- Query optimizer selects operators
- Each operator instance:
  - Implements **iterator interface**
  - Efficiently executes operator logic forwarding tuples to next operator



On-the-fly Project Iterator
π_{sname}
Index NL Join Iterator
$\bowtie$
On-the-fly Project Iterator
π_{sid}
On-the-fly Project Iterator
$\bowtie$
Index NL Join Iterator
π_{bid}
On-the-fly Select Iterator
$\sigma_{color='red}$
Operator Code
B+-Tree Indexed Scan Iterator
Boats
B+-Tree Indexed Scan Iterator
Reserves
B+-Tree Indexed Scan Iterator
Sailors

Iterator Interface

The relational operators implemented as subclasses of the class Iterator:

```
abstract class iterator {
      void setup(List<Iterator> inputs);
      void init(args);
      tuple next();
      void close();
 }
```

Notes:

- **Pull-**based computation model
  - e.g., Console calls **init** and **next** which propagates down graph
  - *init/next* can result in either *streaming ("on-the-fly")* or *blocking ("batch")* algorithm**:**
    - streaming: small, constant amount of work per call
    - blocking: does not produce output until it consumes its entire input!
- **Encapsulation**: any iterator can be input to any other!
- **State:** iterators may maintain substantial "internal" state
  - e.g., hash tables, running counts, large sorted files …

Example: Select (on-the-fly)

- init(predicate):

```
child.init()
pred = predicate;
current = NULL;
```

- next():

```
while (current != EOF && !pred(current))
    current = child.next();
}
return current;
```

- close():

```
child.close()
```

Example: Heap Scan

- init(relation):

```
heap = open heap file for this relation;
cur_page = heap.first_page(); // first page
cur_slot = cur_page.first_slot(); // first slot on that page
```

- next():

```
if (cur_page == NULL) return EOF;  // End Of Fun
current = [cur_page, cur_slot]; // we will return this recordId
// advance the slot
cur_slot = cur_slot.next();
if (cur_slot == NULL) {
  // advance to next page, first slot
  cur_page = cur_page.next();
  if (cur_page != NULL)
     cur_slot = cur_page.first_slot();
}
return current;
```

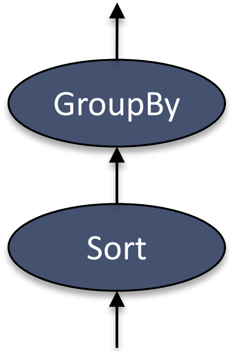
- close():

```
heap.close()
```

Example: Sort (2-pass)

```
• init(keys):                    // all of pass 0 in init, a blocking call
    child.init()
    repeatedly call child.next() and generate the sorted runs on disk, until child gives EOF
    // set up for pass 1, assumes enough buffers to merge
    open each sorted run file and load into input buffer for pass 1
• next():                  // pass 1 (assumes enough buffers to merge)
    output = min tuple across all buffers
    if min tuple was last one in its buffer, fetch next page from that run into buffer
    return output (or EOF -- “End of Fun” -- if no tuples remain)
• close():
    deallocate the runs files
    child.close()
```

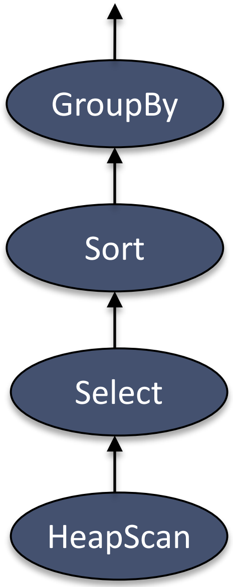
Example: Group By on Sorted input



agg_type	state	init	merge(x)	final
COUNT	count	0	count ++	count
SUM	sum	0	sum += x	sum
AVG	[count, sum]	[0, 0]	[count++, sum+=x]	sum / count
MIN	min	+infinity	min > x ? x : min	min

```
• init(group_keys, aggs):
    child.init()
    cur_group = NULL;
• next():
    result = NULL
    do {
            tup = child.next();
            if (group(tup) != cur_group) { // New group!
                if (cur_group != NULL)     // Form a result for current group
                    result = [cur_group, final() of all aggs]
                cur_group = group(tup);
                call init() on all the aggs
            }
            call merge(tup) on all the aggs
    } while (!result);
    return result;
• close():
    child.close()
```

A Full Query Plan



- A Query Plan is Single-threaded!
- Trace the calls:
  - Call init() on the root GroupBy
    - How does init() recurse down the chain and return?
  - call next() on root
    - How does next() recurse down the chain and return a tuple?
- Note how the blocking operator (sort) interacts with the other, streaming operators
- Note how we don’t store operator output on disk; tuples stream through the plan’s call stack
  - Some operators like Sort use disk internally

Join Operators

R&G 14.4

Schema for Examples

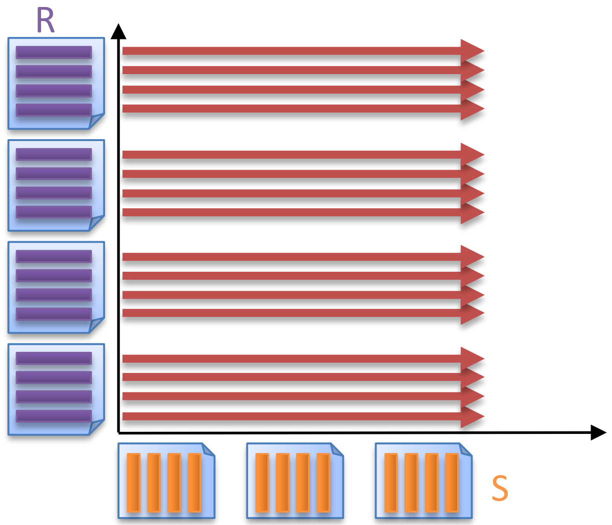
- Cost Notation
  - [R] : the number of pages to store R
  - p_R : number of records per page of R
  - |R| : the cardinality (number of records) of R
    - $|R| = p_R * [R]$

- Reserves (sid: int, bid: int, day: date, rname: string)
  - [R]=1000, p_R=100, |R| = 100,000
- Sailors (sid: int, sname: string, rating: int, age: real)
  - [S]=500, p_S=80, |S| = 40,000

Simple Nested Loops Join

```
foreach record r in R do
    foreach record s in S do
        if θ(ri, sj)  then add <ri, sj> to result buffer
```

Note: for simplicity we do not present iterator implementations for the join algorithms.



$[R]=1000,\ p_R=100,\ |R| = 100{,}000$

$[S]=500,\ p_S=80,\ |S| = 40{,}000$

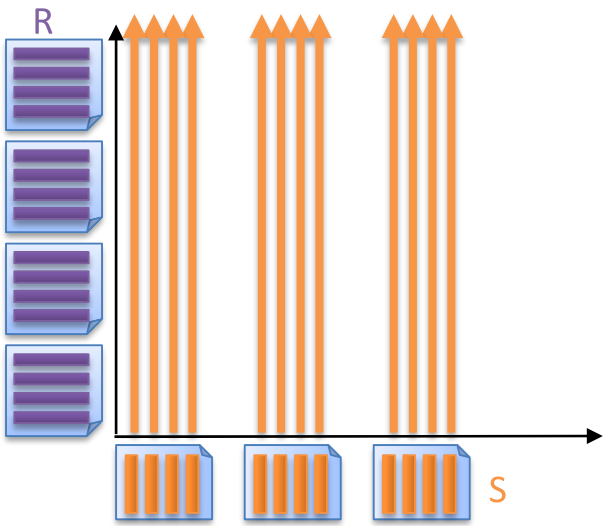
Cost:

$[R] + |R|[S]$

$= 50{,}001{,}000$

Changing the Join Order

```
foreach record s in S do
    foreach record r in R do
        if θ(ri, sj)  then add <ri, sj> to result buffer
```



$[R]=1000$, $p_R=100$, $|R| = 100{,}000$

$[S]=500$, $p_S=80$, $|S| = 40{,}000$

Cost:

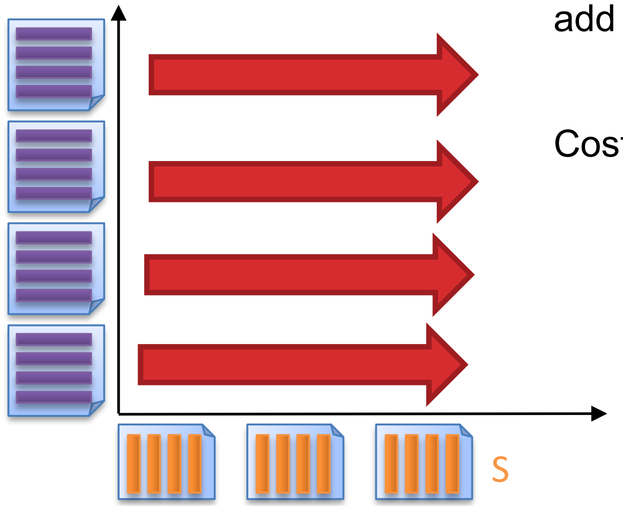
$[S] + |S|[R]$

$= \quad 40{,}000{,}500$

vs. 50,001,000

Page Nested Loop Join

```
for each rpage in R:
    for each spage in S:
        for each rtuple in rpage:
            for each stuple in spage:
                if join_condition(rtuple, stuple):
                    add <rtuple, stuple> to result buffer
```

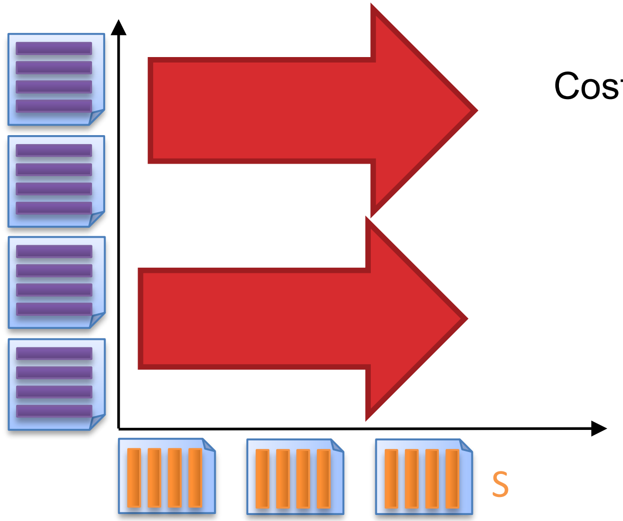


Cost $= [R] + ([R] * [S])$
$= 501{,}000$

“Chunk”

~~“Block”~~ Nested Loop Join

```
for each rchunk of B-2 pages of R:
    for each spage of S:
        for all matching tuples in spage and rchunk:
            add <rtuple, stuple> to result buffer
```



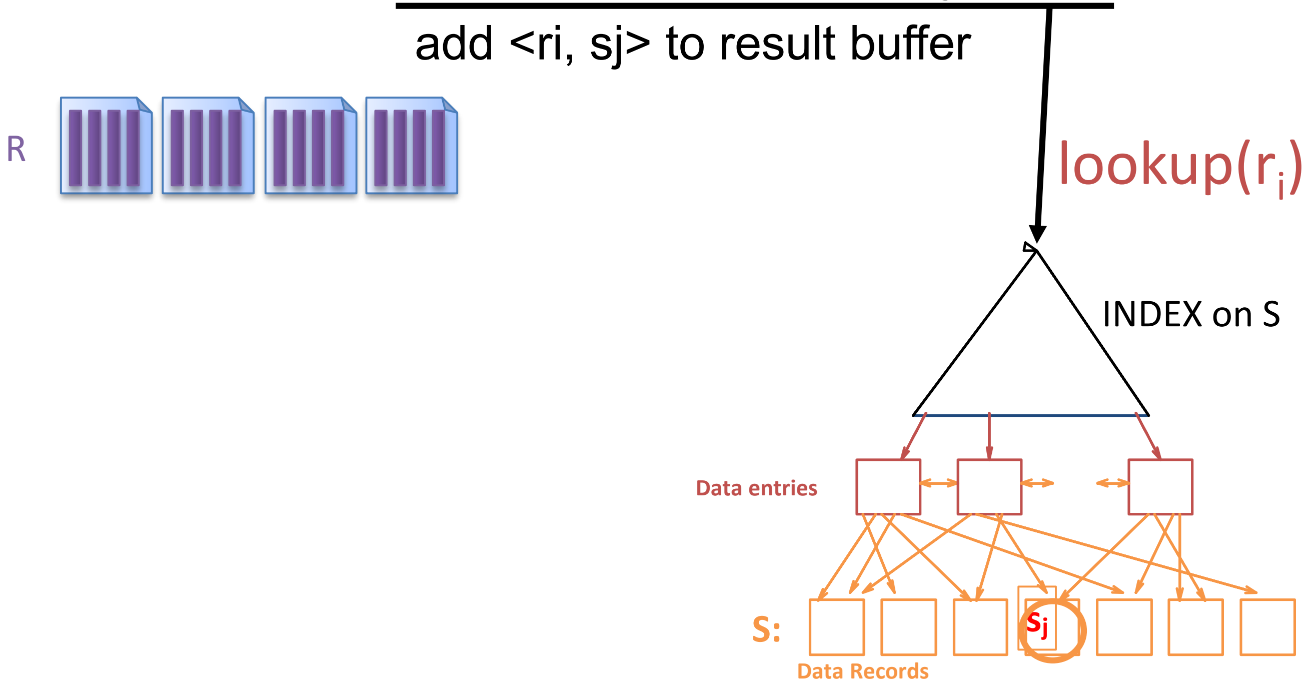
Cost = [R] + ⌈ [R]/(B-2) ⌉ * [S]

= 1000 + ⌈ 1000/(B-2) ⌉ * 500

= 6,000 for B=102 (~100x better than Page NL!)

Index Nested Loops Join

```
foreach tuple r in R do
        foreach tuple s in S where ri == sj  do
                add <ri, sj> to result buffer
```



Index Nested Loops Join Cost

```
foreach tuple r in R do
          foreach tuple s in S where ri == sj  do
                    add <ri, sj> to result
```

Cost = [R] + |R| * cost to find matching S tuples

- If index uses Alt. 1 → cost to traverse tree from root to leaf. (e.g., 2-4 IOs)
- For Alt. 2 or 3:
  - Cost to lookup RID(s); typically 2-4 IOs for B+Tree.
  - Cost to retrieve records from RID(s)
    - Clustered index: 1 I/O per *page of matching S tuples*.
    - Unclustered: up to 1 I/O per matching S tuple

Index Nested Loops Join Cost, Part 2

- Reserves (*sid*: int, *bid*: int, *day*: date, *rname*: string)
  - [R]=1000, p_R=100, |R| = 100,000
- Sailors (***sid***: int, *sname*: string, *rating*: int, *age*: real)
  - [S]=500, p_S=80, |S| = 40,000
  - Index on sid

Index Nested Loops Join Cost, Part 3

- Unclustered Cost(R,S) = [R] + |R| * (Search + # matching tuples)
- Clustered Cost(R,S): [R] + |R| * (Search + # of matching pages)
- Here, sid is the primary key for Sailors, so there is exactly one matching sailor for each tuple in R
- Unclustered B+-Tree height 2 (3 I/Os from root to leaf):
  – R⋈S: 1000 + (100,000)*(3 + 1) = 401,000

- Clustered B+-tree height 2 (3 I/Os from root to leaf)
  – R⋈S: 1000 + (100,000)*(3 +1) = 401,000

Sort-Merge Join

- Requires equality predicate:
  - Equi-Joins & Natural Joins
- Two Stages:
  - Sort tuples in R and S by join key
    - All tuples with same key in consecutive order
    - Input might already be sorted … why?
  - Join Pass: Merge-scan the sorted partitions and emit tuples that match

Sort-Merge Join, Part 1

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

Sort-Merge Join, Part 2

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

Sort-Merge Join, Part 3

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

Sort-Merge Join, Part 4

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

Sort-Merge Join, Part 5

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of “block” of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

Sort-Merge Join, Part 6

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

Sort-Merge Join, Part 7

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

Sort-Merge Join, Part 8

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103

Sort-Merge Join, Part 9

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103

Sort-Merge Join, Part 10

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103

Sort-Merge Join, Part 11

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103

Sort-Merge Join, Part 12

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103

Sort-Merge Join, Part 13

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 14

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 15

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of “block” of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 16

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 17

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 18

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 19

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of “block” of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 20

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of “block” of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 21

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

	sid	sname
	22	dustin
	28	yuppy
➡	31	lubber
	31	lubber2
	44	guppy
	58	rusty

	sid	bid
	28	103
	28	104
➡	31	101
	31	102
	42	142
	58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 22

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 23

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 24

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 25

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of “block” of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104

Sort-Merge Join, Part 26

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101

Sort-Merge Join, Part 27

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101

Sort-Merge Join, Part 28

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101

Sort-Merge Join, Part 29

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101

Sort-Merge Join, Part 30

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of “block” of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101

Sort-Merge Join, Part 31

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 32

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 33

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of “block” of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 34

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 35

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 36

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 37

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 38

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 39

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 40

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 41

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102

Sort-Merge Join, Part 42

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101

Sort-Merge Join, Part 43

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101

Sort-Merge Join, Part 44

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of “block” of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101

Sort-Merge Join, Part 45

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101

Sort-Merge Join, Part 46

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101

Sort-Merge Join, Part 47

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 48

```
do {
   if (!mark) {
      while (r < s) { advance r }
      while (r > s) { advance s }
      // mark start of "block" of S
      mark = s
   }
   if (r == s) {
      result = <r, s>
      advance s
      return result
   }
   else {
      reset s to mark
      advance r
      mark = NULL
   }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 49

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 50

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 51

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 52

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 53

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
→ 44	guppy
58	rusty

sid	bid
28	103
28	104
→ 31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 54

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

	sid	sname
	22	dustin
	28	yuppy
	31	lubber
	31	lubber2
→	44	guppy
	58	rusty

	sid	bid
	28	103
	28	104
→	31	101
	31	102
	42	142
	58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 55

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
→ 44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
→ 58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 56

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 57

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 57

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 58

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 59

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 60

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 61

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 62

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 63

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 64

```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102

Sort-Merge Join, Part 65

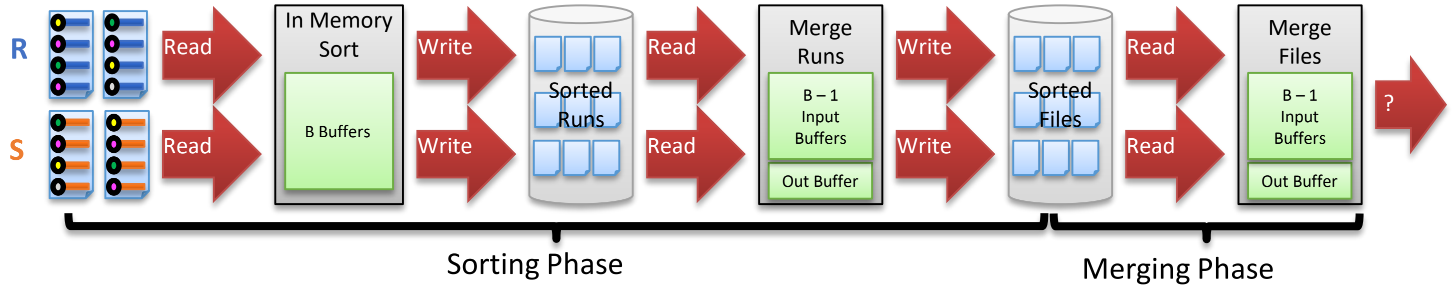
```
do {
  if (!mark) {
    while (r < s) { advance r }
    while (r > s) { advance s }
    // mark start of "block" of S
    mark = s
  }
  if (r == s) {
    result = <r, s>
    advance s
    return result
  }
  else {
    reset s to mark
    advance r
    mark = NULL
  }
}
```

sid	sname
22	dustin
28	yuppy
31	lubber
31	lubber2
44	guppy
58	rusty

sid	bid
28	103
28	104
31	101
31	102
42	142
58	107

sid	sname	bid
28	yuppy	103
28	yuppy	104
31	lubber	101
31	lubber	102
31	lubber2	101
31	lubber2	102
58	rusty	107

Cost of Sort-Merge Join



- Cost: Sort R + Sort S + ([R]+[S])
  - But in worst case, last term could be |R| *[S] (very unlikely!)
  - Q: what is worst case?

- Question: How big does the buffer have to be to sort both R and S in two passes each?

- Suppose buffer B > $\sqrt{(\max([R], [S]))}$
  - Both R and S can be sorted in 2 passes
  - 4*1000 + 4*500 + (1000 + 500) = 7500