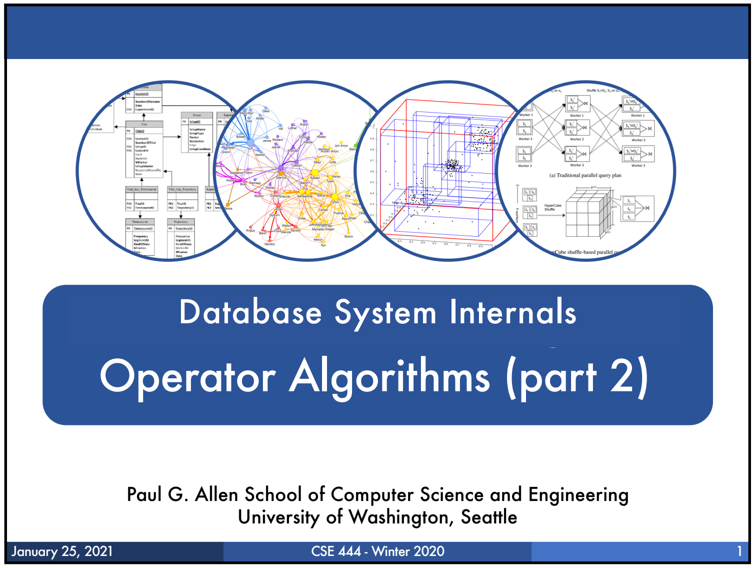
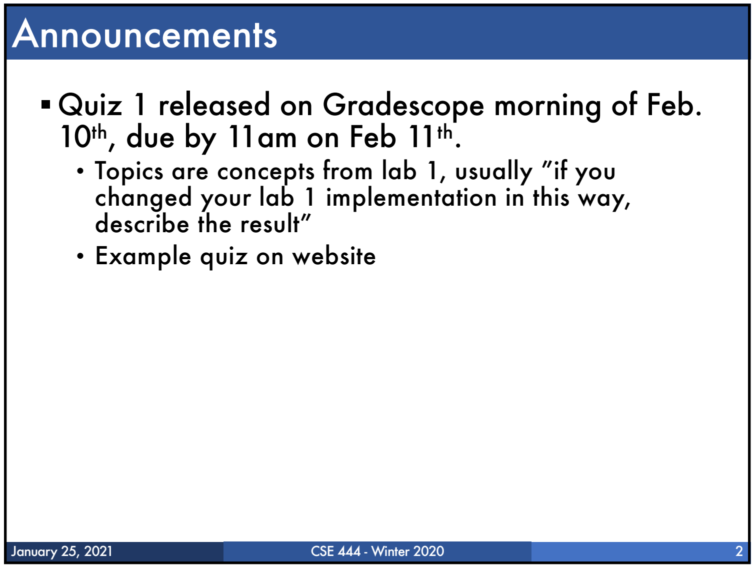


Database System Internals

Operator Algorithms (part 2)

Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle

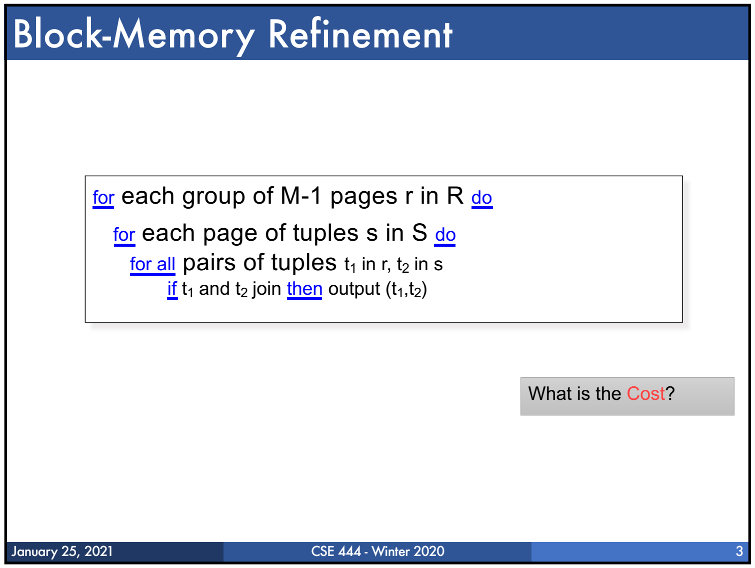
1



Announcements

- Quiz 1 released on Gradescope morning of Feb. 10th, due by 11am on Feb 11th.
  - Topics are concepts from lab 1, usually "if you changed your lab 1 implementation in this way, describe the result"
  - Example quiz on website

2

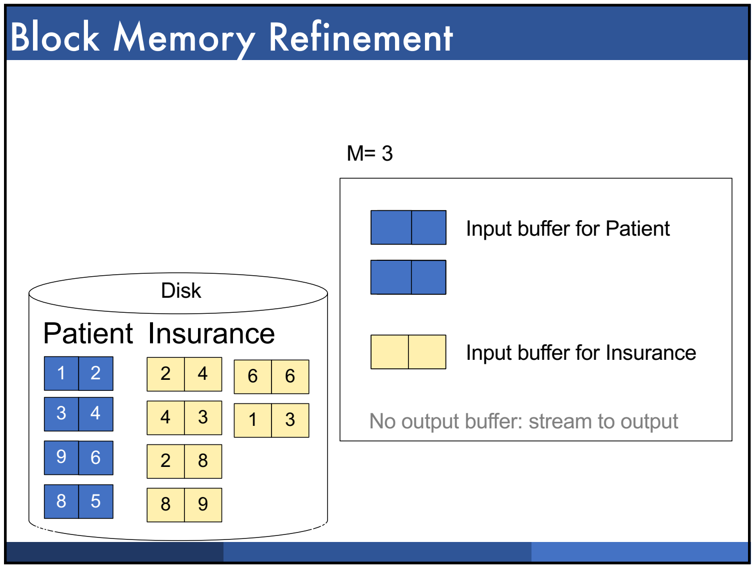


Block-Memory Refinement

```
for each group of M-1 pages r in R do
  for each page of tuples s in S do
    for all pairs of tuples $t_1$ in r, $t_2$ in s
      if $t_1$ and $t_2$ join then output ($t_1$,$t_2$)
```

What is the Cost?

3



Block Memory Refinement

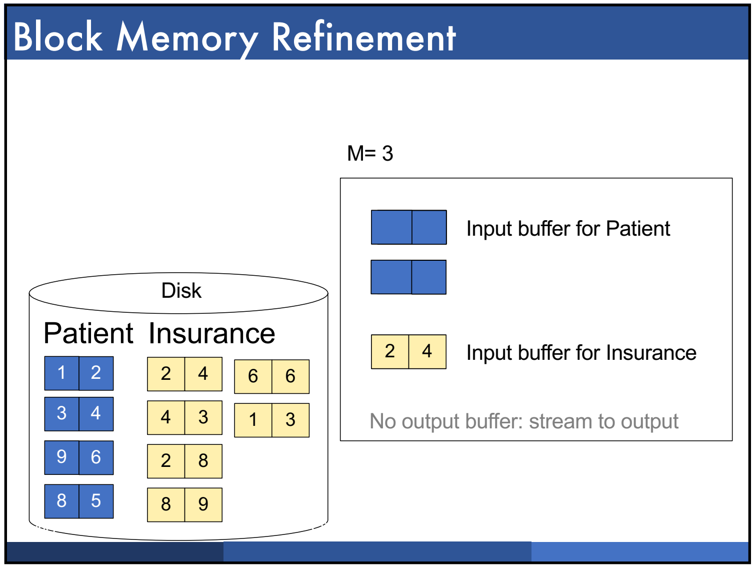
M= 3

Input buffer for Patient

Input buffer for Insurance

No output buffer: stream to output

4



Block Memory Refinement

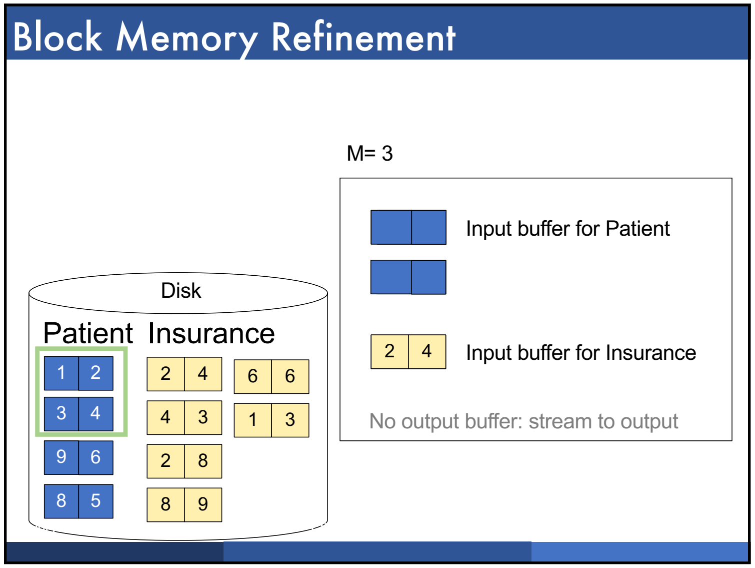
M= 3

Input buffer for Patient

2 4 Input buffer for Insurance

No output buffer: stream to output

5



Block Memory Refinement

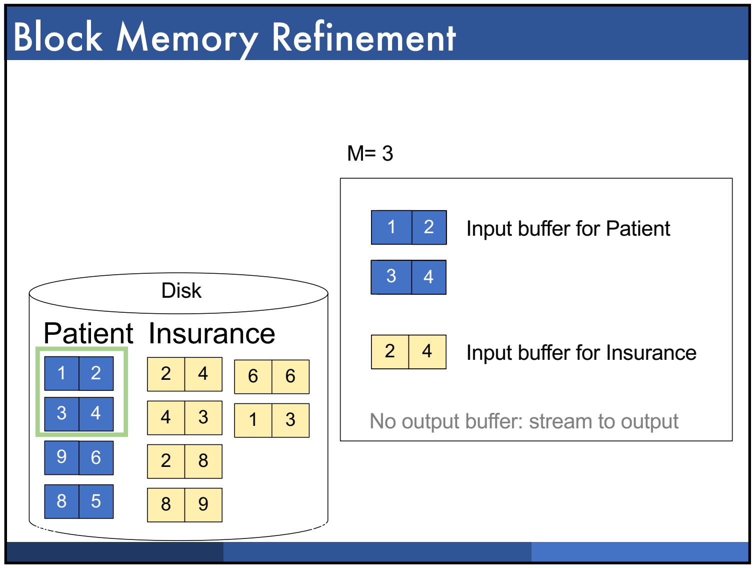
M= 3

Input buffer for Patient

2 4 Input buffer for Insurance

No output buffer: stream to output

6



Block Memory Refinement

M= 3

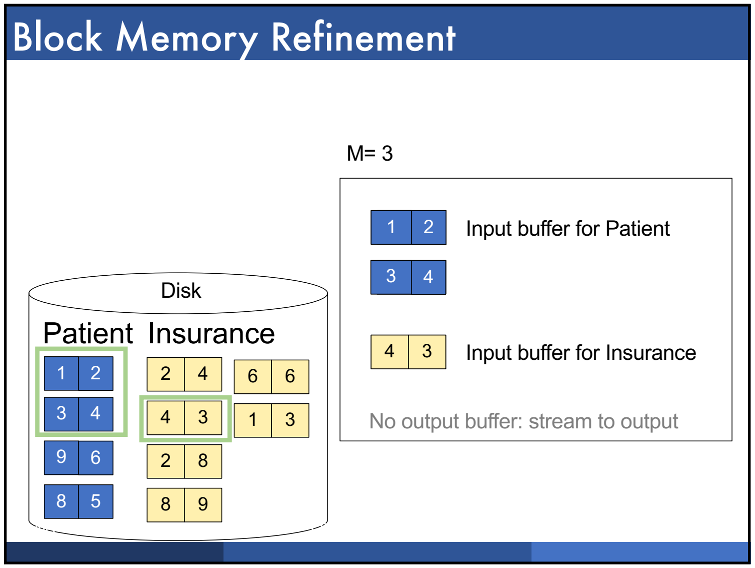
1 2 Input buffer for Patient

3 4

2 4 Input buffer for Insurance

No output buffer: stream to output

7



Block Memory Refinement

M= 3

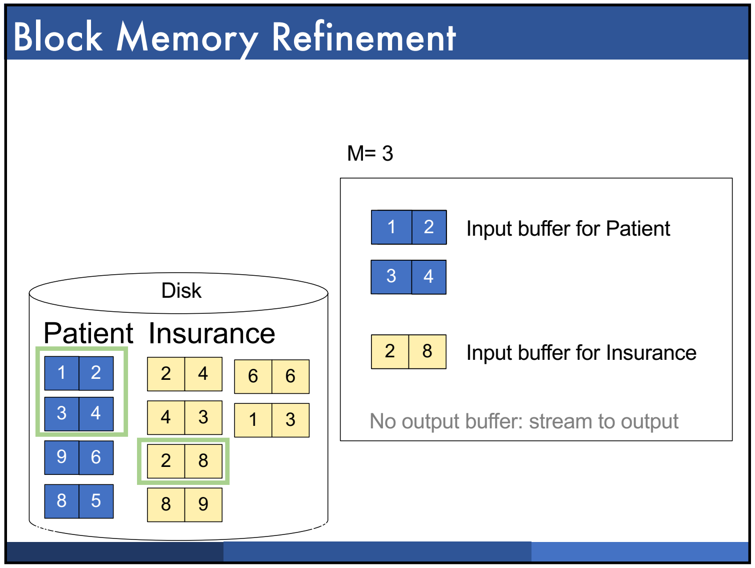
1 2 Input buffer for Patient

3 4

4 3 Input buffer for Insurance

No output buffer: stream to output

8



Block Memory Refinement

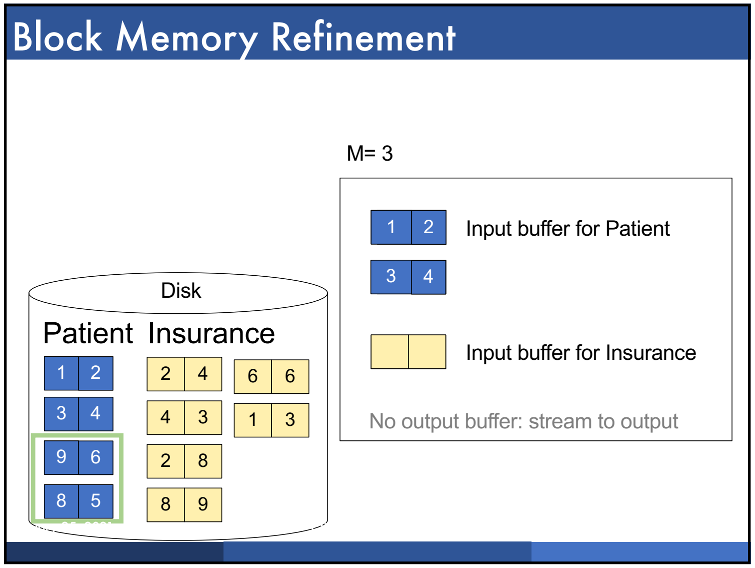
M= 3

Input buffer for Patient

Input buffer for Insurance

No output buffer: stream to output

9



Block Memory Refinement

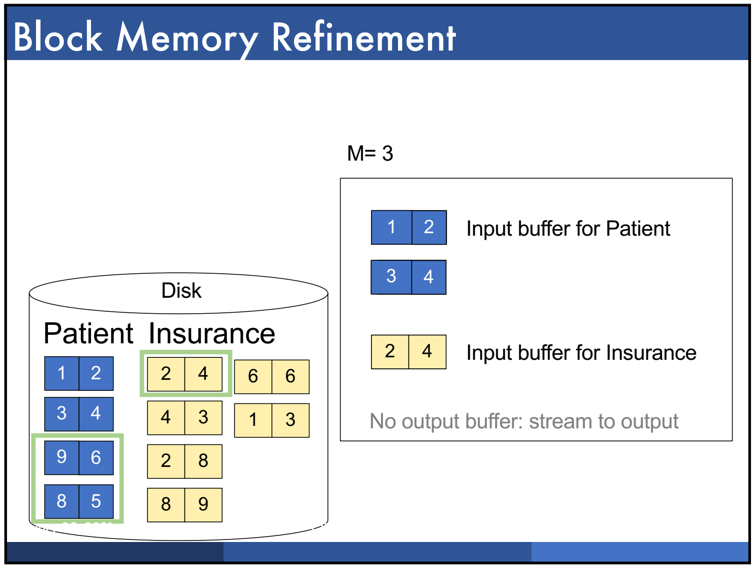
M= 3

Input buffer for Patient

Input buffer for Insurance

No output buffer: stream to output

10



Block Memory Refinement

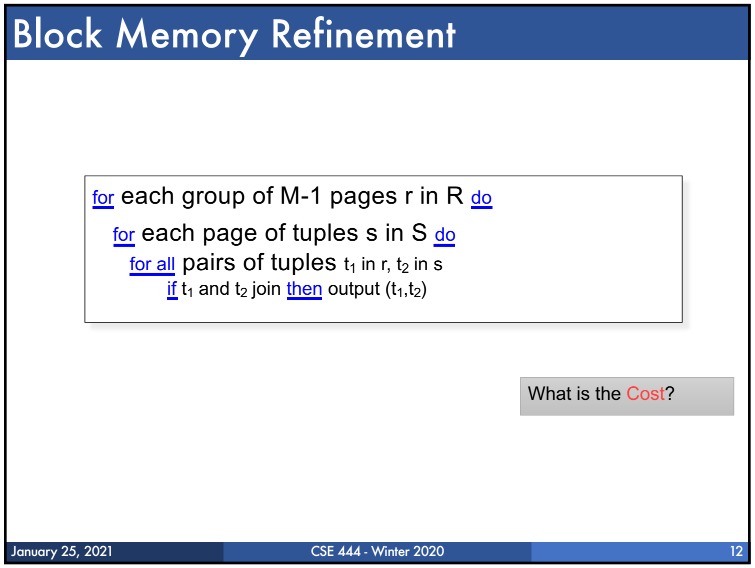
M= 3

Input buffer for Patient

Input buffer for Insurance

No output buffer: stream to output

11



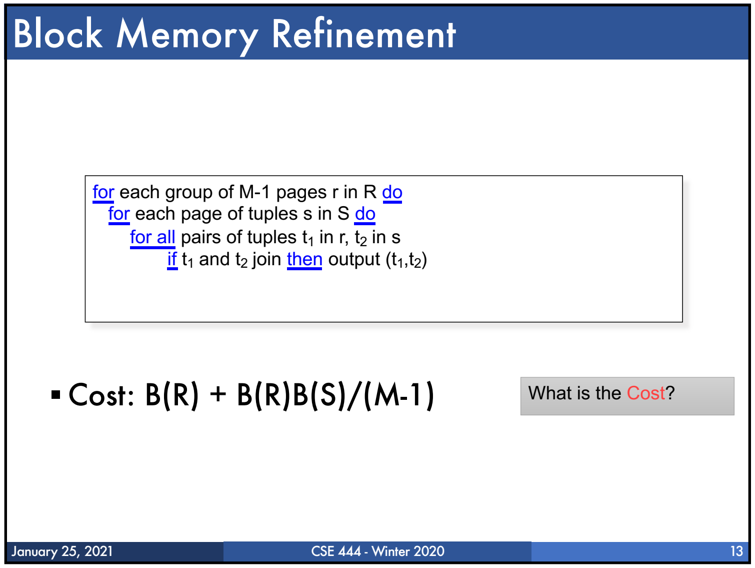
Block Memory Refinement

```
for each group of M-1 pages r in R do
   for each page of tuples s in S do
      for all pairs of tuples $t_1$ in r, $t_2$ in s
         if $t_1$ and $t_2$ join then output $(t_1,t_2)$
```

What is the Cost?

January 25, 2021 CSE 444 - Winter 2020 12

12



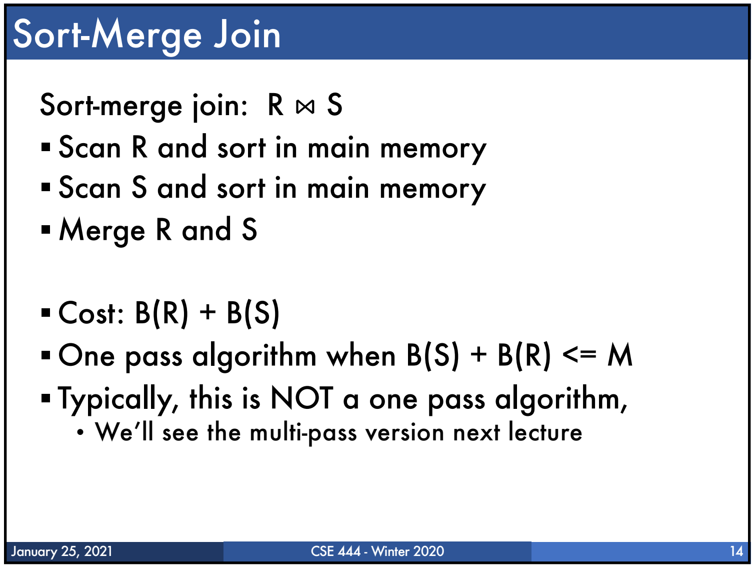
Block Memory Refinement

```
for each group of M-1 pages r in R do
  for each page of tuples s in S do
     for all pairs of tuples $t_1$ in r, $t_2$ in s
          if $t_1$ and $t_2$ join then output ($t_1$,$t_2$)
```

- Cost: B(R) + B(R)B(S)/(M-1)

What is the Cost?

13



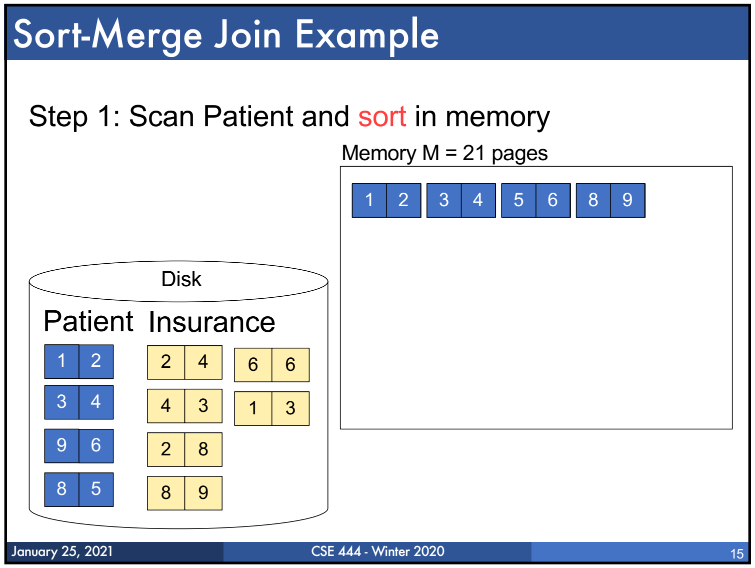
Sort-Merge Join

Sort-merge join: R ⋈ S

- Scan R and sort in main memory
- Scan S and sort in main memory
- Merge R and S

- Cost: B(R) + B(S)
- One pass algorithm when B(S) + B(R) <= M
- Typically, this is NOT a one pass algorithm,
  - We'll see the multi-pass version next lecture

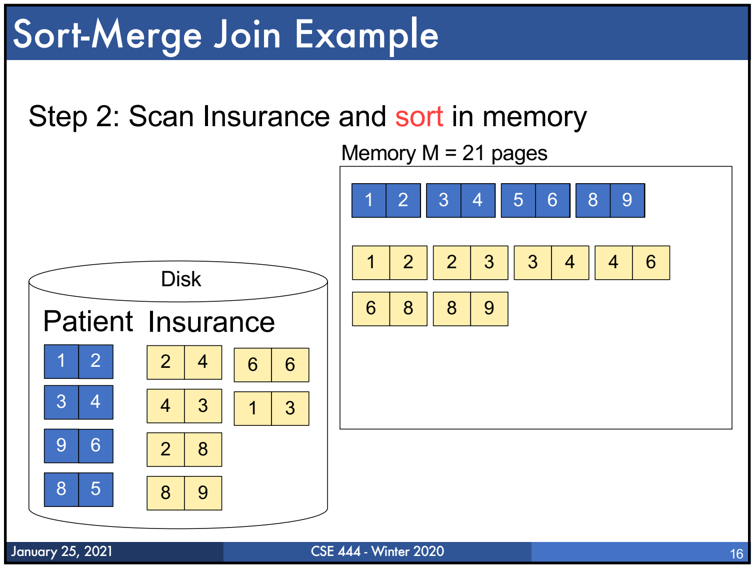
14



Sort-Merge Join Example

Step 1: Scan Patient and sort in memory

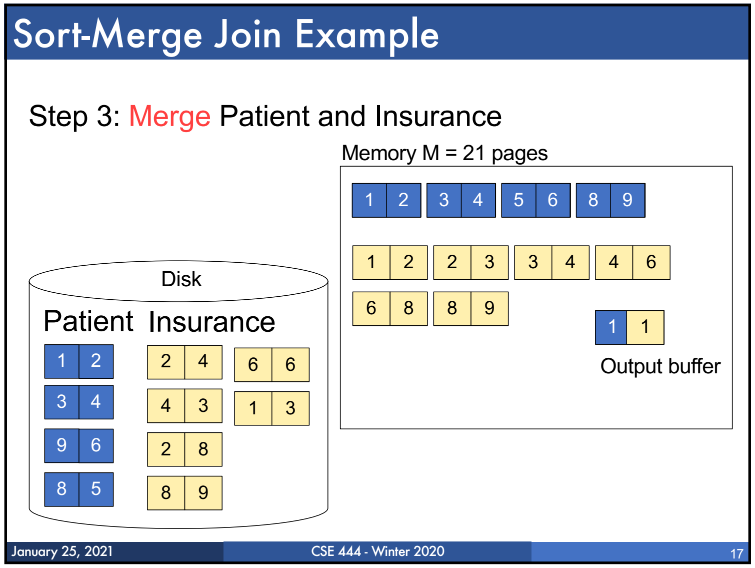
15



Sort-Merge Join Example

Step 2: Scan Insurance and sort in memory

16



Sort-Merge Join Example

Step 3: Merge Patient and Insurance

Memory M = 21 pages

1 2 | 3 4 | 5 6 | 8 9

1 2 | 2 3 | 3 4 | 4 6

6 8 | 8 9

1 1

Output buffer

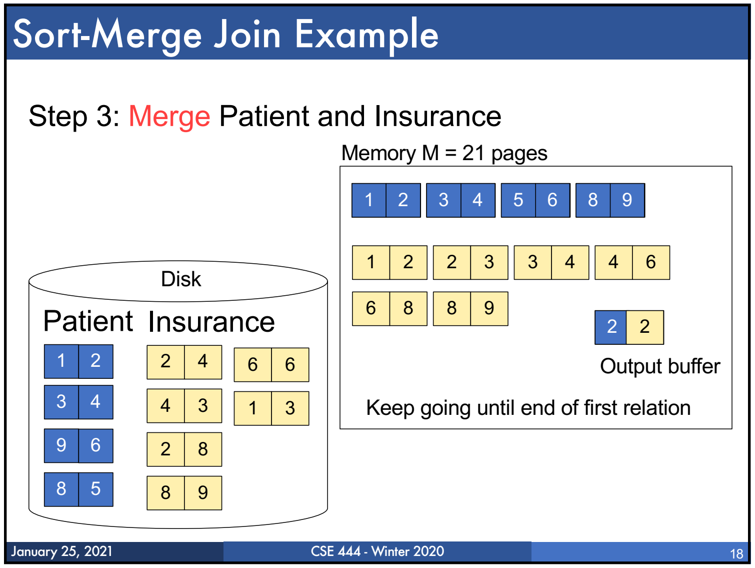
Disk

Patient: 1 2 | 3 4 | 9 6 | 8 5

Insurance: 2 4 | 4 3 | 2 8 | 8 9 | 6 6 | 1 3

January 25, 2021 CSE 444 - Winter 2020 17

17



Sort-Merge Join Example

Step 3: Merge Patient and Insurance

Memory M = 21 pages

1 2 | 3 4 | 5 6 | 8 9

1 2 | 2 3 | 3 4 | 4 6

6 8 | 8 9

2 2

Output buffer

Keep going until end of first relation

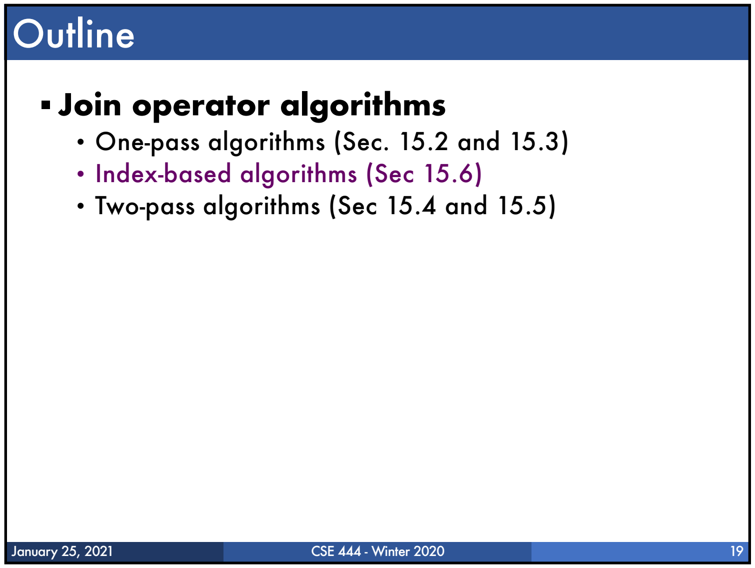
Disk

Patient: 1 2 | 3 4 | 9 6 | 8 5

Insurance: 2 4 | 4 3 | 2 8 | 8 9 | 6 6 | 1 3

January 25, 2021 CSE 444 - Winter 2020 18

18

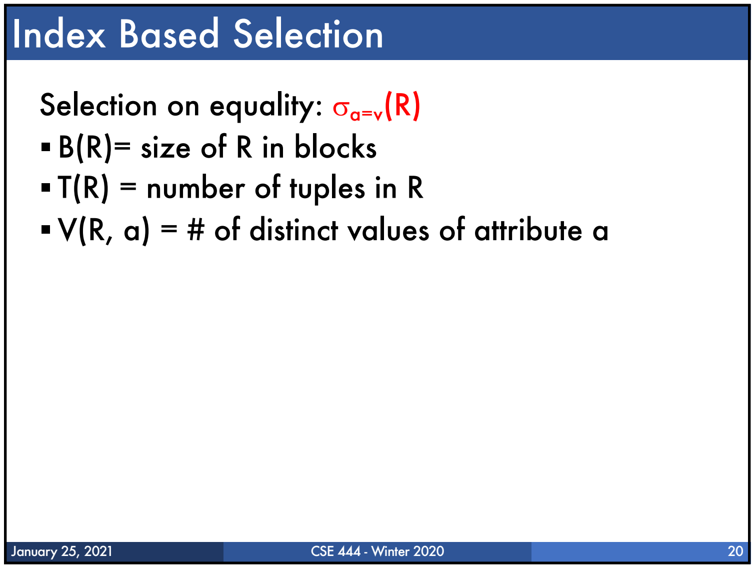


Outline

- **Join operator algorithms**
  - One-pass algorithms (Sec. 15.2 and 15.3)
  - Index-based algorithms (Sec 15.6)
  - Two-pass algorithms (Sec 15.4 and 15.5)

January 25, 2021 CSE 444 - Winter 2020 19

19



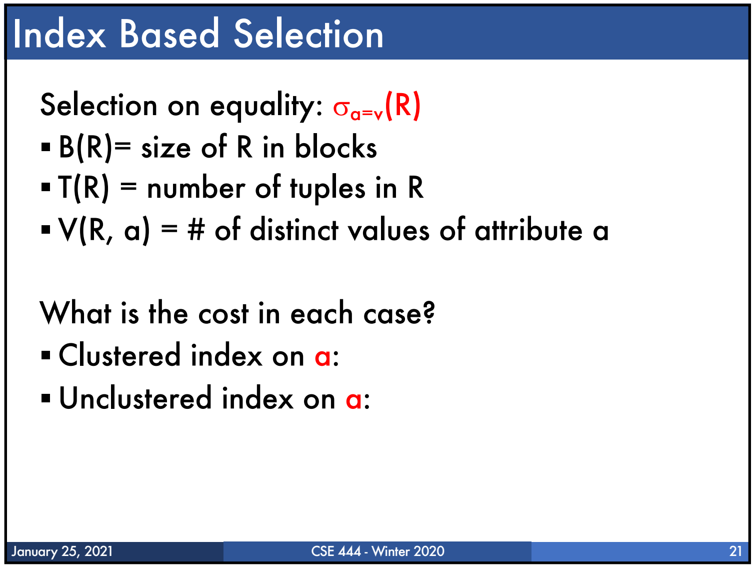
Index Based Selection

Selection on equality: $\sigma_{a=v}(R)$

- B(R)= size of R in blocks
- T(R) = number of tuples in R
- V(R, a) = # of distinct values of attribute a

January 25, 2021 CSE 444 - Winter 2020 20

20



Index Based Selection

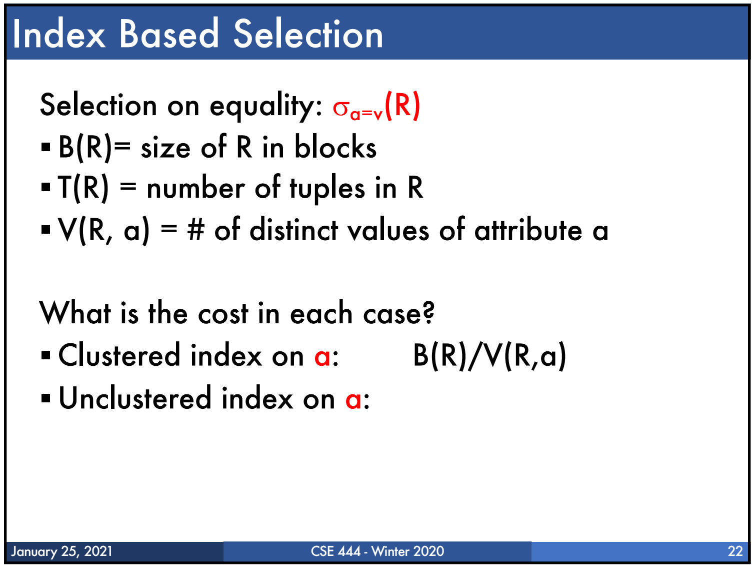
Selection on equality: $\sigma_{a=v}(R)$

- B(R)= size of R in blocks
- T(R) = number of tuples in R
- V(R, a) = # of distinct values of attribute a

What is the cost in each case?

- Clustered index on a:
- Unclustered index on a:

21



Index Based Selection

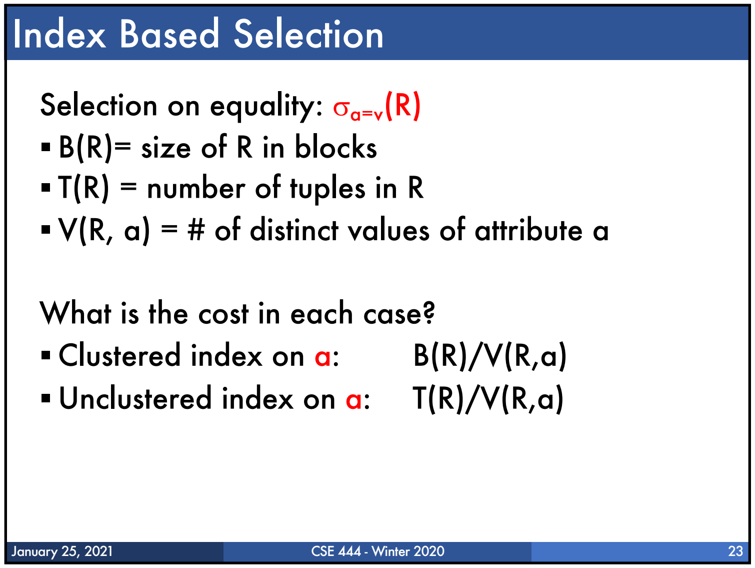
Selection on equality: $\sigma_{a=v}(R)$

- B(R)= size of R in blocks
- T(R) = number of tuples in R
- V(R, a) = # of distinct values of attribute a

What is the cost in each case?

- Clustered index on a: B(R)/V(R,a)
- Unclustered index on a:

22



Index Based Selection

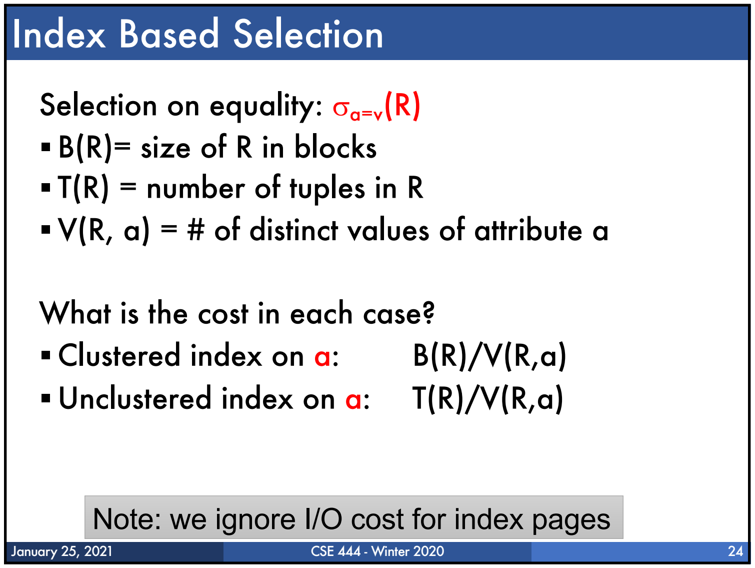
Selection on equality: $\sigma_{a=v}(R)$

- B(R)= size of R in blocks
- T(R) = number of tuples in R
- V(R, a) = # of distinct values of attribute a

What is the cost in each case?

- Clustered index on a: B(R)/V(R,a)
- Unclustered index on a: T(R)/V(R,a)

23



Index Based Selection

Selection on equality: $\sigma_{a=v}(R)$

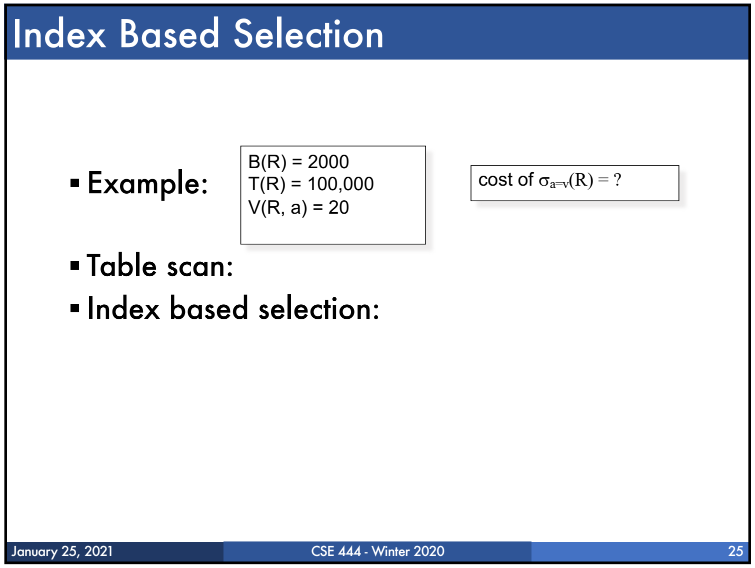
- B(R)= size of R in blocks
- T(R) = number of tuples in R
- V(R, a) = # of distinct values of attribute a

What is the cost in each case?

- Clustered index on a: B(R)/V(R,a)
- Unclustered index on a: T(R)/V(R,a)

Note: we ignore I/O cost for index pages

24



Index Based Selection

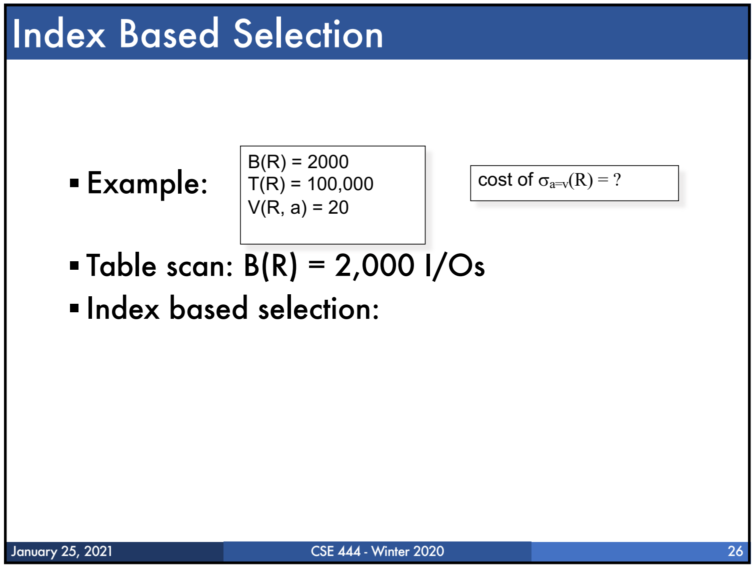
- Example:

B(R) = 2000
T(R) = 100,000
V(R, a) = 20

cost of $\sigma_{a=v}(R)$ = ?

- Table scan:
- Index based selection:

25



Index Based Selection

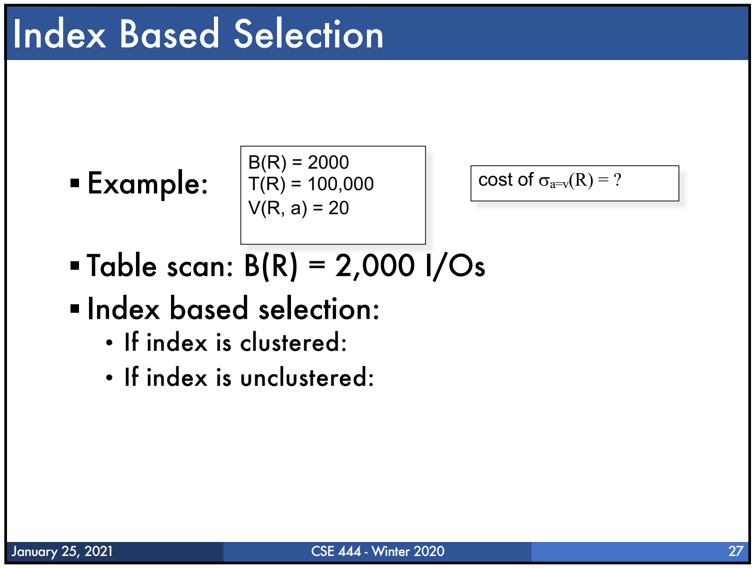
- Example:

B(R) = 2000
T(R) = 100,000
V(R, a) = 20

cost of $\sigma_{a=v}(R)$ = ?

- Table scan: B(R) = 2,000 I/Os
- Index based selection:

26



Index Based Selection

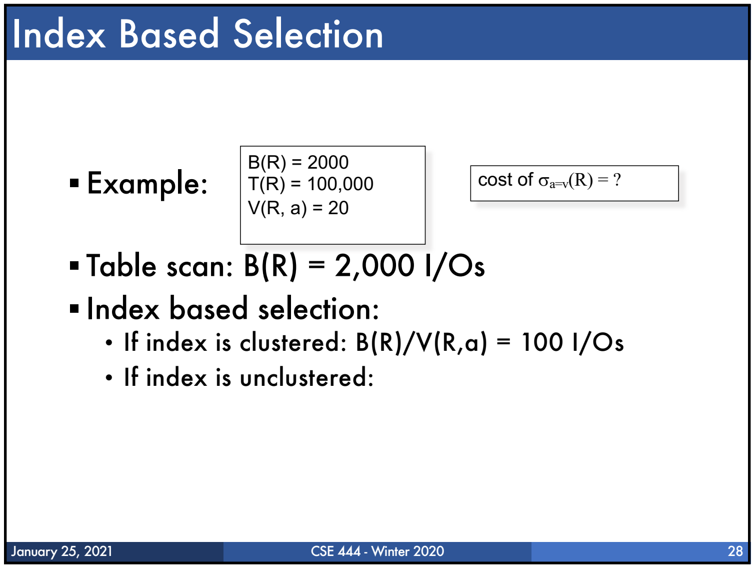
- Example:

B(R) = 2000
T(R) = 100,000
V(R, a) = 20

cost of $\sigma_{a=v}(R)$ = ?

- Table scan: B(R) = 2,000 I/Os
- Index based selection:
  - If index is clustered:
  - If index is unclustered:

27



Index Based Selection

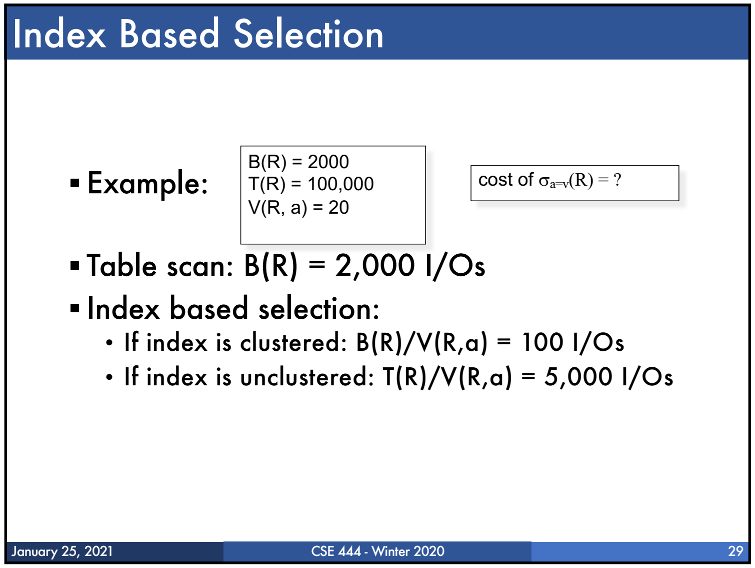
- Example:

B(R) = 2000
T(R) = 100,000
V(R, a) = 20

cost of $\sigma_{a=v}(R)$ = ?

- Table scan: B(R) = 2,000 I/Os
- Index based selection:
  - If index is clustered: B(R)/V(R,a) = 100 I/Os
  - If index is unclustered:

28



Index Based Selection

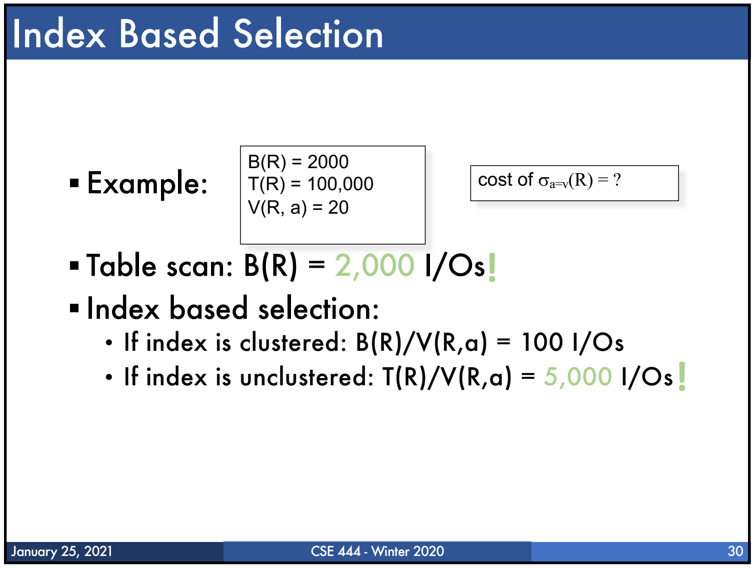
- Example:

B(R) = 2000 T(R) = 100,000 V(R, a) = 20

cost of $\sigma_{a=v}(R)$ = ?

- Table scan: B(R) = 2,000 I/Os
- Index based selection:
  - If index is clustered: B(R)/V(R,a) = 100 I/Os
  - If index is unclustered: T(R)/V(R,a) = 5,000 I/Os

29



Index Based Selection

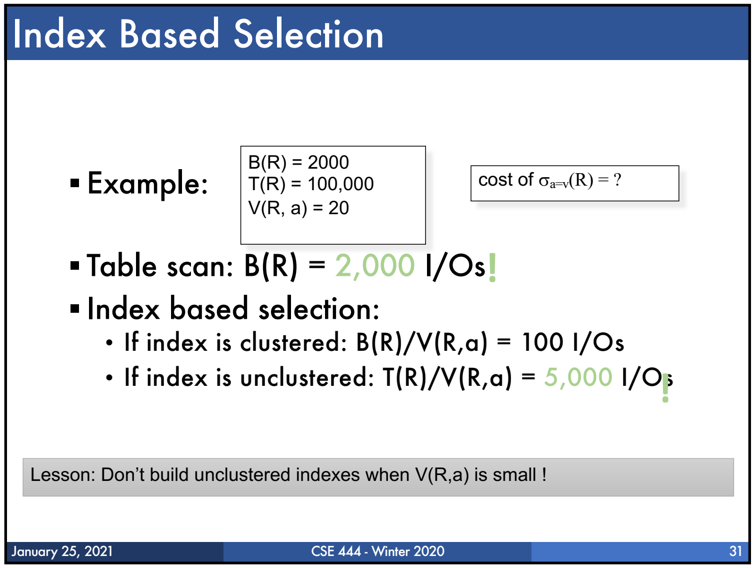
- Example:

B(R) = 2000 T(R) = 100,000 V(R, a) = 20

cost of $\sigma_{a=v}(R)$ = ?

- Table scan: B(R) = 2,000 I/Os!
- Index based selection:
  - If index is clustered: B(R)/V(R,a) = 100 I/Os
  - If index is unclustered: T(R)/V(R,a) = 5,000 I/Os!

30



Index Based Selection

- Example:

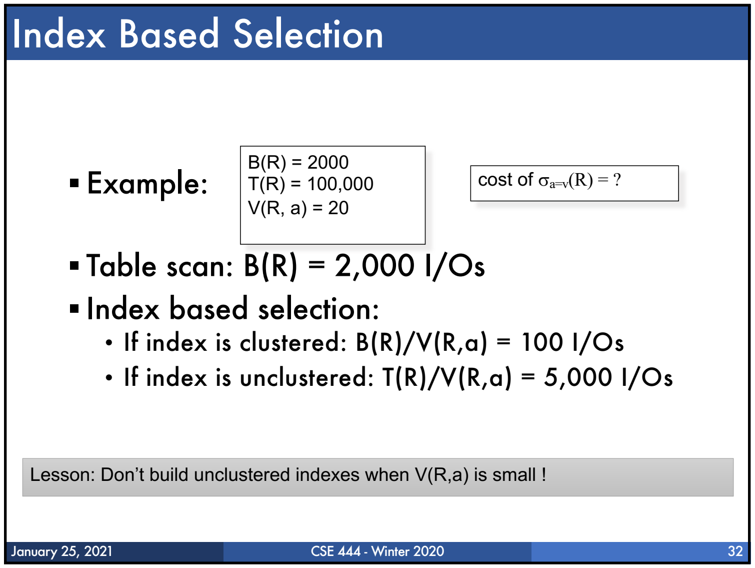
B(R) = 2000 T(R) = 100,000 V(R, a) = 20

cost of $\sigma_{a=v}(R)$ = ?

- Table scan: B(R) = 2,000 I/Os!
- Index based selection:
  - If index is clustered: B(R)/V(R,a) = 100 I/Os
  - If index is unclustered: T(R)/V(R,a) = 5,000 I/Os!

Lesson: Don't build unclustered indexes when V(R,a) is small !

31



Index Based Selection

- Example:

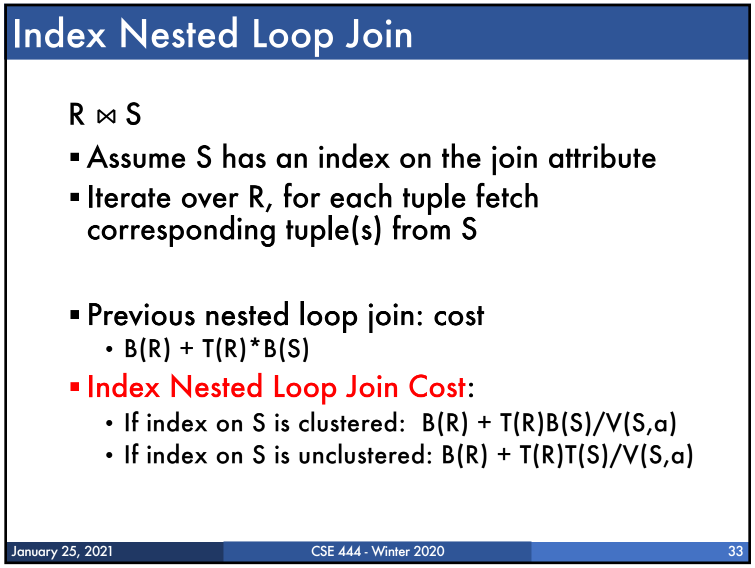
B(R) = 2000 T(R) = 100,000 V(R, a) = 20

cost of $\sigma_{a=v}(R)$ = ?

- Table scan: B(R) = 2,000 I/Os
- Index based selection:
  - If index is clustered: B(R)/V(R,a) = 100 I/Os
  - If index is unclustered: T(R)/V(R,a) = 5,000 I/Os

Lesson: Don't build unclustered indexes when V(R,a) is small !

32



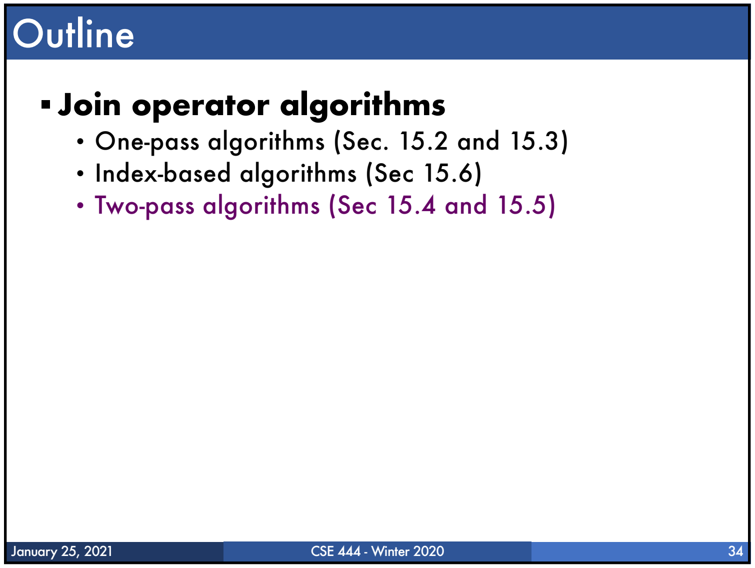
Index Nested Loop Join

$R \bowtie S$

- Assume S has an index on the join attribute
- Iterate over R, for each tuple fetch corresponding tuple(s) from S

- Previous nested loop join: cost
  - B(R) + T(R)*B(S)
- Index Nested Loop Join Cost:
  - If index on S is clustered: B(R) + T(R)B(S)/V(S,a)
  - If index on S is unclustered: B(R) + T(R)T(S)/V(S,a)

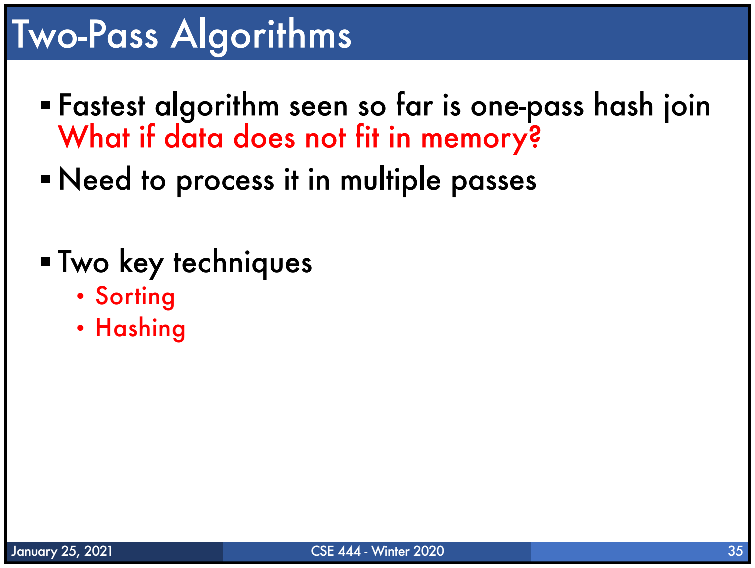
33



Outline

- **Join operator algorithms**
  - One-pass algorithms (Sec. 15.2 and 15.3)
  - Index-based algorithms (Sec 15.6)
  - Two-pass algorithms (Sec 15.4 and 15.5)

34

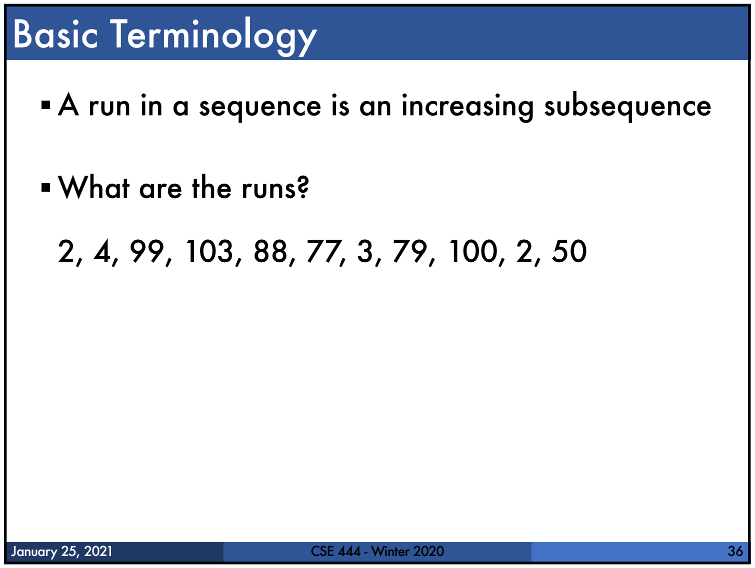


Two-Pass Algorithms

- Fastest algorithm seen so far is one-pass hash join
  What if data does not fit in memory?
- Need to process it in multiple passes

- Two key techniques
  - Sorting
  - Hashing

35



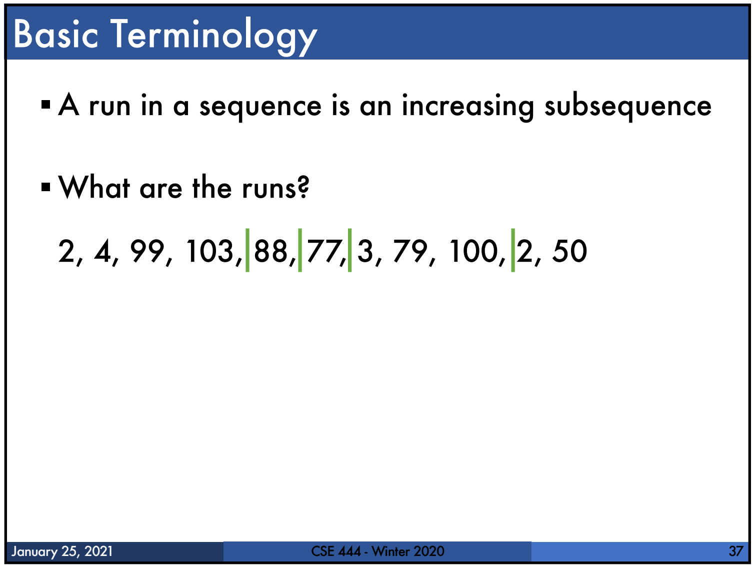
Basic Terminology

- A run in a sequence is an increasing subsequence

- What are the runs?

  2, 4, 99, 103, 88, 77, 3, 79, 100, 2, 50

36

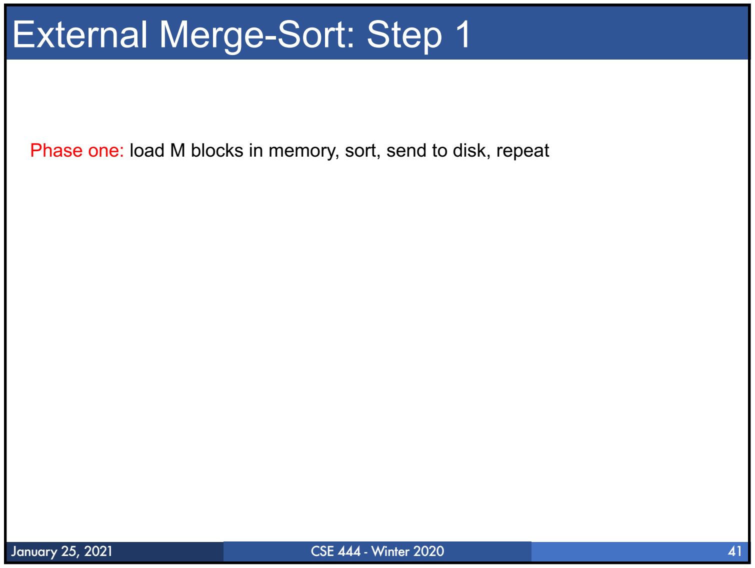


Basic Terminology

- A run in a sequence is an increasing subsequence
- What are the runs?

2, 4, 99, 103,|88,|77,|3, 79, 100,|2, 50

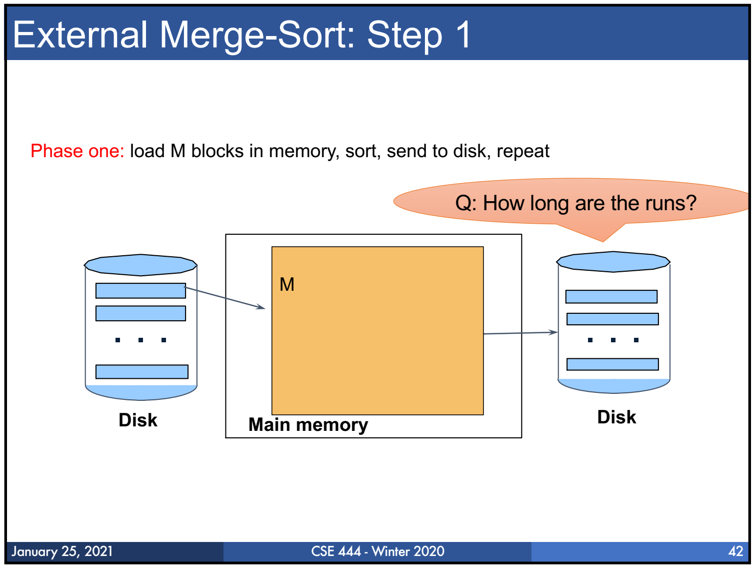
37



External Merge-Sort: Step 1

Phase one: load M blocks in memory, sort, send to disk, repeat

41

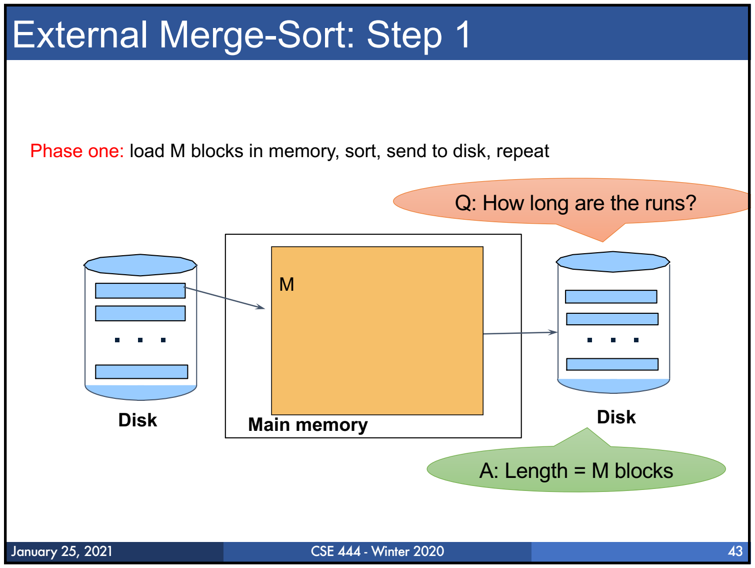


External Merge-Sort: Step 1

Phase one: load M blocks in memory, sort, send to disk, repeat

Q: How long are the runs?

42



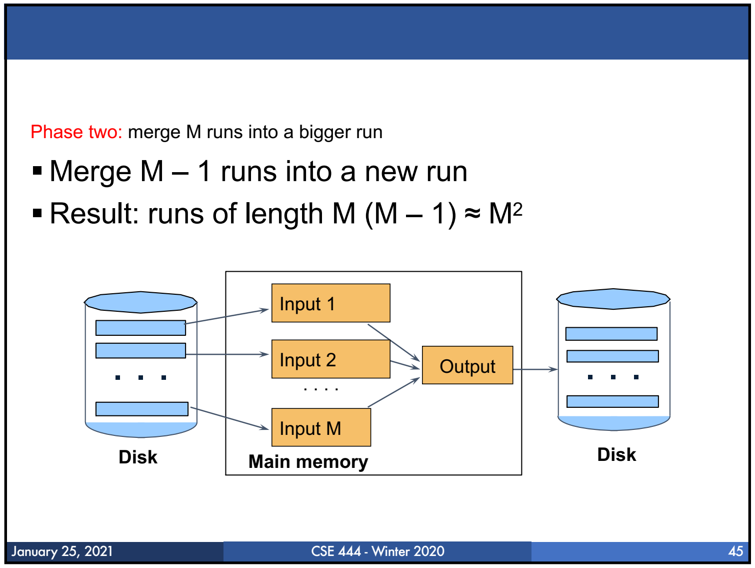
External Merge-Sort: Step 1

Phase one: load M blocks in memory, sort, send to disk, repeat

Q: How long are the runs?

A: Length = M blocks

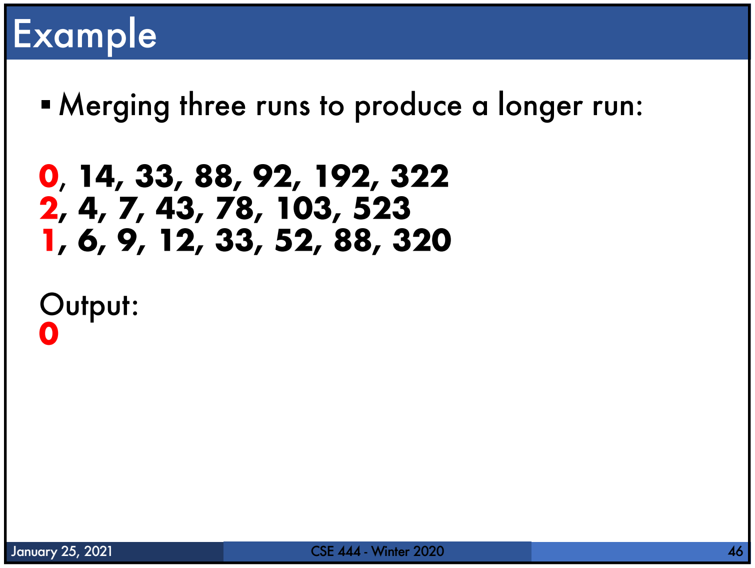
43



Phase two: merge M runs into a bigger run

- Merge M – 1 runs into a new run
- Result: runs of length M (M – 1) ≈ M^2

45



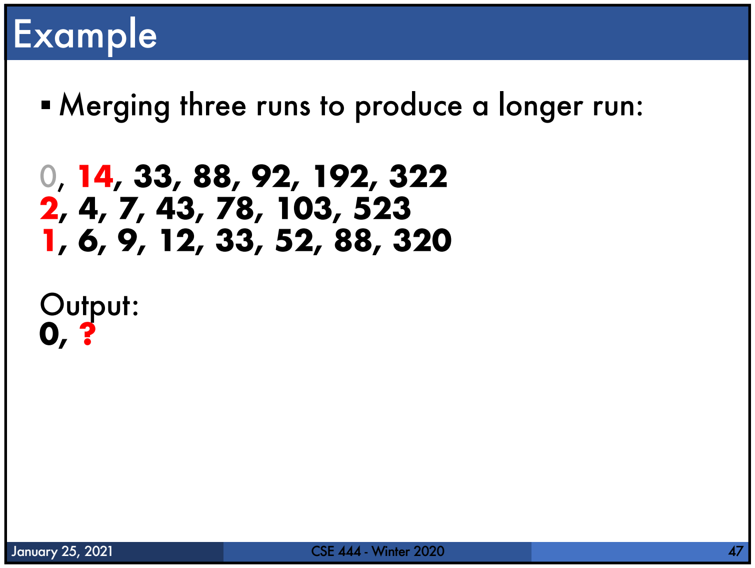
Example

- Merging three runs to produce a longer run:

0, 14, 33, 88, 92, 192, 322
2, 4, 7, 43, 78, 103, 523
1, 6, 9, 12, 33, 52, 88, 320

Output:
0

46



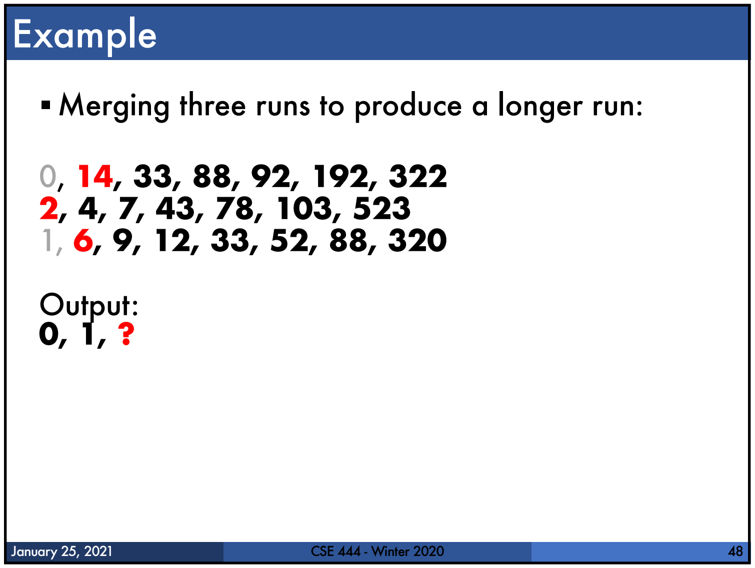
Example

- Merging three runs to produce a longer run:

0, **14, 33, 88, 92, 192, 322**
2, 4, 7, 43, 78, 103, 523
1, 6, 9, 12, 33, 52, 88, 320

Output:
0, ?

47



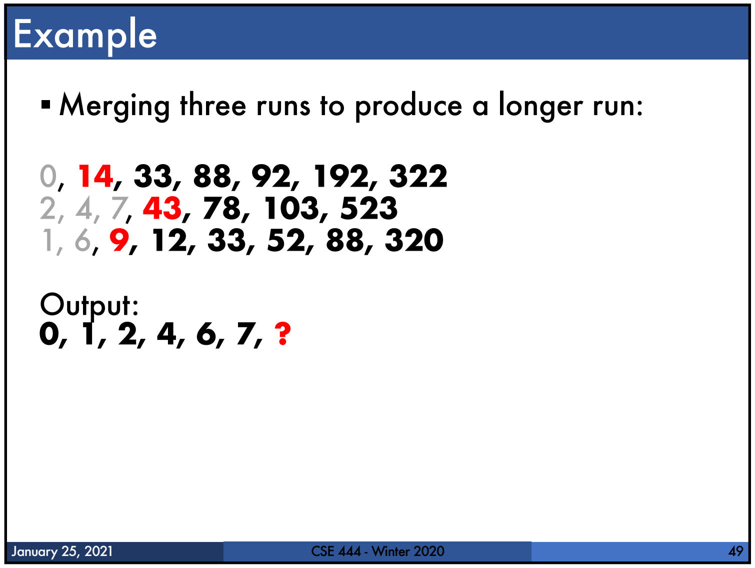
Example

- Merging three runs to produce a longer run:

0, **14, 33, 88, 92, 192, 322**
2, 4, 7, 43, 78, 103, 523
1, **6, 9, 12, 33, 52, 88, 320**

Output:
0, 1, ?

48



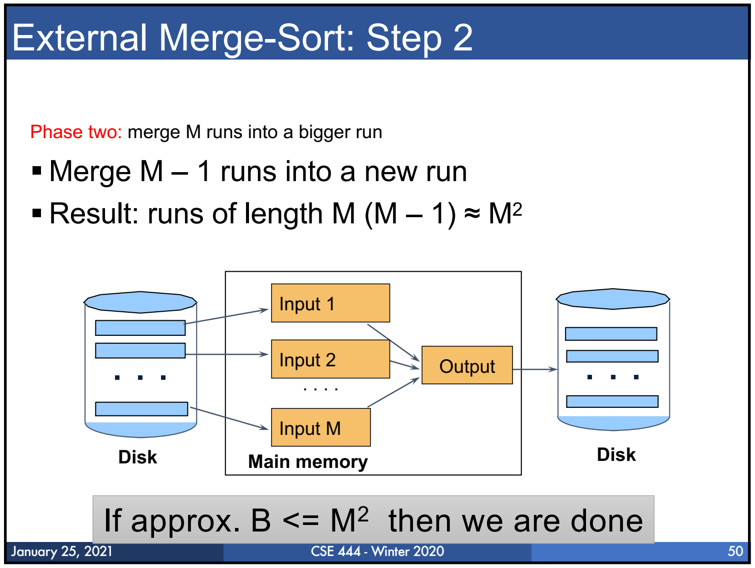
Example

- Merging three runs to produce a longer run:

0, **14, 33, 88, 92, 192, 322**
2, 4, 7, **43, 78, 103, 523**
1, 6, **9, 12, 33, 52, 88, 320**

Output:
0, 1, 2, 4, 6, 7, ?

49



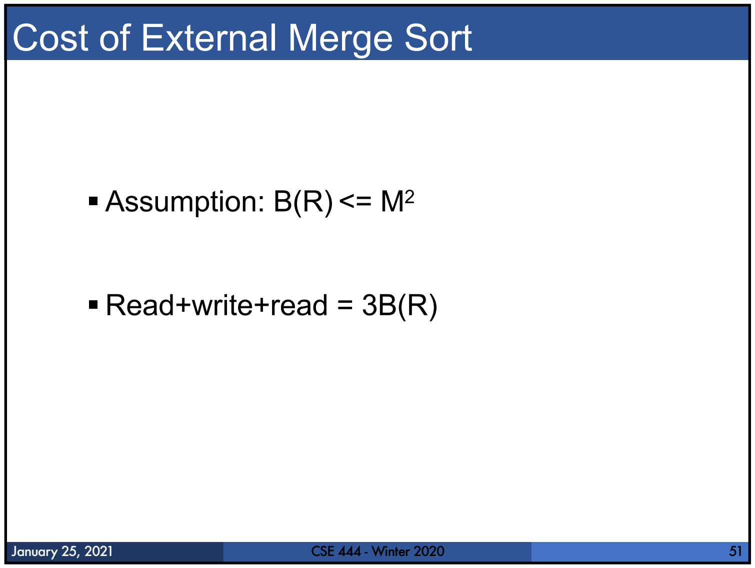
External Merge-Sort: Step 2

Phase two: merge M runs into a bigger run

- Merge M – 1 runs into a new run
- Result: runs of length M (M – 1) ≈ M^2

If approx. B <= M^2 then we are done

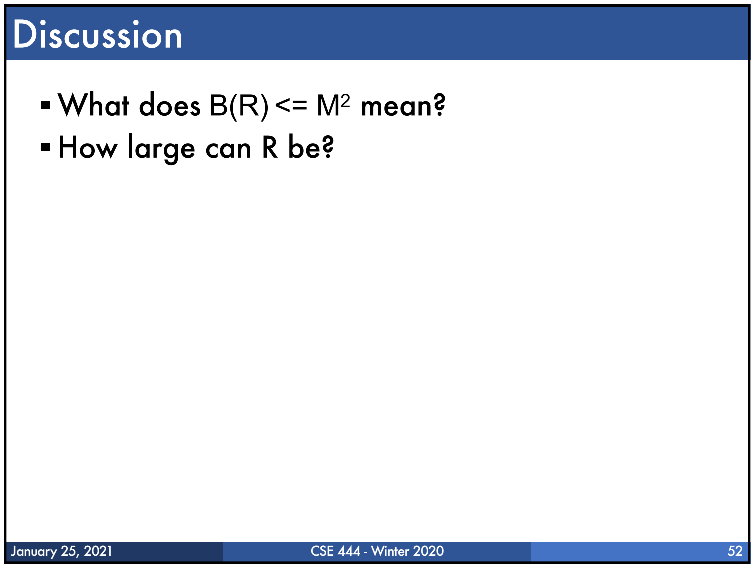
50



Cost of External Merge Sort

- Assumption: B(R) <= M^2
- Read+write+read = 3B(R)

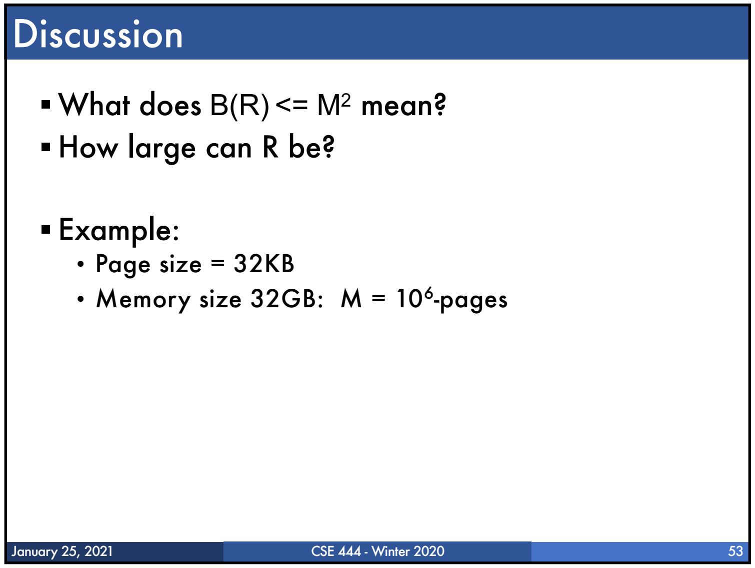
51



Discussion

- What does B(R) <= M^2 mean?
- How large can R be?

52

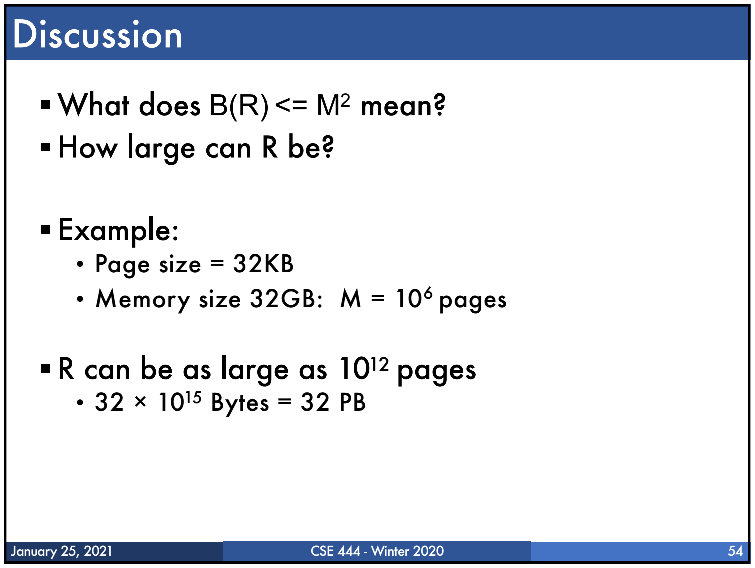


Discussion

- What does $B(R) <= M^2$ mean?
- How large can R be?

- Example:
  - Page size = 32KB
  - Memory size 32GB: $M = 10^6$-pages

53



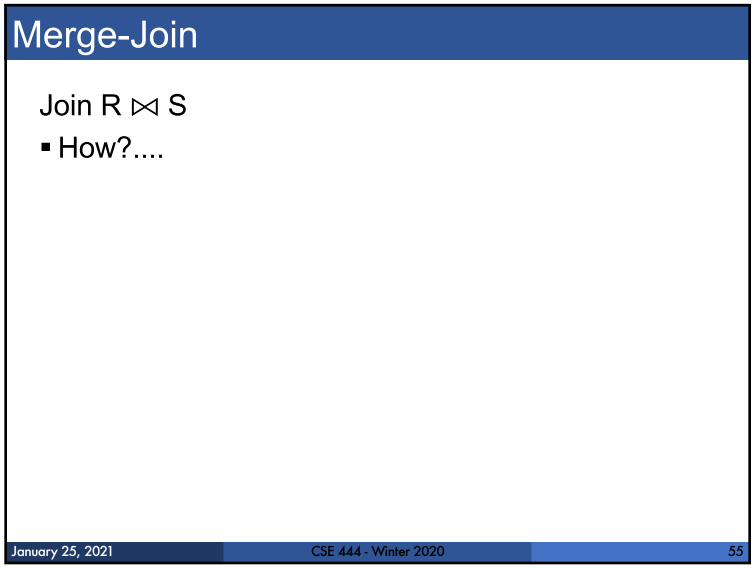
Discussion

- What does $B(R) <= M^2$ mean?
- How large can R be?

- Example:
  - Page size = 32KB
  - Memory size 32GB: $M = 10^6$ pages

- R can be as large as 10^{12} pages
  - 32×10^{15} Bytes = 32 PB

54

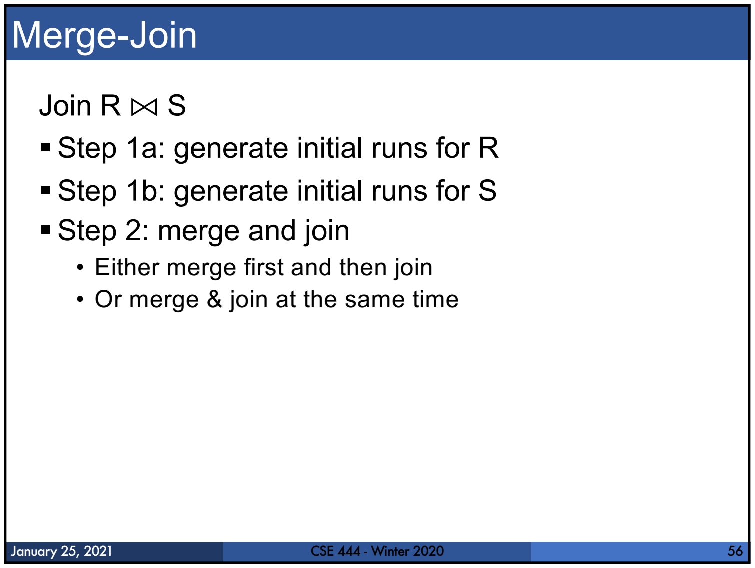


Merge-Join

Join $R \bowtie S$

- How?....

55

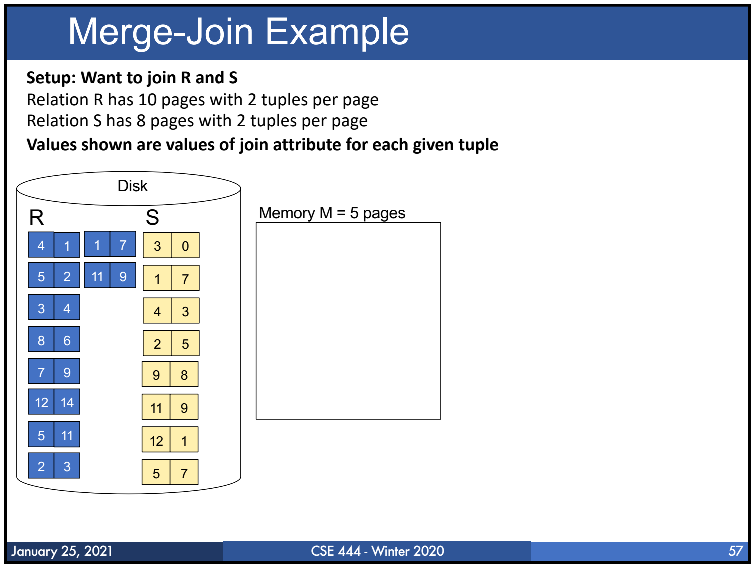


Merge-Join

Join $R \bowtie S$

- Step 1a: generate initial runs for R
- Step 1b: generate initial runs for S
- Step 2: merge and join
  - Either merge first and then join
  - Or merge & join at the same time

56

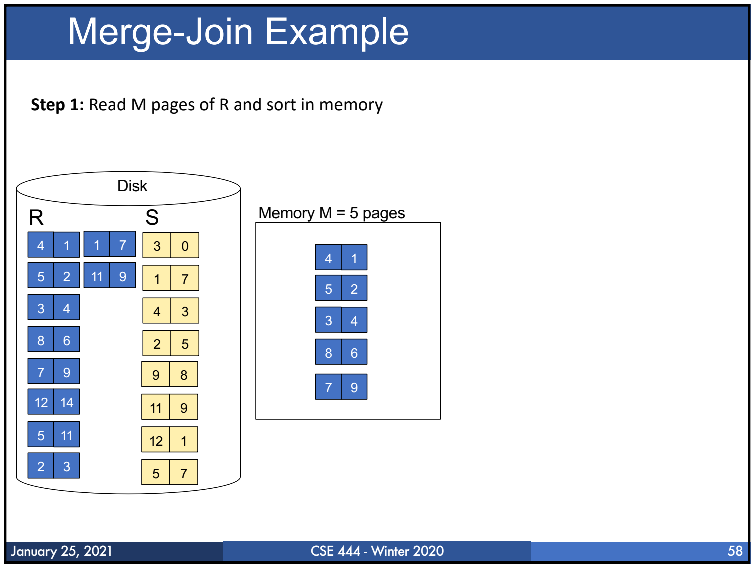


Merge-Join Example

Setup: Want to join R and S
Relation R has 10 pages with 2 tuples per page
Relation S has 8 pages with 2 tuples per page
Values shown are values of join attribute for each given tuple

January 25, 2021 CSE 444 - Winter 2020 57

57

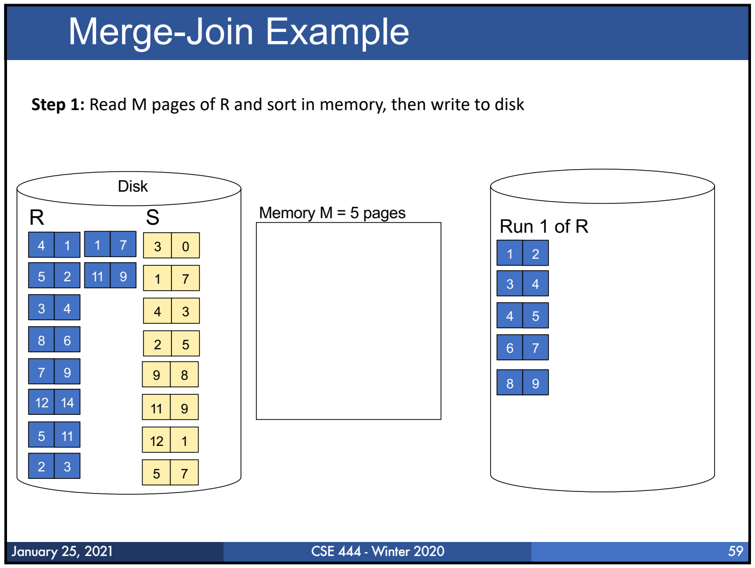


Merge-Join Example

Step 1: Read M pages of R and sort in memory

January 25, 2021 CSE 444 - Winter 2020 58

58

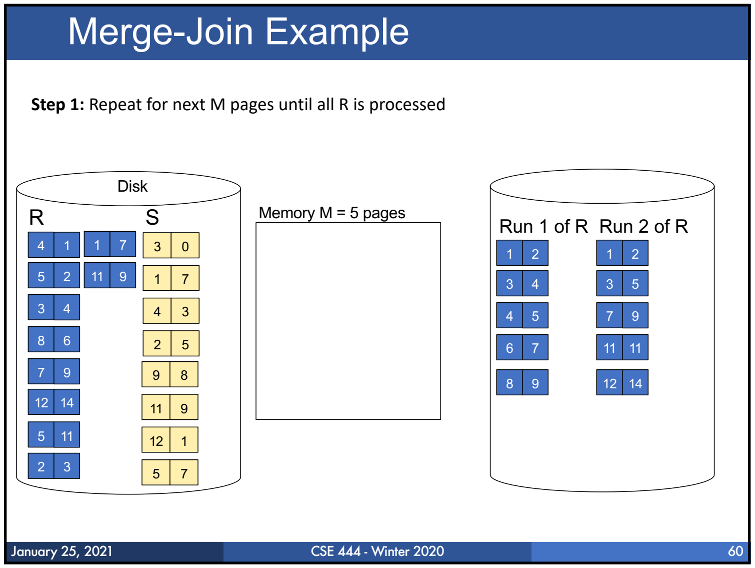


Merge-Join Example

Step 1: Read M pages of R and sort in memory, then write to disk

January 25, 2021 CSE 444 - Winter 2020 59

59

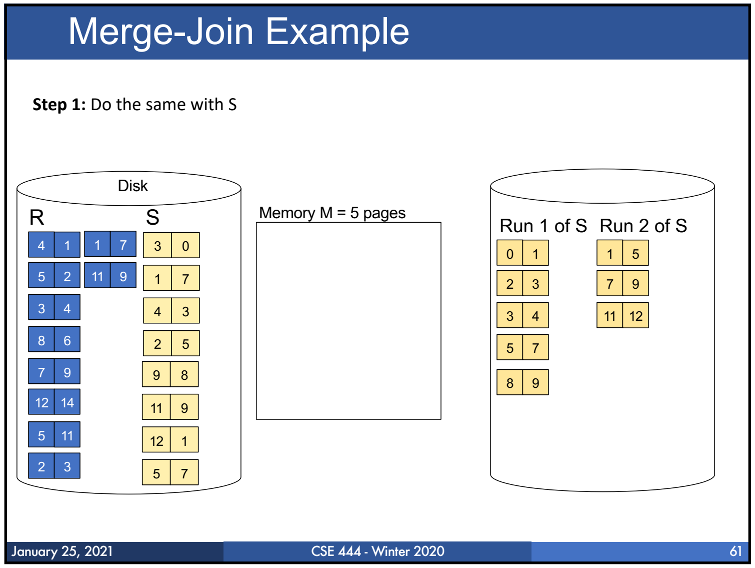


Merge-Join Example

Step 1: Repeat for next M pages until all R is processed

January 25, 2021 CSE 444 - Winter 2020 60

60



Merge-Join Example

Step 1: Do the same with S

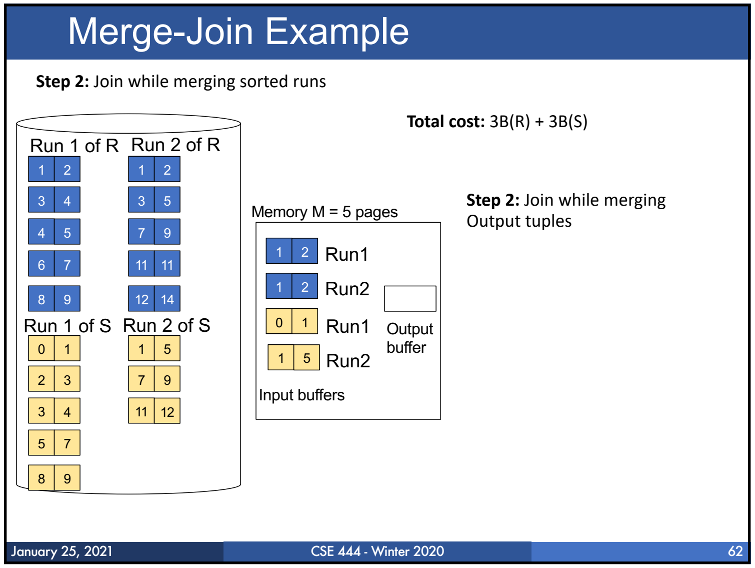
Disk

R		S
4 1	1 7	3 0
5 2	11 9	1 7
3 4		4 3
8 6		2 5
7 9		9 8
12 14		11 9
5 11		12 1
2 3		5 7

Memory M = 5 pages

Run 1 of S	Run 2 of S
0 1	1 5
2 3	7 9
3 4	11 12
5 7	
8 9	

61



Merge-Join Example

Step 2: Join while merging sorted runs

Total cost: 3B(R) + 3B(S)

Run 1 of R	Run 2 of R
1 2	1 2
3 4	3 5
4 5	7 9
6 7	11 11
8 9	12 14

Run 1 of S	Run 2 of S
0 1	1 5
2 3	7 9
3 4	11 12
5 7	
8 9	

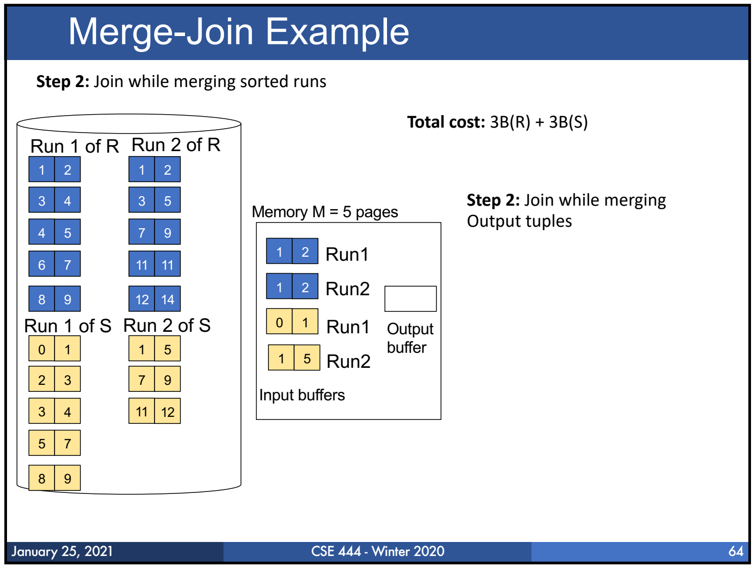
Memory M = 5 pages

1 2 Run1
1 2 Run2
0 1 Run1
1 5 Run2
Input buffers

Output buffer

Step 2: Join while merging
Output tuples

62



Merge-Join Example

Step 2: Join while merging sorted runs

Total cost: 3B(R) + 3B(S)

Run 1 of R	Run 2 of R
1 2	1 2
3 4	3 5
4 5	7 9
6 7	11 11
8 9	12 14

Run 1 of S	Run 2 of S
0 1	1 5
2 3	7 9
3 4	11 12
5 7	
8 9	

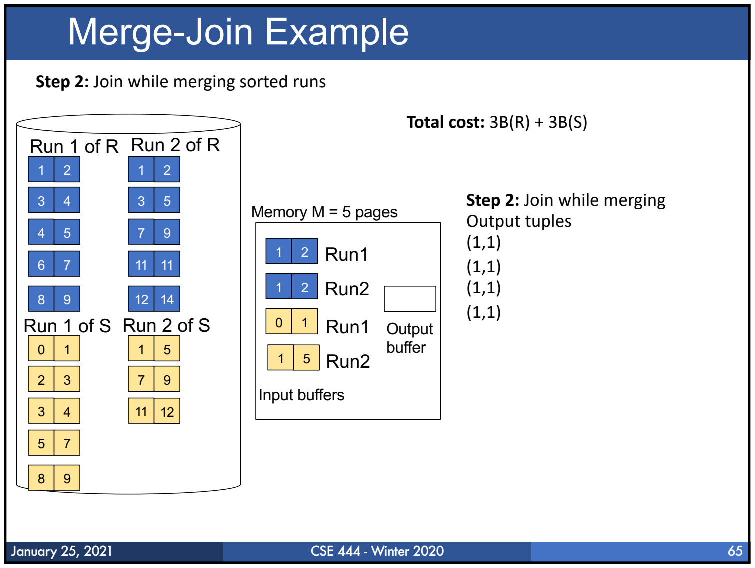
Memory M = 5 pages

1 2 Run1
1 2 Run2
0 1 Run1
1 5 Run2
Input buffers

Output buffer

Step 2: Join while merging
Output tuples

64



Merge-Join Example

Step 2: Join while merging sorted runs

Total cost: 3B(R) + 3B(S)

Run 1 of R	Run 2 of R
1 2	1 2
3 4	3 5
4 5	7 9
6 7	11 11
8 9	12 14

Run 1 of S	Run 2 of S
0 1	1 5
2 3	7 9
3 4	11 12
5 7	
8 9	

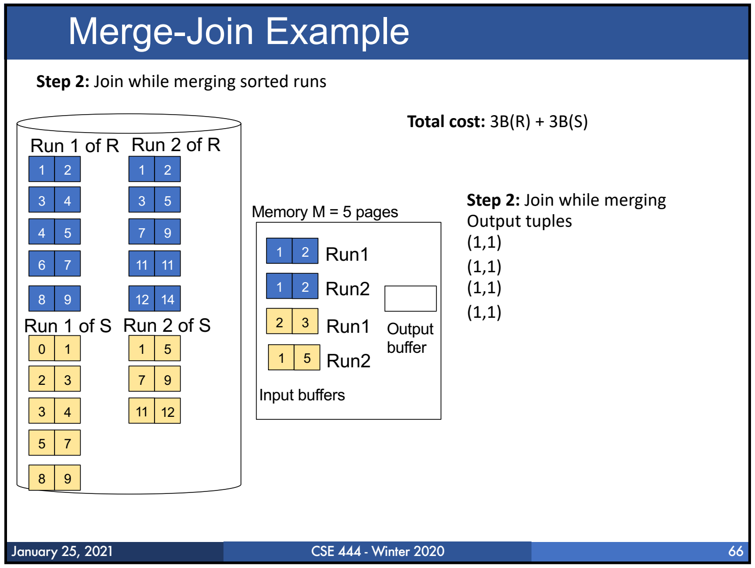
Memory M = 5 pages

1 2 Run1
1 2 Run2
0 1 Run1
1 5 Run2
Input buffers

Output buffer

Step 2: Join while merging
Output tuples
(1,1)
(1,1)
(1,1)
(1,1)

65



Merge-Join Example

Step 2: Join while merging sorted runs

Total cost: 3B(R) + 3B(S)

Run 1 of R		Run 2 of R	
1	2	1	2
3	4	3	5
4	5	7	9
6	7	11	11
8	9	12	14

Run 1 of S		Run 2 of S	
0	1	1	5
2	3	7	9
3	4	11	12
5	7		
8	9		

Memory M = 5 pages

1	2	Run1
1	2	Run2
2	3	Run1
1	5	Run2

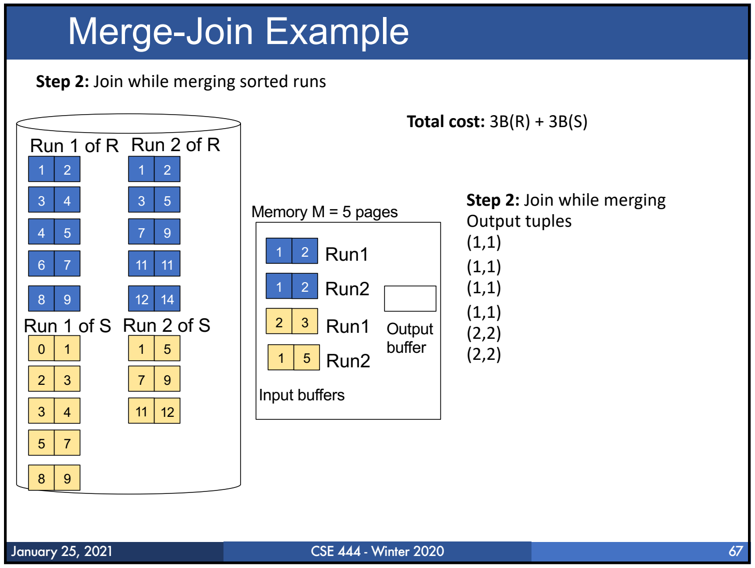
Input buffers

Output buffer

Step 2: Join while merging
Output tuples
(1,1)
(1,1)
(1,1)
(1,1)

January 25, 2021 CSE 444 - Winter 2020 66

66



Merge-Join Example

Step 2: Join while merging sorted runs

Total cost: 3B(R) + 3B(S)

Run 1 of R		Run 2 of R	
1	2	1	2
3	4	3	5
4	5	7	9
6	7	11	11
8	9	12	14

Run 1 of S		Run 2 of S	
0	1	1	5
2	3	7	9
3	4	11	12
5	7		
8	9		

Memory M = 5 pages

1	2	Run1
1	2	Run2
2	3	Run1
1	5	Run2

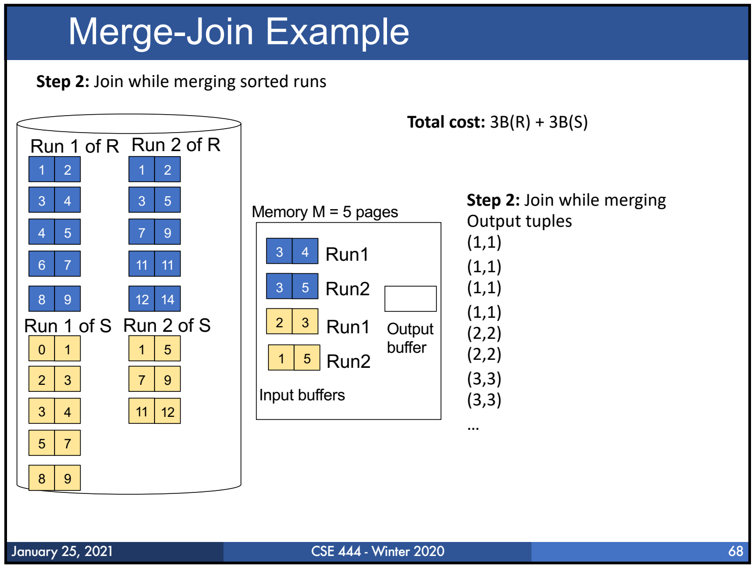
Input buffers

Output buffer

Step 2: Join while merging
Output tuples
(1,1)
(1,1)
(1,1)
(1,1)
(2,2)
(2,2)

January 25, 2021 CSE 444 - Winter 2020 67

67



Merge-Join Example

Step 2: Join while merging sorted runs

Total cost: 3B(R) + 3B(S)

Run 1 of R		Run 2 of R	
1	2	1	2
3	4	3	5
4	5	7	9
6	7	11	11
8	9	12	14

Run 1 of S		Run 2 of S	
0	1	1	5
2	3	7	9
3	4	11	12
5	7		
8	9		

Memory M = 5 pages

3	4	Run1
3	5	Run2
2	3	Run1
1	5	Run2

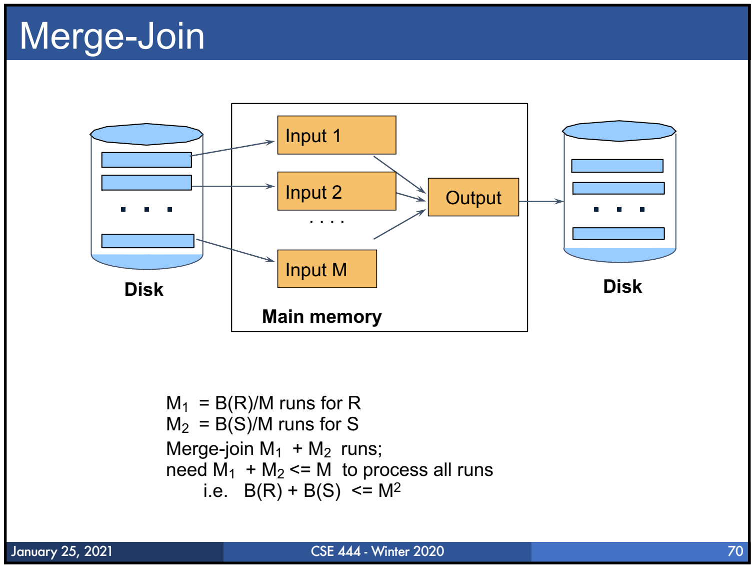
Input buffers

Output buffer

Step 2: Join while merging
Output tuples
(1,1)
(1,1)
(1,1)
(1,1)
(2,2)
(2,2)
(3,3)
(3,3)
…

January 25, 2021 CSE 444 - Winter 2020 68

68



Merge-Join

M_1 = B(R)/M runs for R
M_2 = B(S)/M runs for S
Merge-join $M_1 + M_2$ runs;
need $M_1 + M_2 <= M$ to process all runs
i.e. B(R) + B(S) <= M^2

January 25, 2021 CSE 444 - Winter 2020 70

70