

CSE 350

Advanced Data Structures

Topic 19: External Algorithms 2 - Joins

Recap: Joins

Δ_B

R	A	B	S	B	C
	1	2		2	4
	1	3		3	5
	2	2		3	6
	3	3			

Diagram illustrating a join operation between two tables R and S. Table R has columns A and B, and Table S has columns B and C. The join is performed on the common attribute B. The result is shown by connecting rows from R to rows in S that share the same value in the B column. Red lines connect R(1,2) to S(2,4). Orange lines connect R(1,3) to S(3,5) and R(2,2) to S(3,6). A third orange line connects R(3,3) to S(3,6), indicating a join on the value 3 in the B column.

Nested Loop Join

Condition



for r in R :

for s in S :

if condition(r, s):

output(ros)

read all of R

read all of S for each row in R

$|R|$

$|S|$

Runtime $O(|R| \cdot |S|)$

$O(1)$ mem

→ T_0 $|R| + |R| \cdot |S|$ if $|S| > \text{mem}$

← read a block of records

for $[r_1, \dots, r_B]$ in R

for s in S :

for i in 1 to B

if condition(r_i, s)

output(r_i, s)

with $B = 100$

R $\overbrace{1\ 2\ 3\ 4\ \dots}^{100}$ $\overbrace{101\ 102\ 103\ \dots}^{100}$

$$\frac{|R|}{B}$$

$\frac{|R|}{B}$ times

Block
Nested
Loop Join

$$I/O: |R| + \frac{|R|}{B} |S|$$

↑ $O(B)$ mem

runtime

$$\frac{|R|}{B} \cdot |S| \cdot \cancel{B} \cdot O(1) = O(|R| \cdot |S|)$$

(same)

1-Pass Hash Join

Insight: Condition usually has the form $\underline{R.A = S.A}$
(or $\underline{R.A = S.A}$ and $R.B = S.B, \rightarrow (R.A, R.B) = (S.A, S.B)$)

$O(1)$ amortized per record

$R \cdot q_1$ [organize R on A] — create DS — Hashmap
w/ A as a key

for s in S :

$S \cdot q_1 \rightarrow$ [for r in elements of R w/ same A :]
[output($r \cdot s$)] $\uparrow$ use DS
] $|S|$
degree of s

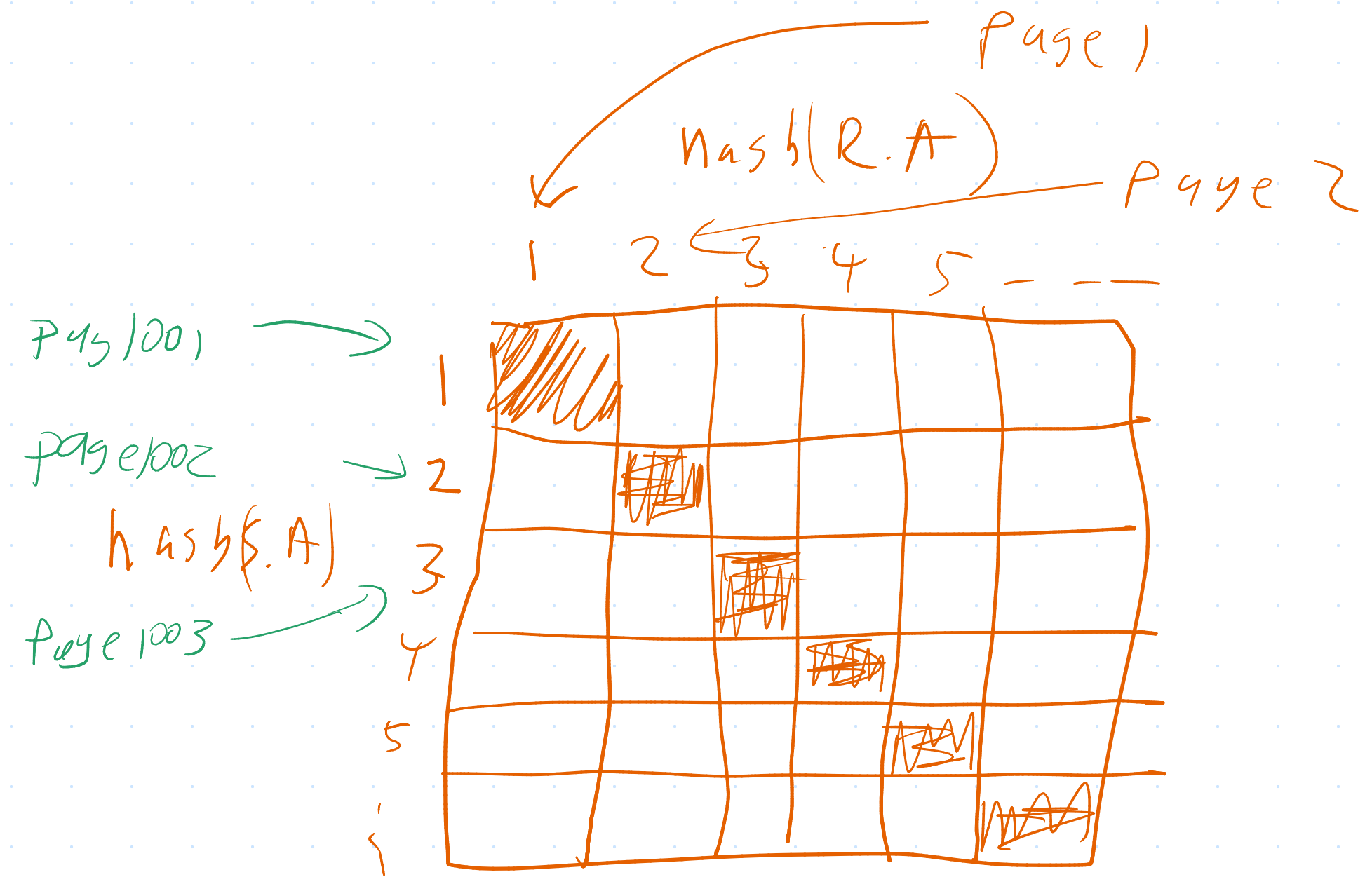
assuming 1
 q_1 record matched $O(1)$ expected

$$\text{Runtime } O(|R| + |S| + |R \times S|)$$

- IO scan over $R \rightarrow \text{read } |R|$
- ↳ Build Hashmap
 - ↳ one read + one write for each record in R
 - ↳ load bucket, insert, write back
 - ↳ Scan over $S \rightarrow \text{read } |S|$
 - ↳ Use Hashmap
 - ↳ one read for each record in S
- Caching reduces this

2-Pass Hash Join

⤴ "Grace" Hash Join



$$|R|=|S|=N$$

$|R|+|S|$ data written $\left[\begin{array}{l} \text{write } R \text{ out to on-disk hash table w/ } B \text{ buckets} \\ \text{write } S \text{ out to on-disk hash table w/ } B \text{ buckets} \end{array} \right] O(B) \text{ mem needed}$

for i in 1 to B :

$|R|+|S|$ data read $\left[\begin{array}{l} \text{for } r \text{ in bucket } i \text{ of } R \\ \text{for } s \text{ in bucket } i \text{ of } S \\ \text{if } r.A = s.A \\ \text{output}(r, s) \end{array} \right] \begin{array}{l} \text{mem needed } \frac{N}{B} \\ O(2 \cdot |\text{bucket}|) \\ \downarrow \\ \text{each bucket read only once} \end{array}$

$$\rightarrow 2(|R|+|S|) I/O_s$$

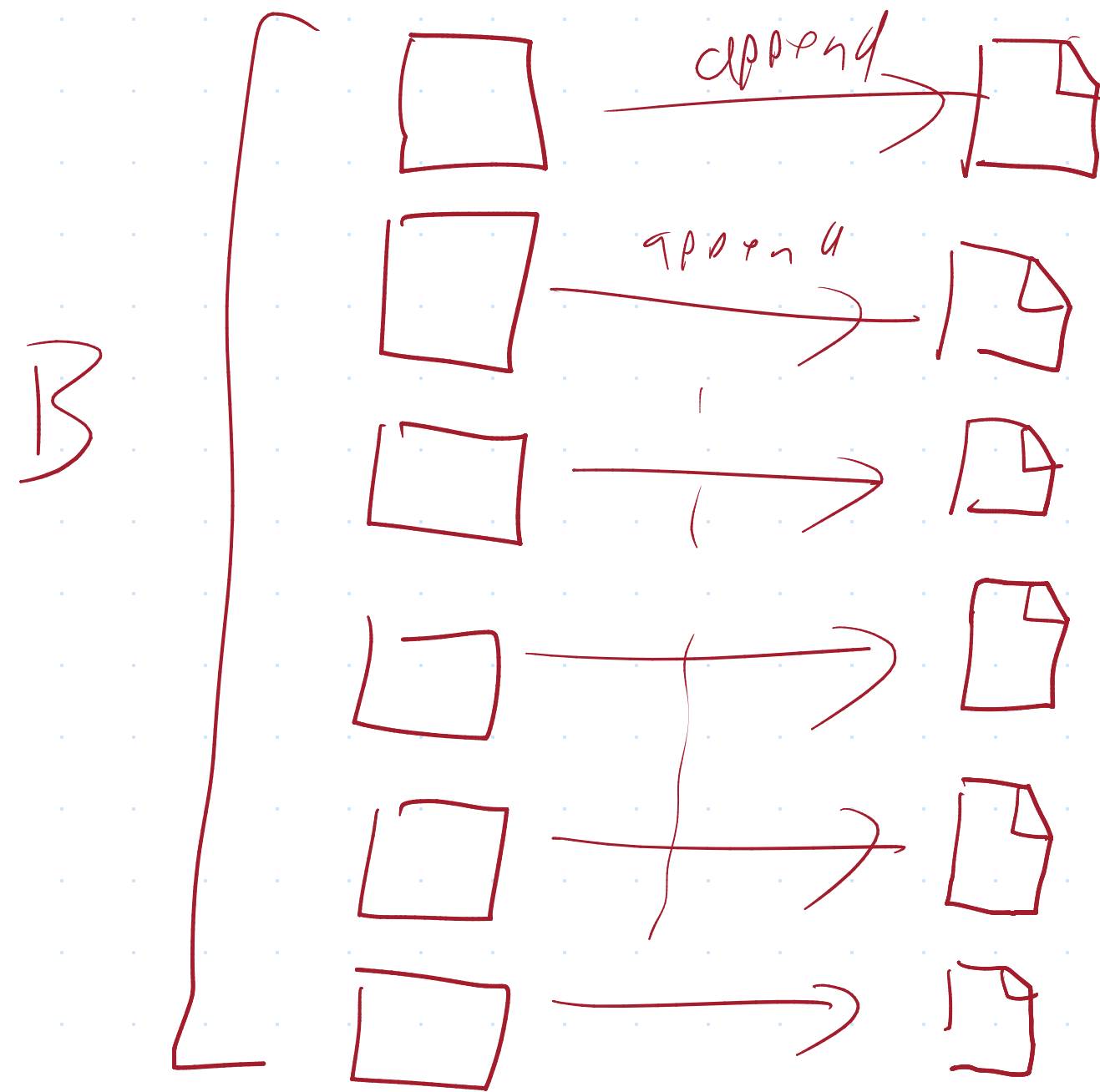
Given M memory

$$\sum \frac{N}{B} < M$$

$$\hookrightarrow \sum \frac{N}{M} < B$$

Buffers

Data on Disk



to create B buckets

allocate B buffers

for each record r in R

append to buffer $hash(r.A)$

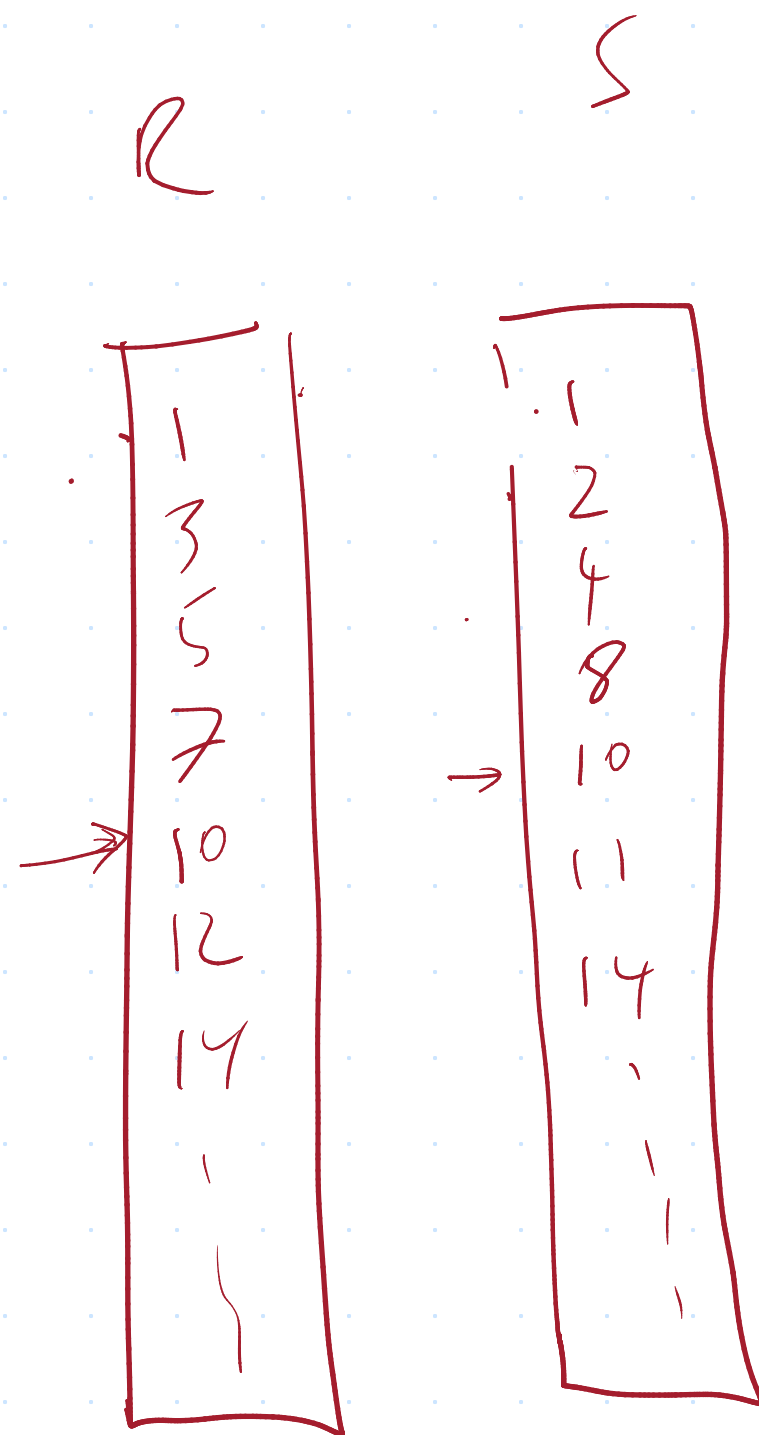
if buffer full, flush &

allocate new buffer

$O(B)$ mem

$|R|$ writes + $O(B)$

Sort-Merge Join



$\langle 1, 1 \rangle$

$\langle 10, 10 \rangle$

Runtime $O(|R| + |S|)$

IO

$|R| + |S|$

Mem

$O(1)$

while R & S have more

if next R.A = next S.A

output it

advance R

advance S

else if next R.A < next S.A

advance R

else

advance S

⚡ Data must already be sorted

R

S

