Introduction to Data Management CSE 344

Lecture 12: Cost Estimation Relational Calculus

Announcements

- WQ3 due tomorrow
- HW3 due Wednesday

- WQ4 and HW4 will be out this week
  - 1 week HW on RA, RC, and Datalog

- Style guide / common mistakes listed on course website
  - See link under Assignments

Midterm

- Monday, November 7th in class
  - Location TBD
- Contents
  - Lectures and sections through November 4th
  - Homework 1 through 4
  - Webquiz 1 through 4
- Closed book. No computers, phones, watches, etc.!
- Can bring one letter-sized piece of paper with notes
  - Can write on both sides
  - You might want to save it for the final

How to Study?

- Lecture slides and section materials
- Homework 1 through 4
- Past midterms posted on website
  - Lots of great examples! With solutions
  - But content changes between quarters
    - So some questions may not apply
    - We may have some new questions not present in past
- Practice Webquiz on gradiance

Today's Outline

- Finish cost estimation
- Relational calculus

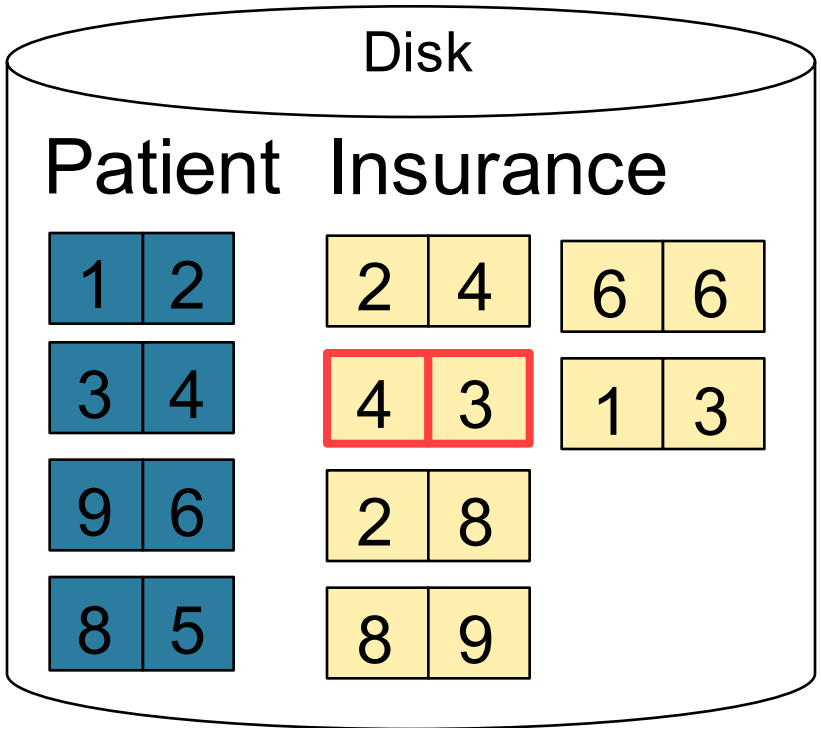
Review

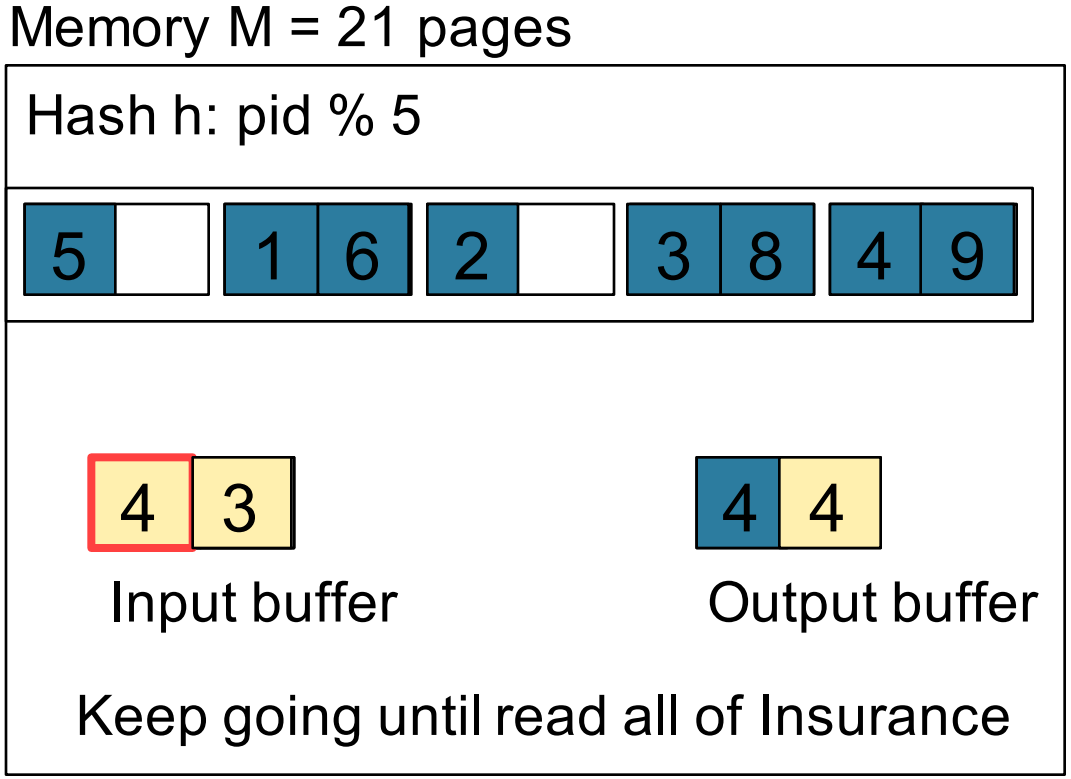
- Estimate cost of physical query plans
  - Based on # of I/O operations
  - Estimate cost for each operator
  - Cost of entire plan = Σ operator cost
- Cost for selection operator
  - Indexed and non-indexed
- Cost for join operator
  - Hash join
  - Nested loop join

Review: Hash Join Example

Step 2: Scan Insurance and probe into hash table

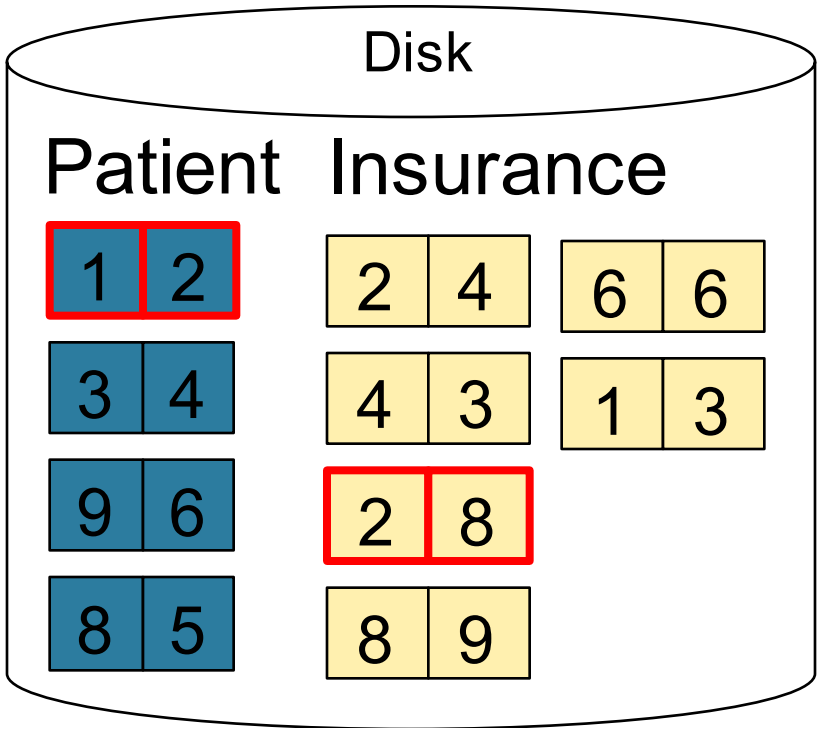
Done during calls to next()

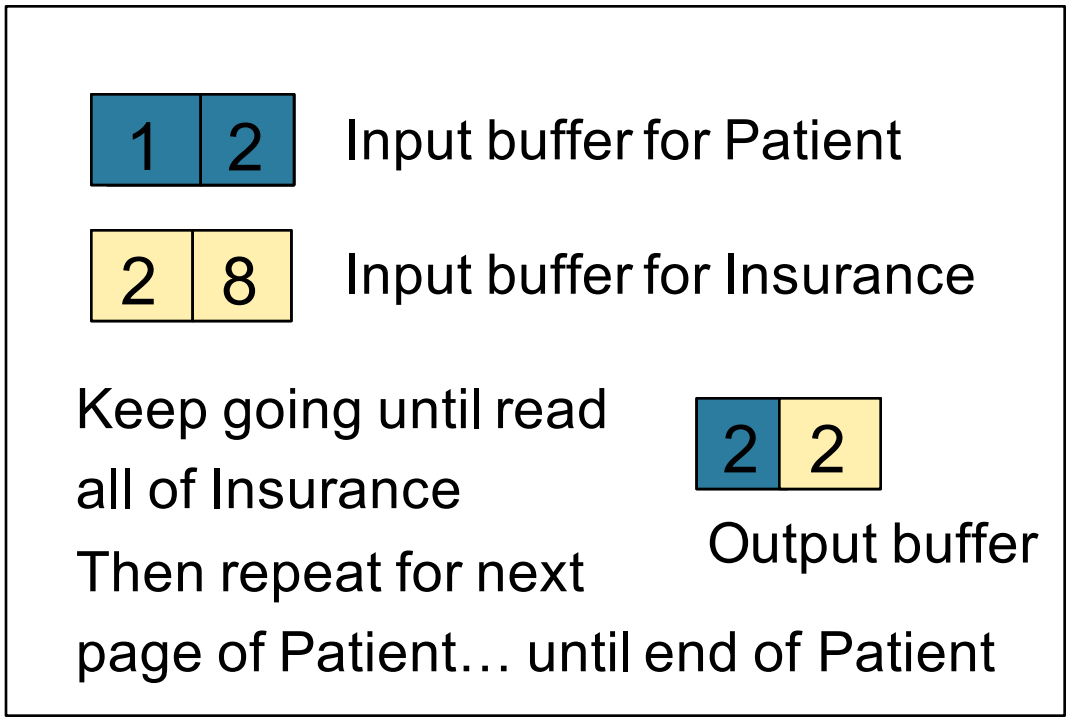




Cost: B(R) + B(S)

Review: Nested Loop: Page-at-a-time Refinement





Cost: B(R) + B(R)B(S)

Block-Nested-Loop Refinement

$R \bowtie S$

```
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 t1 in r, t2 in s
               if t1 and t2 join then output (t1,t2)
```

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

What is the Cost?

Index Nested Loop Join

R ⋈ S

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

- Cost:
  - If index on S is clustered:
    B(R) + T(R) * (B(S) * 1/V(S,a))
  - If index on S is unclustered:
    B(R) + T(R) * (T(S) * 1/V(S,a))

Sort-Merge Join

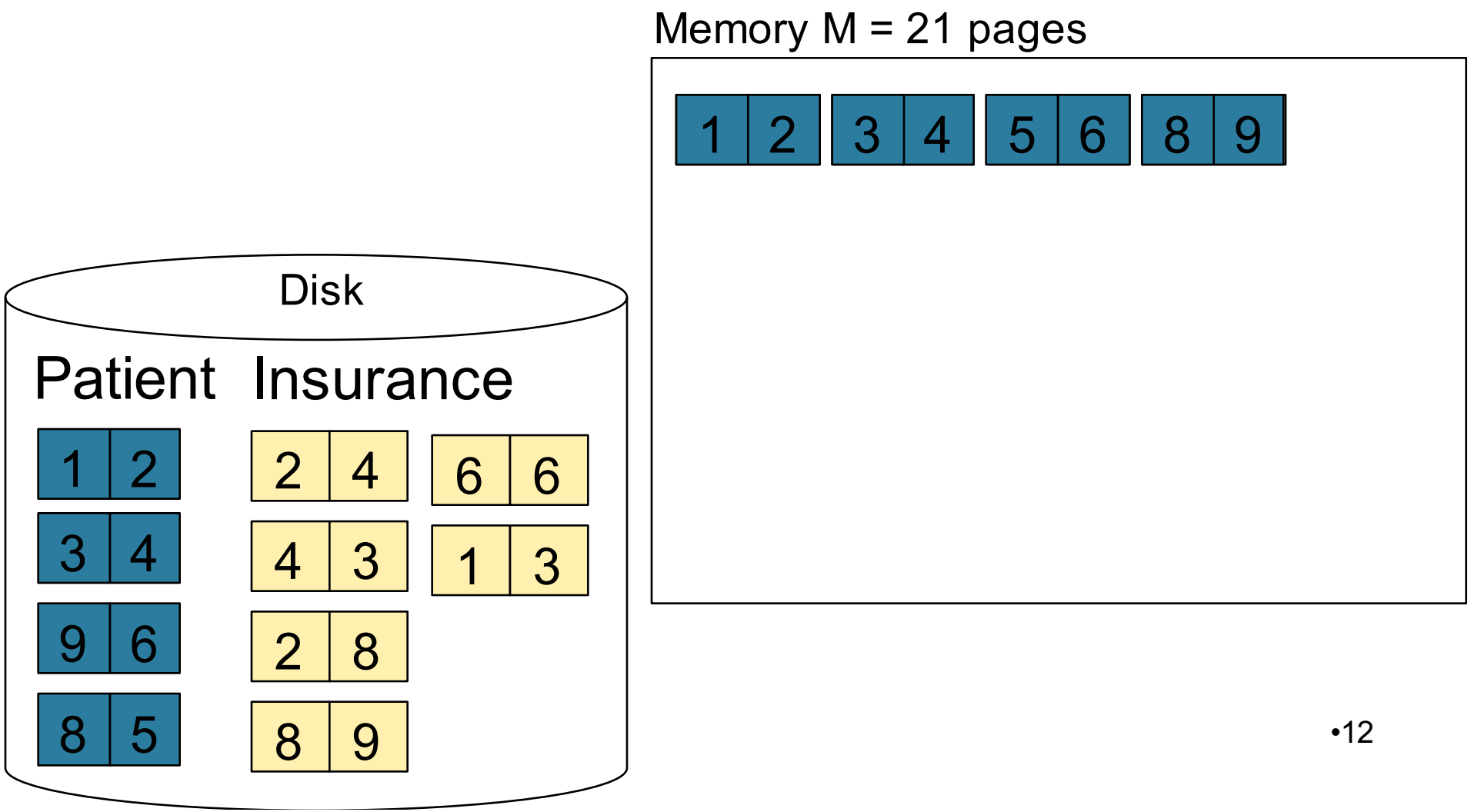
Sort-merge join: $R \bowtie S$

- 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

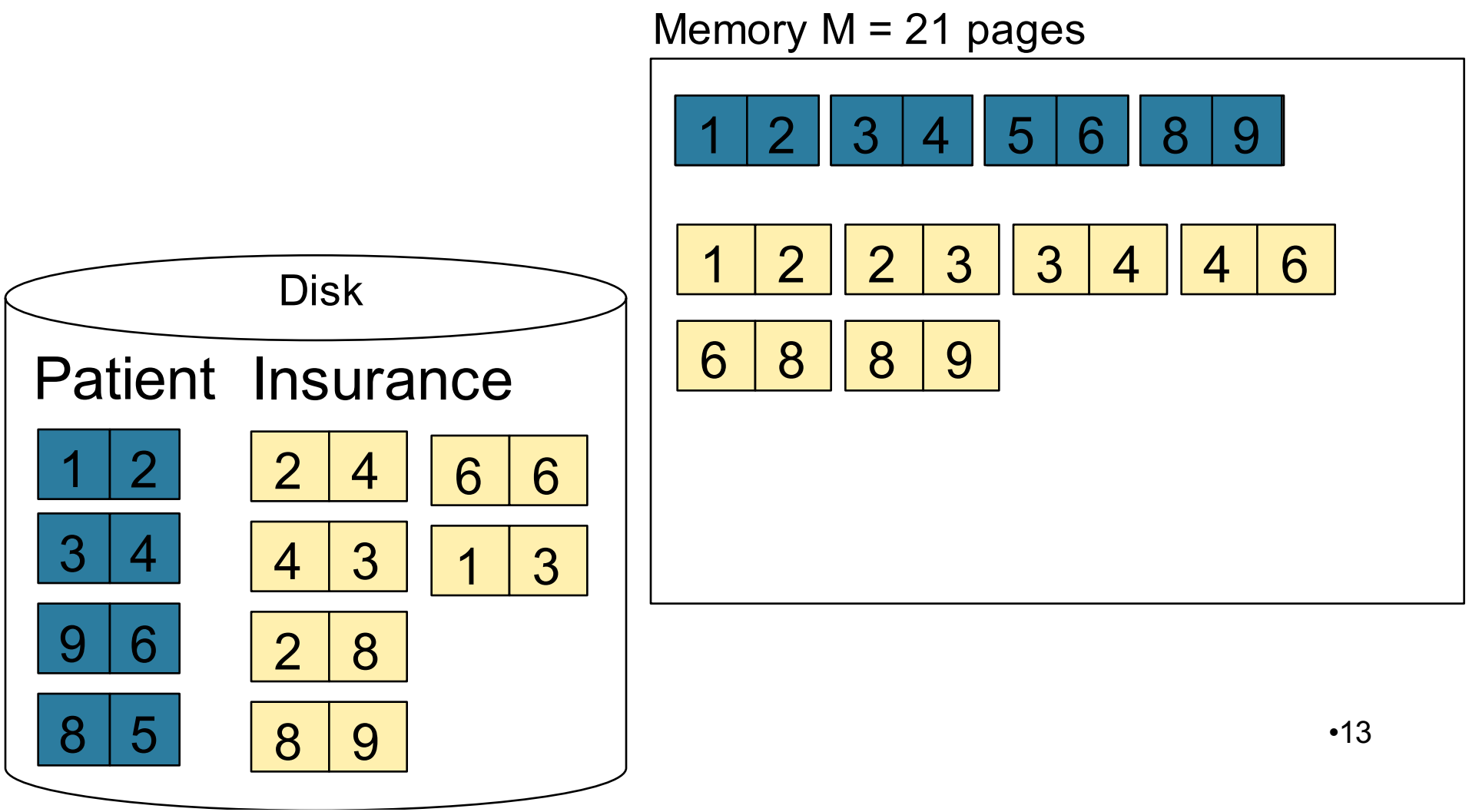
Sort-Merge Join Example

Step 1: Scan Patient and sort in memory



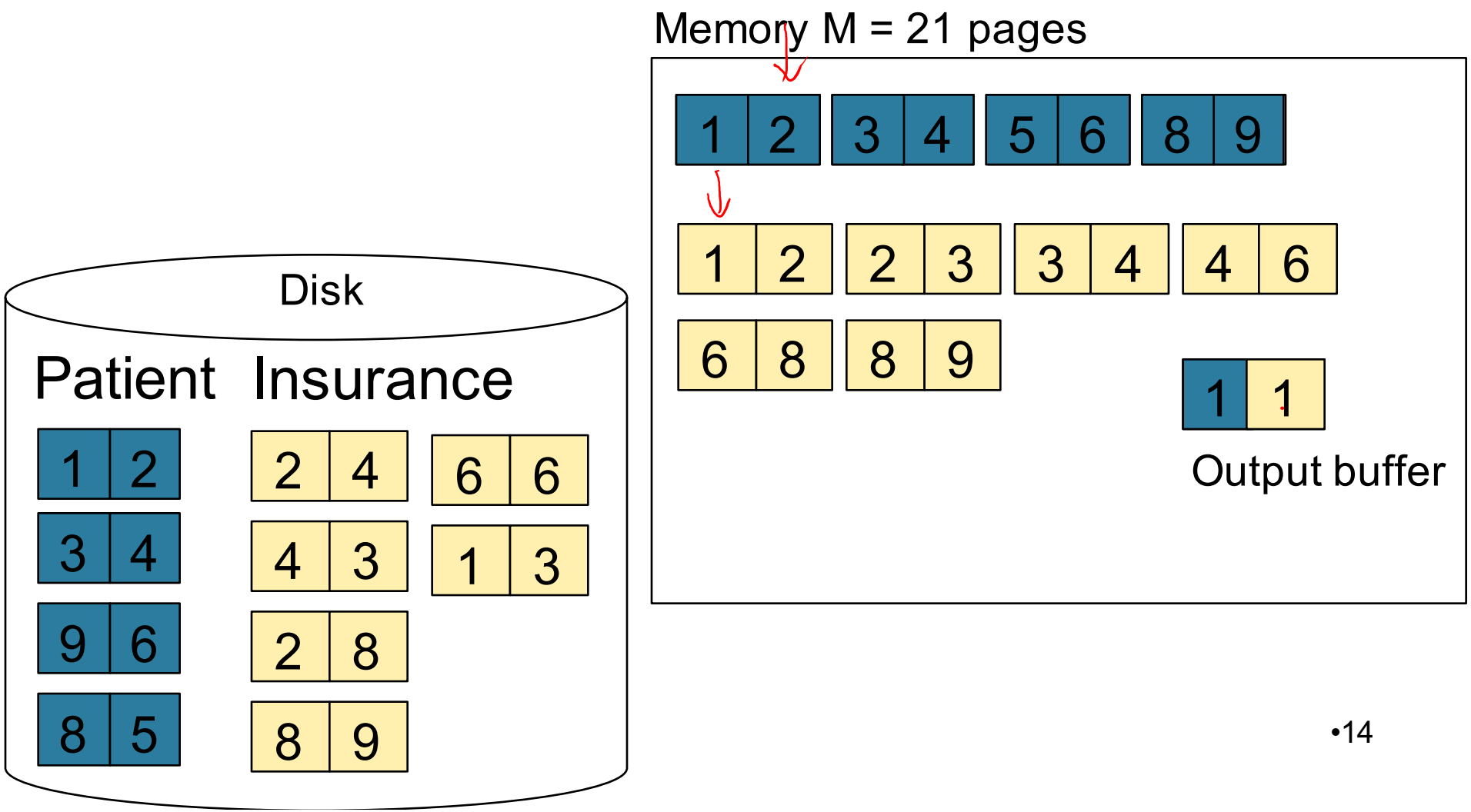
Sort-Merge Join Example

Step 2: Scan Insurance and sort in memory



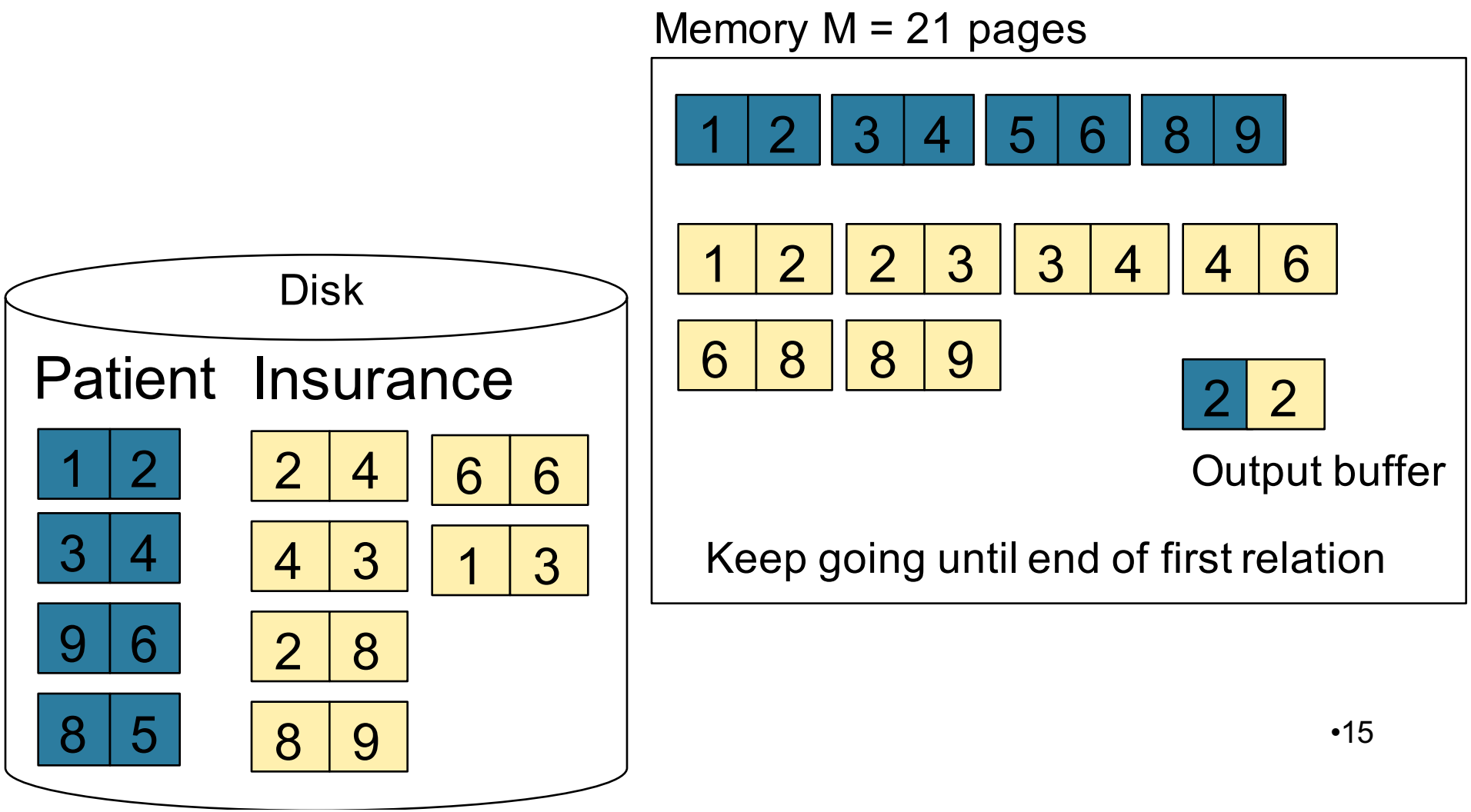
Sort-Merge Join Example

Step 3: Merge Patient and Insurance



Sort-Merge Join Example

Step 3: Merge Patient and Insurance



Cost of Query Plans

T(Supplier) = 1000
T(Supply) = 10,000

B(Supplier) = 100
B(Supply) = 100

V(Supplier,scity) = 20
V(Supplier,state) = 10
V(Supply,pno) = 2,500

M = 11

Physical Query Plan 1

(On the fly) π_{sname}

Selection and project on-the-fly
→ No additional cost.

(On the fly) $\sigma_{scity='Seattle' \text{ and } sstate='WA' \text{ and } pno=2}$

(Nested loop) sid = sid

Total cost of plan is thus cost of join:
= B(Supplier)+B(Supplier)*B(Supply)
= 100 + 100 * 100
= 10,100 I/Os

Supplier
(File scan)

Supply
(File scan)

```
SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
    and y.pno = 2
    and x.scity = 'Seattle'
    and x.sstate = 'WA'
```

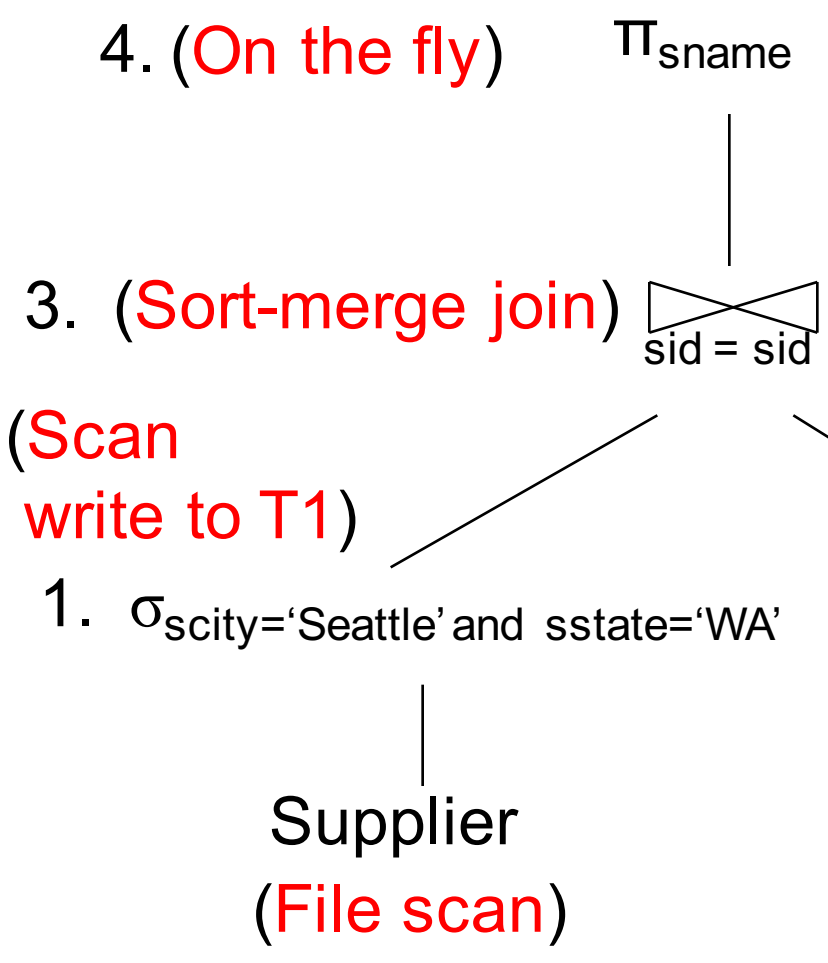
T(Supplier) = 1000
T(Supply) = 10,000

B(Supplier) = 100
B(Supply) = 100

V(Supplier,scity) = 20
V(Supplier,state) = 10
V(Supply,pno) = 2,500

M = 11

Physical Query Plan 2



file scan read cost

read cost

write T_1 to disk

write T_2 to disk

Total cost
= 100 + 100 * 1/20 * 1/10
(step 1)
+ 100 + 100 * 1/2500
(step 2)
+ 2
(step 3)
+ 0
(step 4)
Total cost ≈ **204 I/Os**

```
SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
    and y.pno = 2
    and x.scity = 'Seattle'
    and x.sstate = 'WA'
```

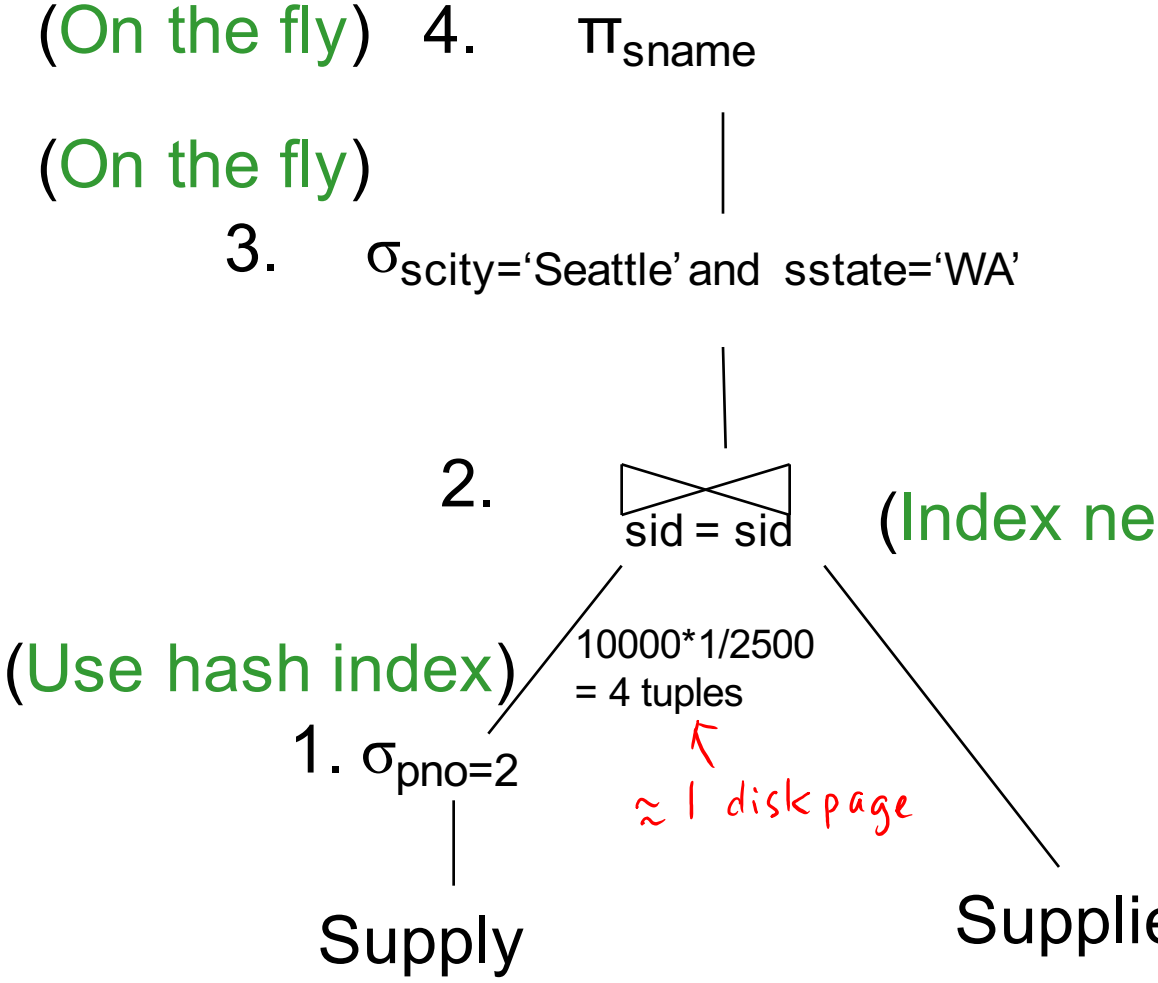
T(Supplier) = 1000
T(Supply) = 10,000

B(Supplier) = 100
B(Supply) = 100

V(Supplier,scity) = 20
V(Supplier,state) = 10
V(Supply,pno) = 2,500

M = 11

Physical Query Plan 3



Total cost
= 1 (step 1.)
+ 4 (step 2.)
+ 0 (step 3.)
+ 0 (step 4.)

read tuples after looking up from index

Total cost ≈ **5 I/Os**

```
SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
    and y.pno = 2
    and x.scity = 'Seattle'
    and x.sstate = 'WA'
```

Query Optimizer Summary

- **Input**: A logical query plan
- **Output**: A good physical query plan
- **Basic query optimization algorithm**
  - Enumerate alternative plans (logical and physical)
  - Compute estimated cost of each plan
    - Compute number of I/Os
    - Optionally take into account other resources
  - Choose plan with lowest cost
  - This is called cost-based optimization

Big Picture

- Relational data model
  - Instance
  - Schema
  - Query language
    - SQL
    - Relational algebra
    - Relational calculus
    - Datalog
- Query processing
  - Logical & physical plans
  - Indexes
  - Cost estimation
  - Query optimization

Why bother with another QL?

- SQL and RA are good for query planning
  - They are not good for *formal reasoning*
  - How do you show that two SQL queries are equivalent / non-equivalent?
  - Two RA plans?
- RC was the first language proposed with the relational model (Codd)

Relational Calculus

- Aka *predicate calculus* or *first order logic*
  - 311 anyone?

- TRC = Tuple Relational Calculus
  - See book

- DRC = Domain Relational Calculus
  - We study only this one
  - Also see *Query Language Primer* on course website