Cost-based Query Optimization

Christoph Koch

Computing the cost of a query plan

- We know how to estimate the cost of each individual operator.
  - But for doing this we need to know the size of the input.
- Compute the size of each relation produced by some operator
  - Bottom-up
  - We need to know estimates of selectivities/reduction factors for both selection and join conditions.
- Given a fixed query plan, if we exchange a particular operator implementation (e.g. NL join against Hashjoin), the output does not change and we do not have to recompute output sizes.

Model of Query plans

- We use a very powerful model.
- Ingredients:
  - Algebra tree
  - For each operation, a choice of implementation
  - For some binary operations (e.g. NL joins), an annotation saying which of the two inputs is the outer and which is the inner loop.
  - A choice of whether an operator's output is to be written back to disk or pipelined into the next operator (sometimes there is no choice).
  - An allocation of memory buffer pages to operators and their input/output lines.
    - In case of pipelining, the allocation may not be operator by operator but span several operations.

A Remark on the cost function for block NL joins

- On this slides I use the formula

  pages_outer +
  round_up(pages_outer / (buffer_pages – 1)) *
  (pages_inner - 1) + 1

- I explained this formula in class but it’s not in the book.
- You can use the formula from the book instead:

  pages_outer +
  round_up(pages_outer / (buffer_pages – 1)) *
  pages_inner

A University Database

Professor (p)			
id	name	grad	room
2125	Socrates	Full	226
2126	Russel	Full	232
2127	Kopernikus	Assoc	310
2133	Popper	Assoc	52
2134	Augustinus	Assoc	309
2136	Curie	Full	36
2137	Kant	Full	7

Student (s)		
sid	name	semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Takes (h)	
sid	cid
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

Course (v)			
cid	name	credits	taught_by
5001	Foundations	4	2137
5041	Ethics	4	2125
5043	Epistemology	3	2126
5049	Maieutics	2	2125
4052	Logics	4	2125
5052	Theory of Science	3	2126
5216	Bioethics	2	2126
5259	The Vienna Circle	2	2133
5022	CS432	2	2134
4630	The three critiques	4	2137

Assumptions

- Relation sizes (#tuples)
  - |p|=800
  - |s|=38000
  - |v|=2000
  - |h|=60000
- Avg. Tuple size
  - p: 50 Bytes
  - s: 50 Bytes
  - v: 100 Bytes
  - h: 16 Bytes
- selectivities
  - Sel[sh] = 2.6 * 10^-5
  - Sel[hv] = 5 * 10^-4
  - Sel[vp] = 1.25 * 10^-3
  - Sel[p.Name=…] =1.25 * 10^-3
- Page size 1024 Bytes
- Main memory buffers m = 20+1 pages

#tuples per page:

p: $\lfloor 1024/50 \rfloor$ =20

s: $\lfloor 1024/50 \rfloor$ = 20

v: $\lfloor 1024/100 \rfloor$ = 10

h: $\lfloor 1024/16 \rfloor$ = 64

(page overhead is ignored)

#pages:

p: 800/20 = 40

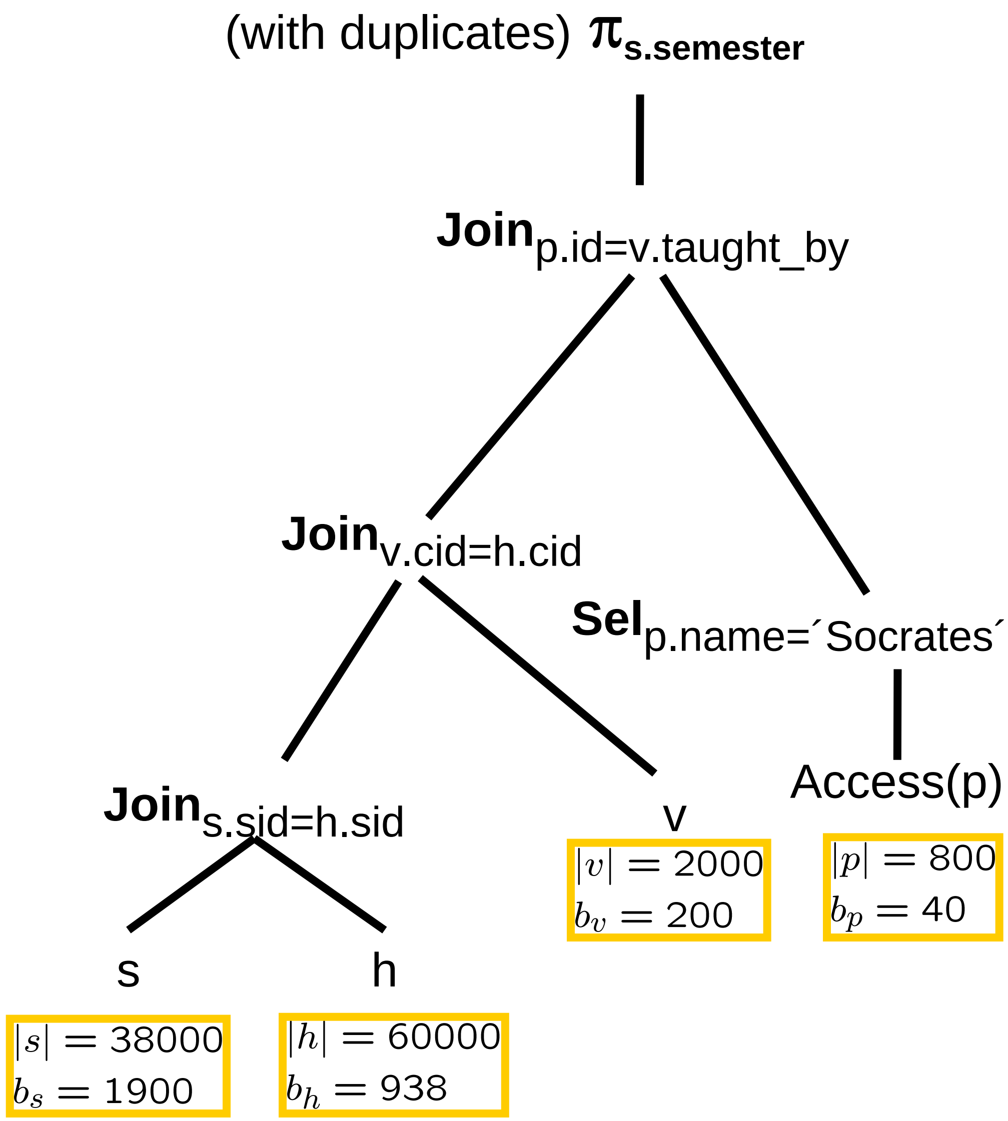
s: 38000/20 = 1900

v: 2000/10 = 200

h: $\lceil 60000/64 \rceil$ = 938

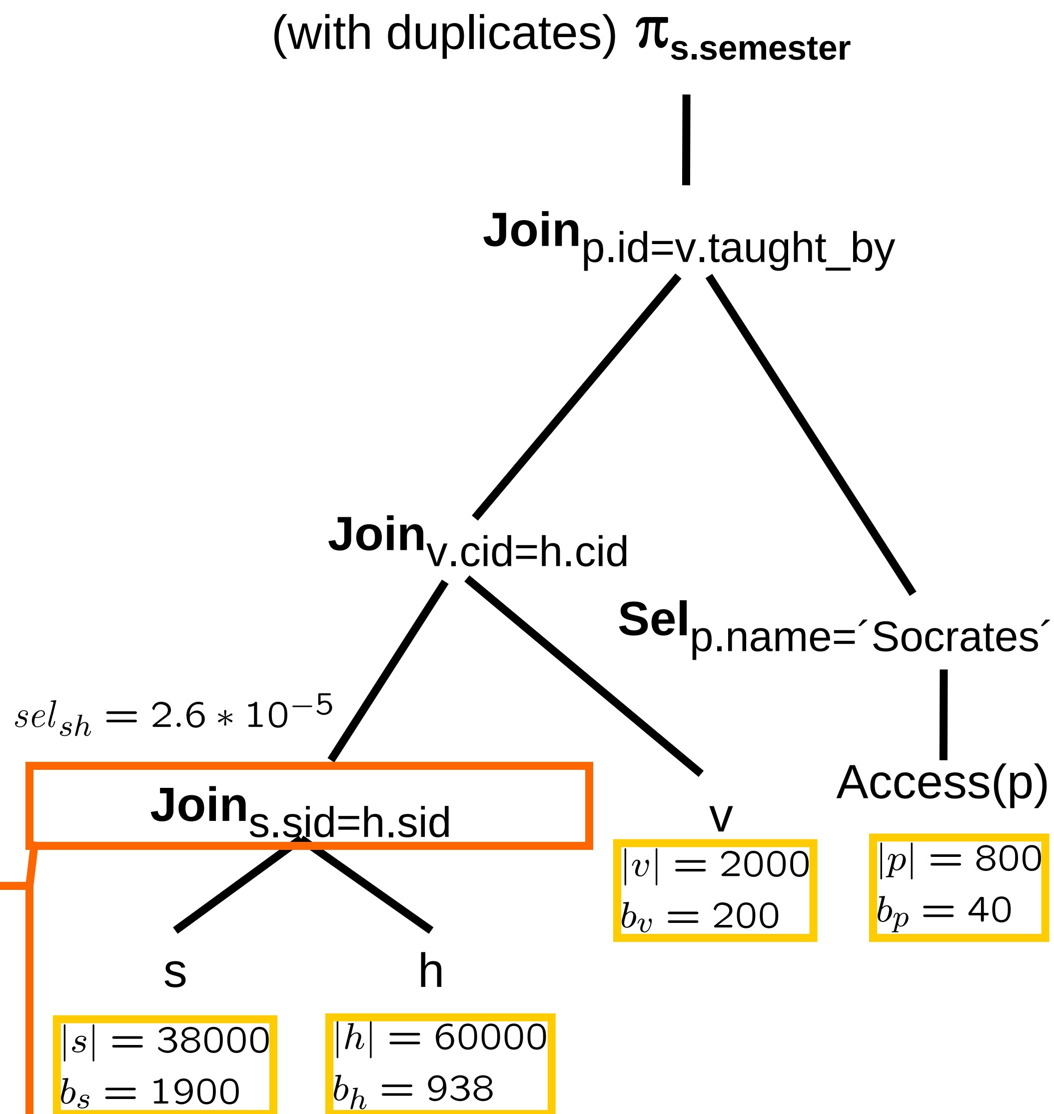
(we ignore the overhead of the primary index.)

Example 1



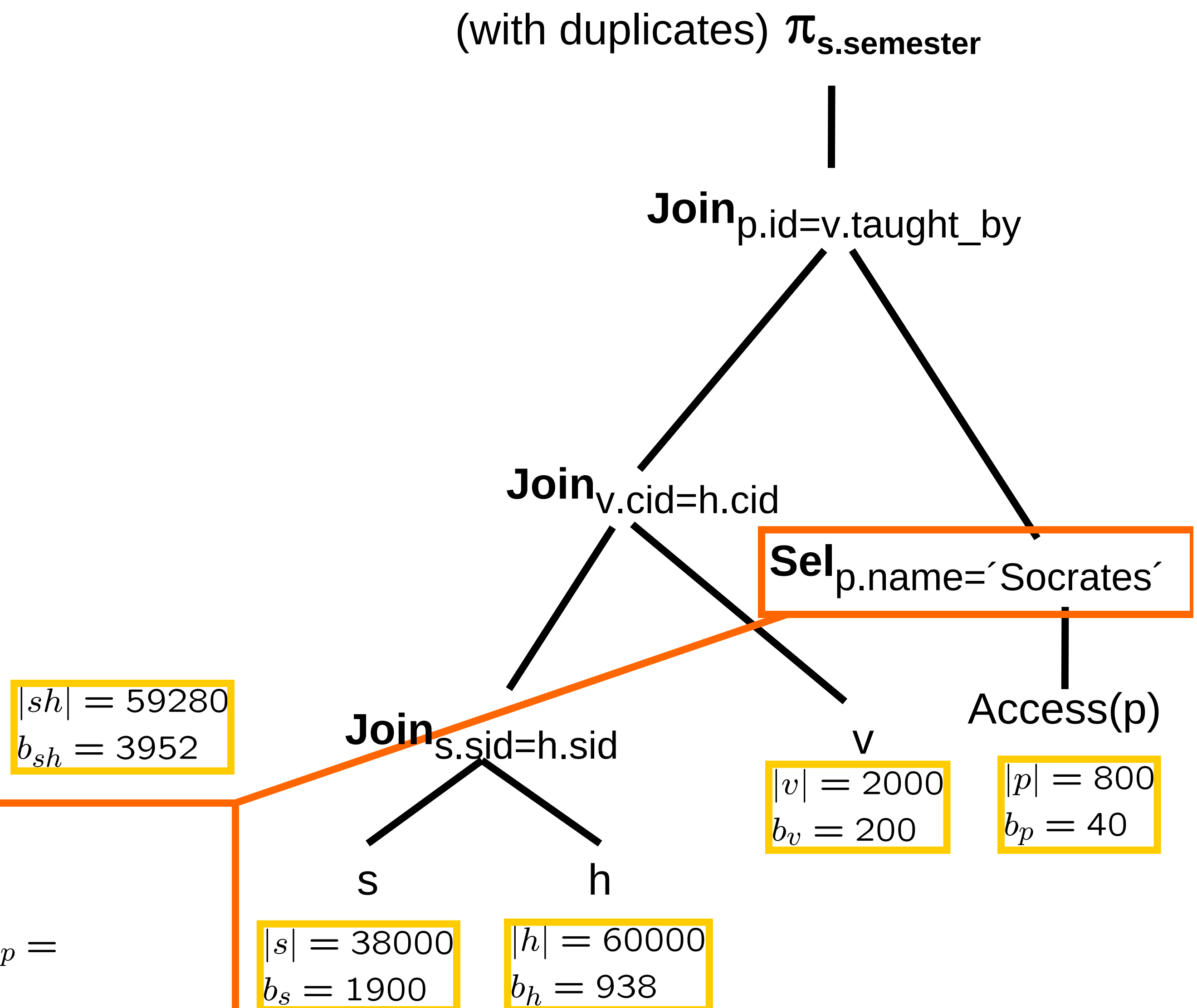
(with duplicates) $\pi_{\text{s.semester}}$
$\textbf{Join}_{\text{p.id=v.taught\_by}}$
$\textbf{Join}_{\text{v.cid=h.cid}}$
$\textbf{Sel}_{\text{p.name=´Socrates´}}$
$\textbf{Join}_{\text{s.sid=h.sid}}$
v
Access(p)
$|v| = 2000$
$b_v = 200$
$|p| = 800$
$b_p = 40$
s
h
$|s| = 38000$
$b_s = 1900$
$|h| = 60000$
$b_h = 938$

Example 1



#tuples: $|sh| = |h| * |s| * sel_{sh} =$
$38000 * 60000 * 2.6 * 10^{-5} = 59280$
Tuple size: $50 + 16 = 66$ Bytes
#tuples/page: $\lfloor 1024/66 \rfloor = 15$
#pages: $b_{sh} = \lceil 59280/15 \rceil = 3952$

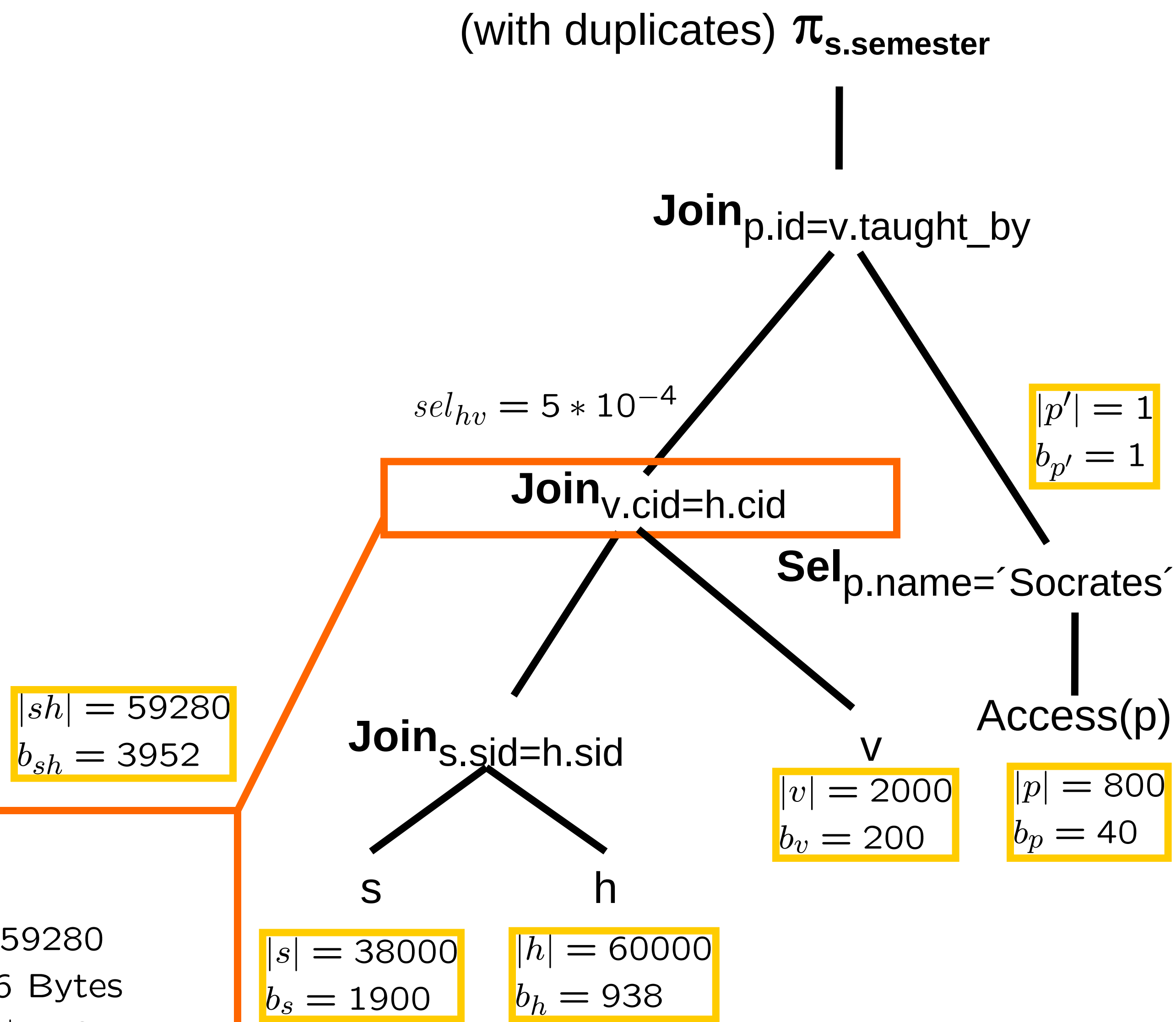
Example 1



#tuples: $|p'| = |p| * sel_p = 800 * 1.25 * 10^{-3} = 1$

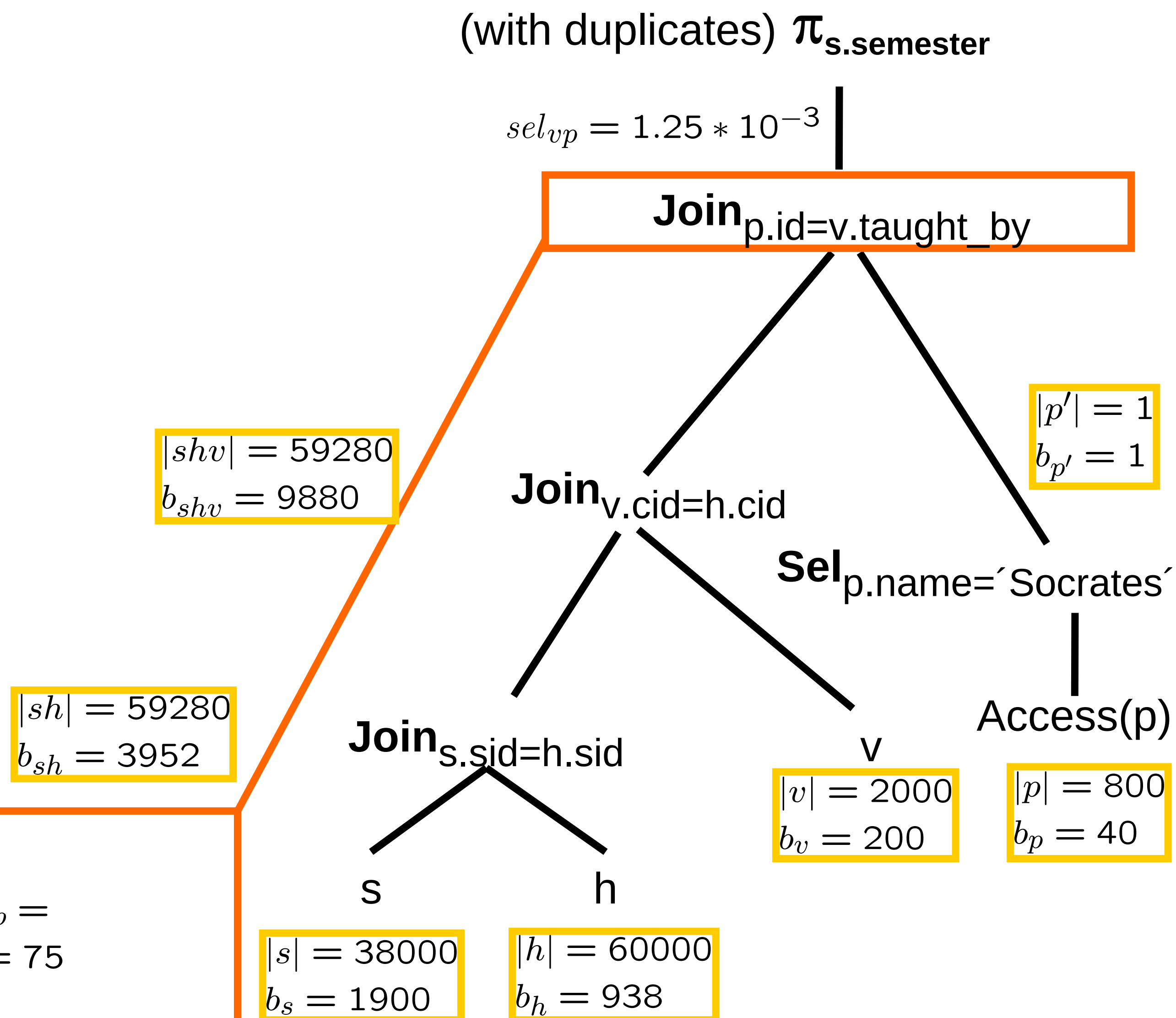
#pages: $b_{p'} = 1$

Example 1



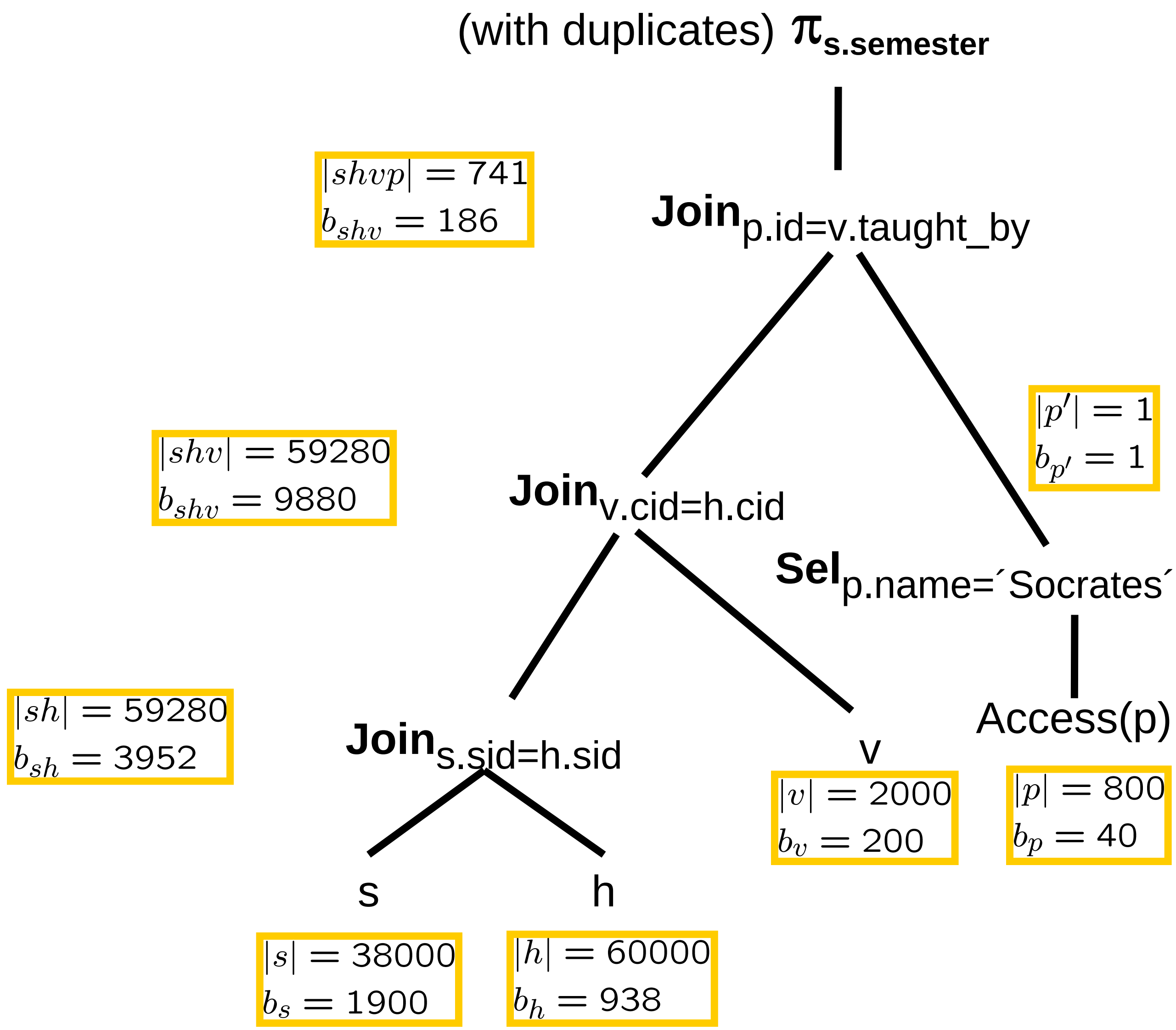
#tuples: $|sh| * |v| * sel_{hv} =$
$59280 * 2000 * 5 * 10^{-4} = 59280$
tuple size: $66 + 100 = 166$ Bytes
#tuples/page: $\lfloor 1024/166 \rfloor = 6$
#pages: $59280/6 = 9880$

Example 1



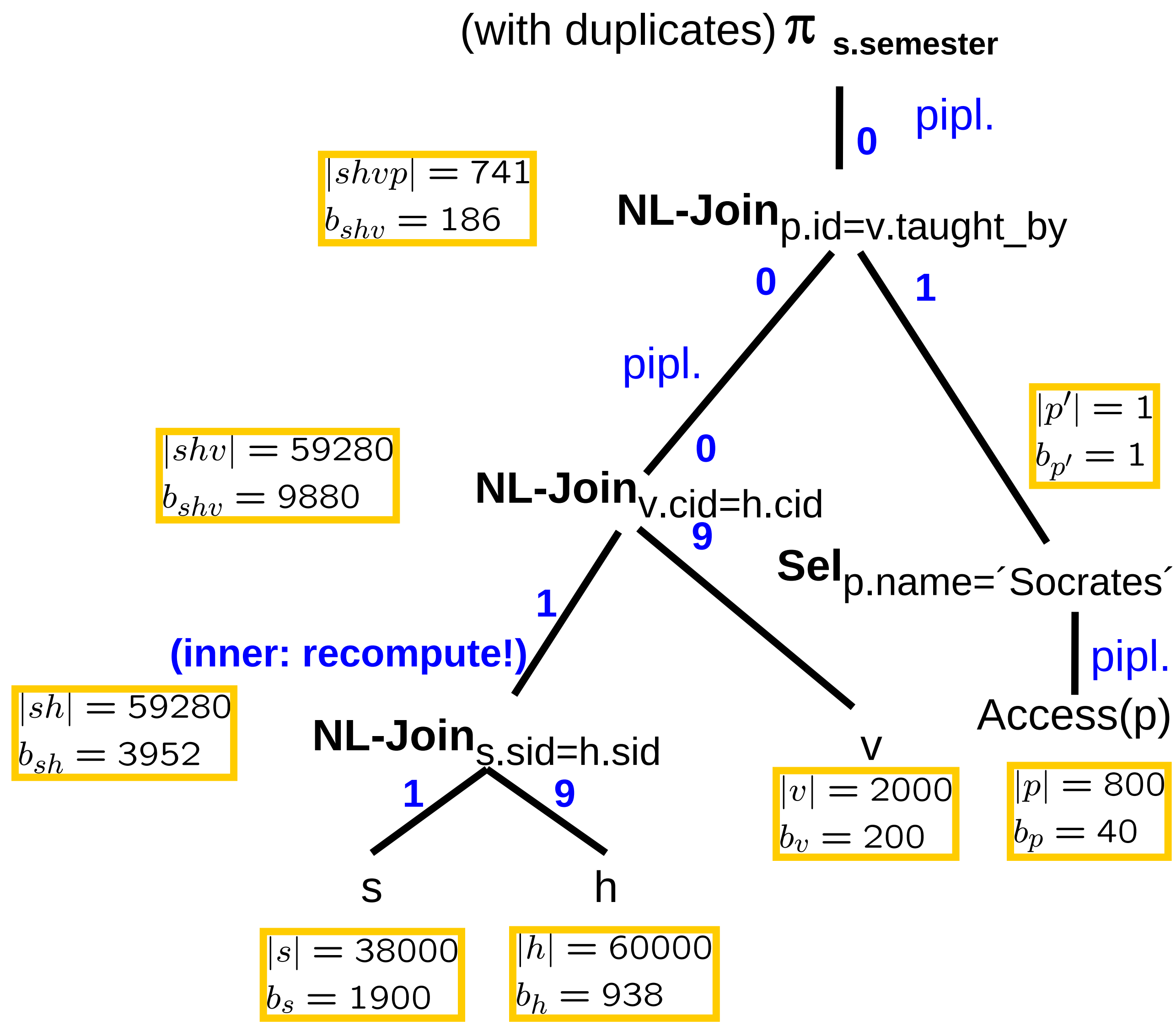
#tuples: $|shv| * |p| * sel_{vp} =$
$59280 * 1 * 1.25 * 10^{-3} = 75$
tuple size: 216 Bytes
#tuples/page: $\lfloor 1024/216 \rfloor = 4$
#pages: $\lceil 75/4 \rceil = 19$

Example 1



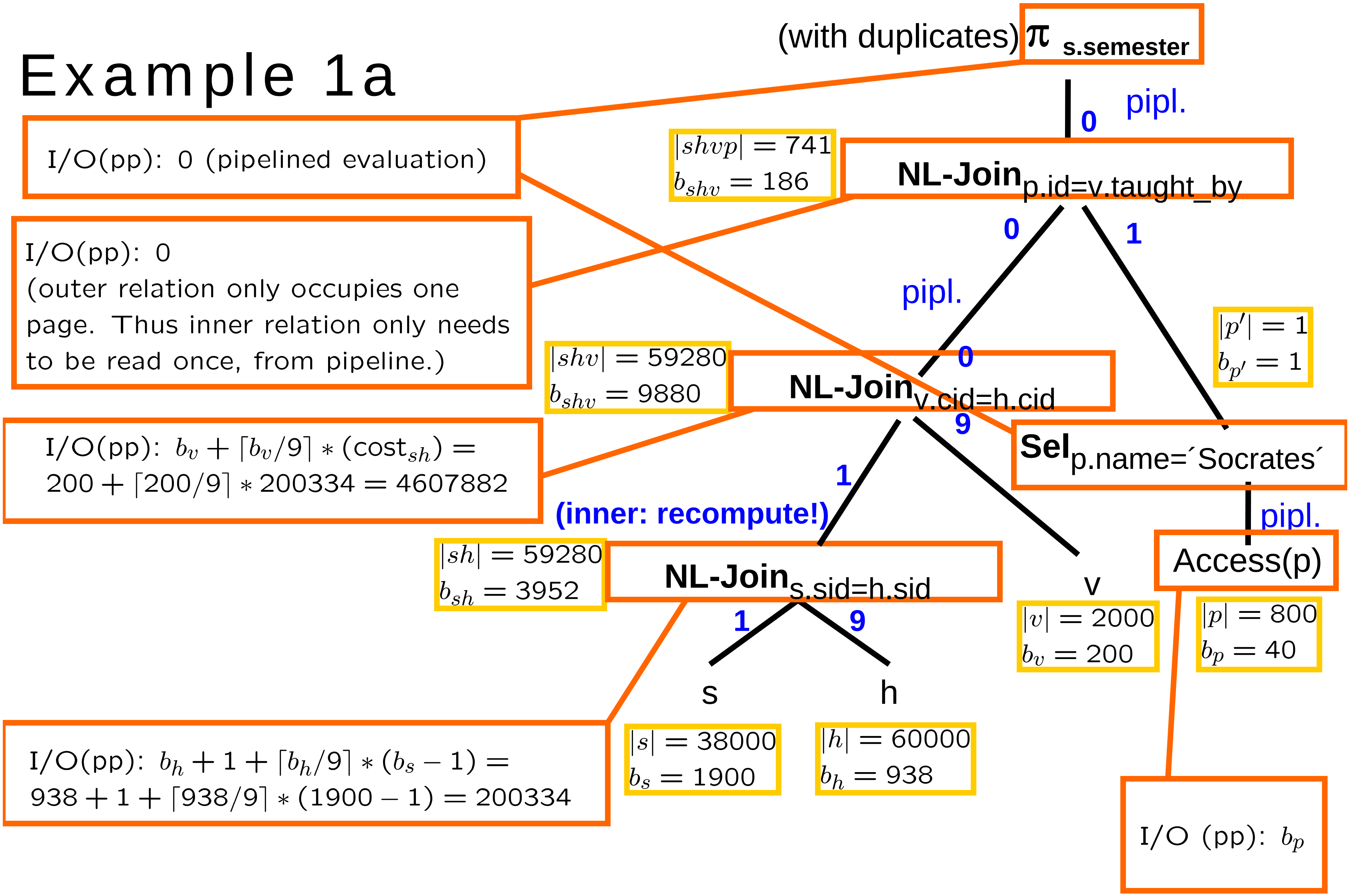
And now we have to allocate memory buffers...

Example 1a



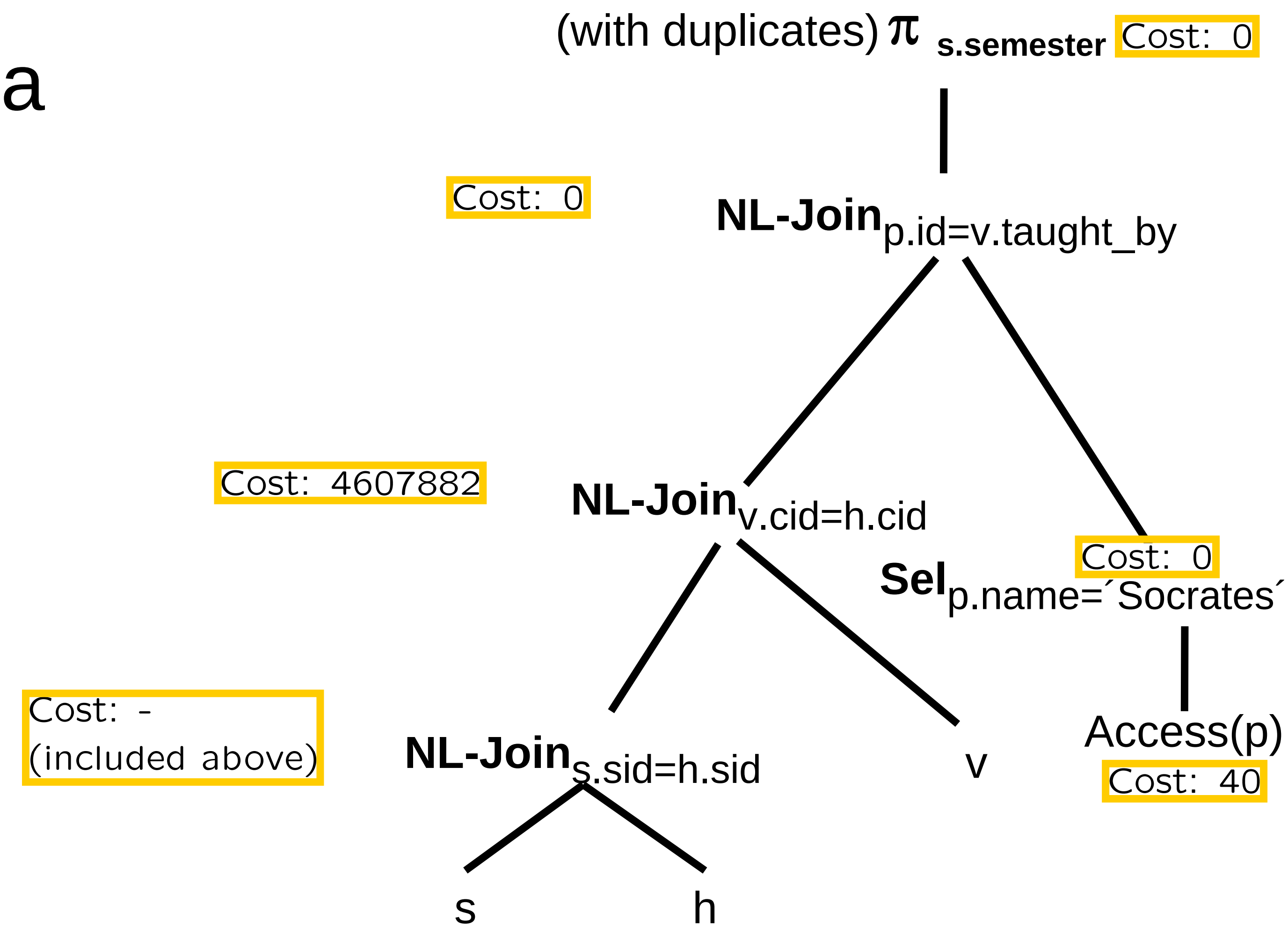
Buffer assignments in this color

Example 1a



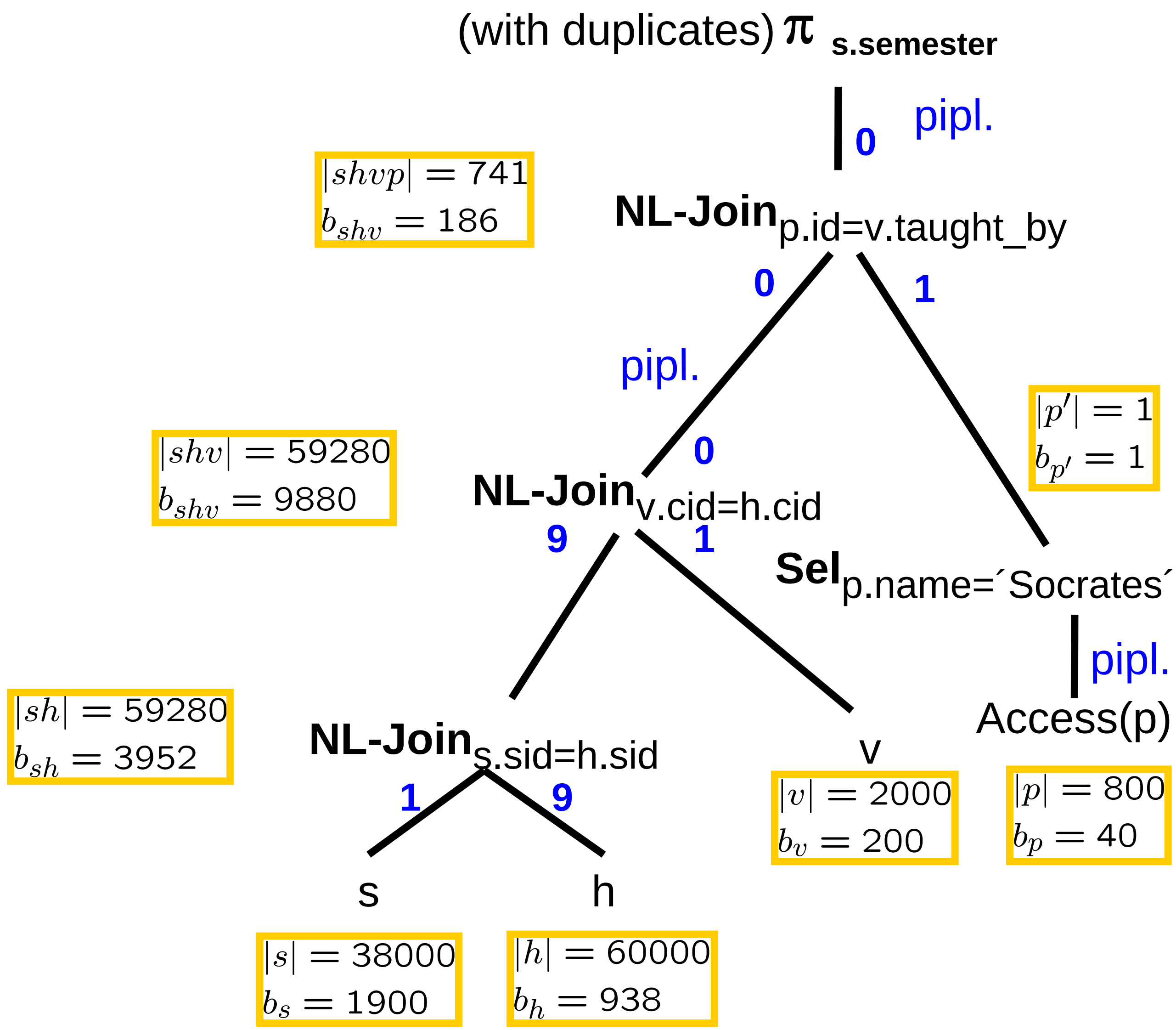
(with duplicates) π s.semester
0
pipl.
$|shvp| = 741$
$b_{shv} = 186$
NL-Join p.id=v.taught_by
I/O(pp): 0 (pipelined evaluation)
0
1
I/O(pp): 0
(outer relation only occupies one page. Thus inner relation only needs to be read once, from pipeline.)
pipl.
$|p'| = 1$
$b_{p'} = 1$
$|shv| = 59280$
$b_{shv} = 9880$
0
NL-Join v.cid=h.cid
9
Sel p.name=´Socrates´
I/O(pp): $b_v + \lceil b_v/9 \rceil * (\text{cost}_{sh}) =$
$200 + \lceil 200/9 \rceil * 200334 = 4607882$
1
(inner: recompute!)
pipl.
$|sh| = 59280$
$b_{sh} = 3952$
NL-Join s.sid=h.sid
Access(p)
v
1
9
$|v| = 2000$
$b_v = 200$
$|p| = 800$
$b_p = 40$
s
h
I/O(pp): $b_h + 1 + \lceil b_h/9 \rceil * (b_s - 1) =$
$938 + 1 + \lceil 938/9 \rceil * (1900 - 1) = 200334$
$|s| = 38000$
$b_s = 1900$
$|h| = 60000$
$b_h = 938$
I/O (pp): b_p

Example 1a



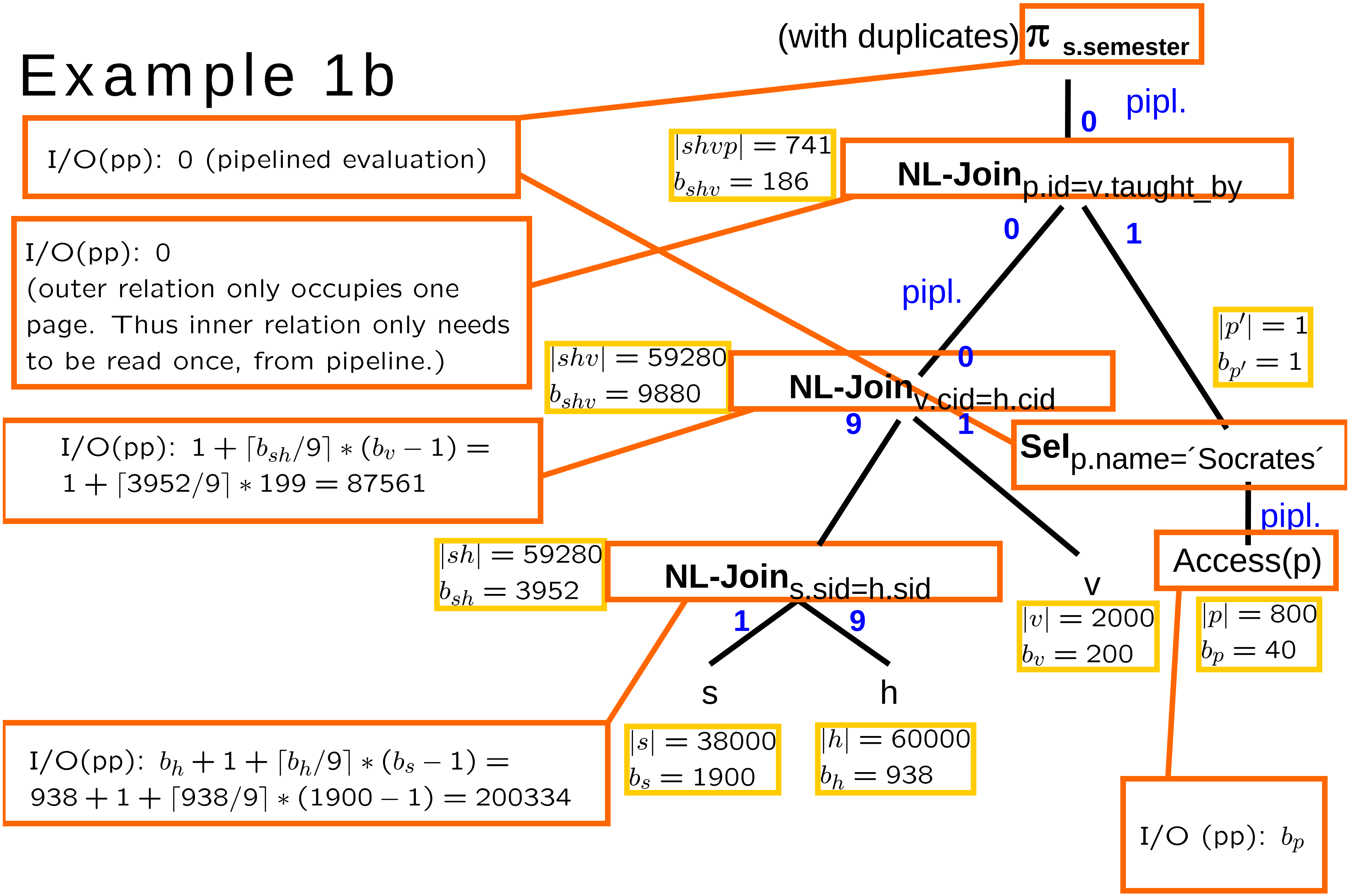
Total cost: 4607922

Example 1b



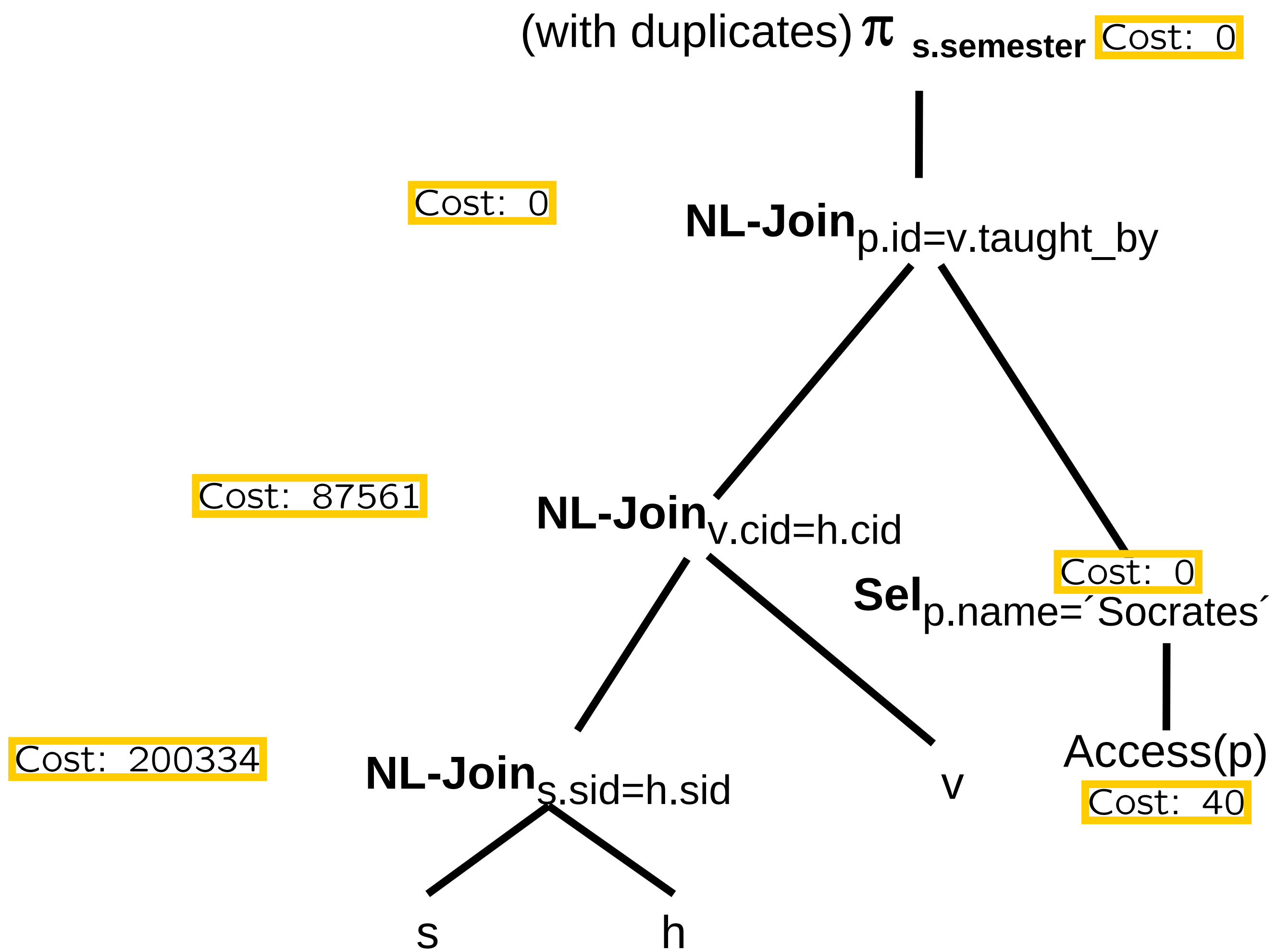
Buffer assignments in this color

Example 1b



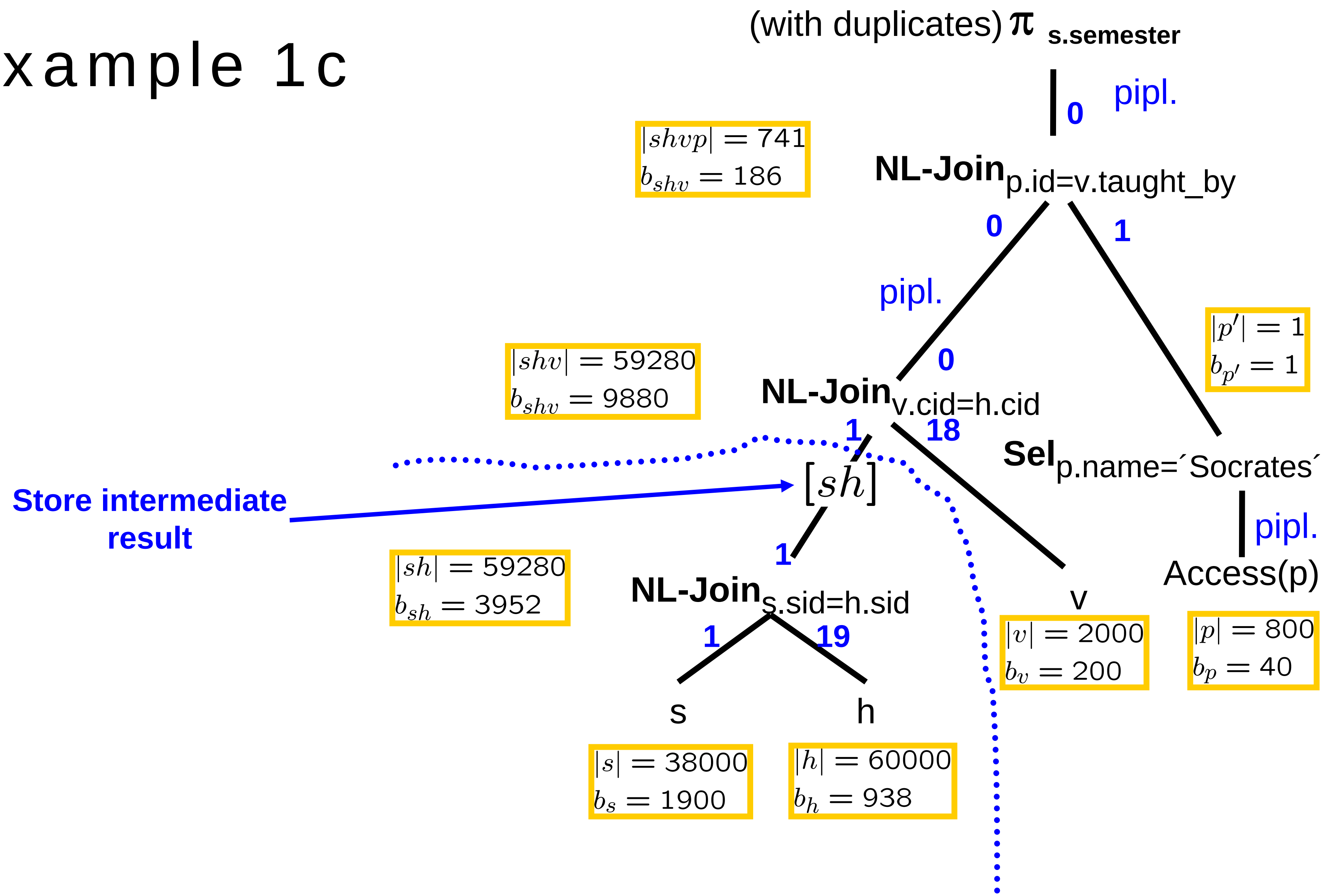
(with duplicates) π s.semester
0
pipl.
I/O(pp): 0 (pipelined evaluation)
$|shvp| = 741$
$b_{shvp} = 186$
NL-Join p.id=v.taught_by
0
1
pipl.
I/O(pp): 0
(outer relation only occupies one page. Thus inner relation only needs to be read once, from pipeline.)
$|p'| = 1$
$b_{p'} = 1$
$|shv| = 59280$
$b_{shv} = 9880$
NL-Join v.cid=h.cid
0
9
1
Sel p.name=´Socrates´
I/O(pp): $1 + \lceil b_{sh}/9 \rceil * (b_v - 1) =$
$1 + \lceil 3952/9 \rceil * 199 = 87561$
pipl.
$|sh| = 59280$
$b_{sh} = 3952$
NL-Join s.sid=h.sid
v
Access(p)
1
9
$|v| = 2000$
$b_v = 200$
$|p| = 800$
$b_p = 40$
s
h
I/O(pp): $b_h + 1 + \lceil b_h/9 \rceil * (b_s - 1) =$
$938 + 1 + \lceil 938/9 \rceil * (1900 - 1) = 200334$
$|s| = 38000$
$b_s = 1900$
$|h| = 60000$
$b_h = 938$
I/O (pp): b_p

Example 1b



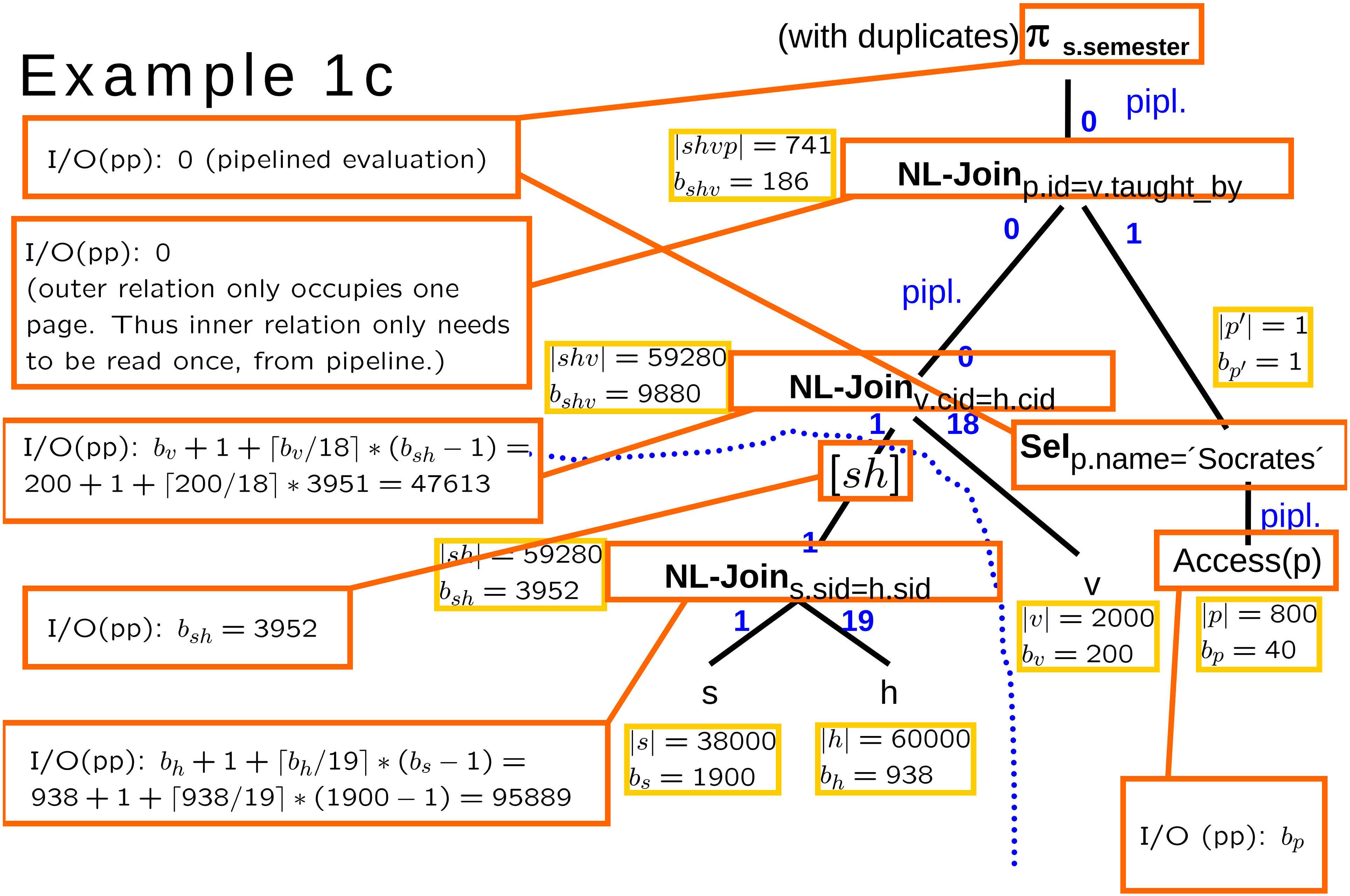
Total cost: 287935

Example 1c



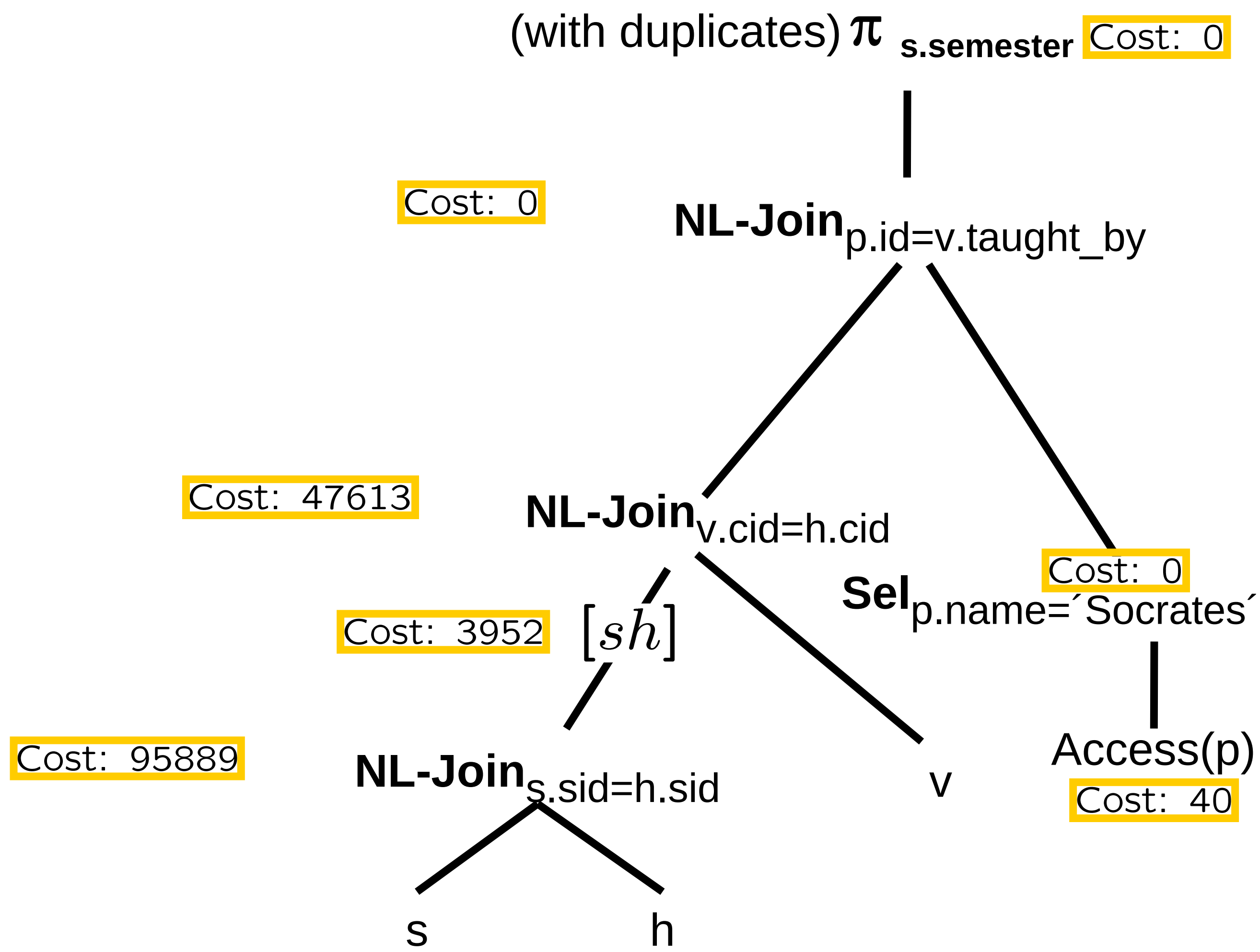
Buffer assignments in this color

Example 1c



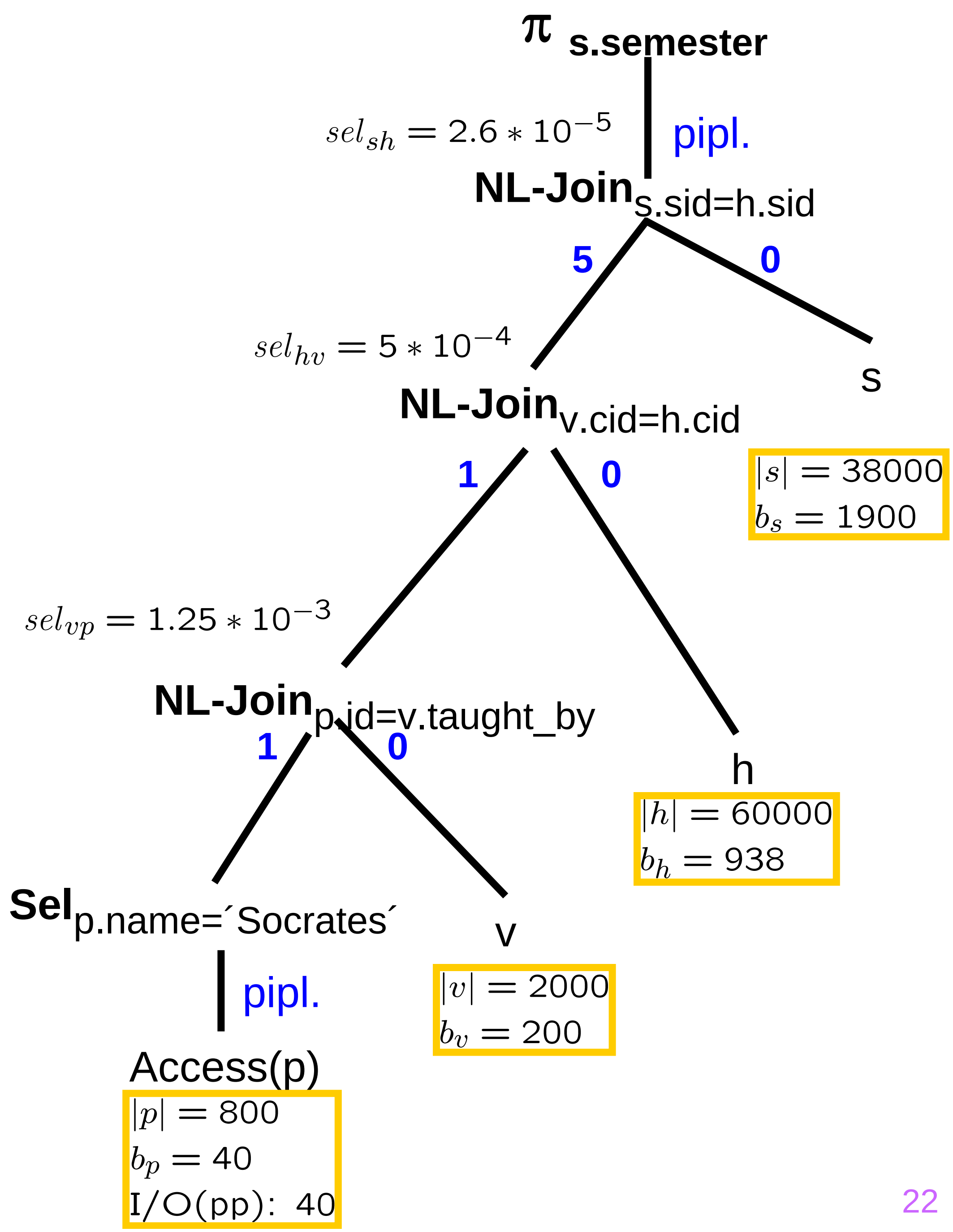
(with duplicates) π s.semester
pipl.
0
I/O(pp): 0 (pipelined evaluation)
|shvp| = 741
b_shv = 186
NL-Join p.id=v.taught_by
0
1
I/O(pp): 0
(outer relation only occupies one page. Thus inner relation only needs to be read once, from pipeline.)
pipl.
|p′| = 1
b_p′ = 1
|shv| = 59280
b_shv = 9880
NL-Join v.cid=h.cid
0
1
18
I/O(pp): b_v + 1 + ⌈b_v/18⌉ ∗ (b_sh − 1) =
200 + 1 + ⌈200/18⌉ ∗ 3951 = 47613
[sh]
Sel p.name=´Socrates´
pipl.
1
|sh| = 59280
b_sh = 3952
NL-Join s.sid=h.sid
Access(p)
v
I/O(pp): b_sh = 3952
1
19
|v| = 2000
b_v = 200
|p| = 800
b_p = 40
s
h
I/O(pp): b_h + 1 + ⌈b_h/19⌉ ∗ (b_s − 1) =
938 + 1 + ⌈938/19⌉ ∗ (1900 − 1) = 95889
|s| = 38000
b_s = 1900
|h| = 60000
b_h = 938
I/O (pp): b_p

Example 1c



Total cost: 147494

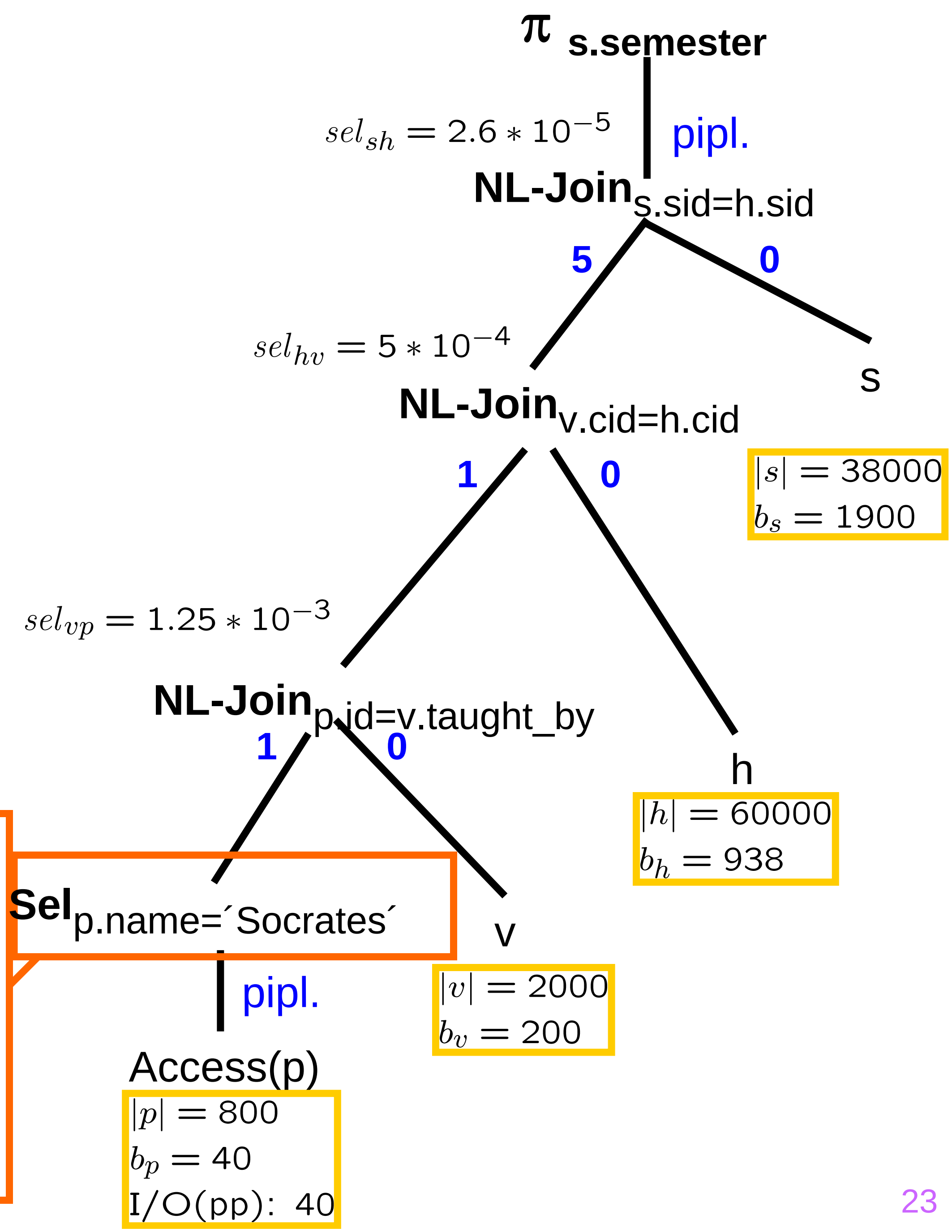
Example 2



π s.semester
$sel_{sh} = 2.6 * 10^{-5}$
pipl.
NL-Join s.sid=h.sid
5
0
$sel_{hv} = 5 * 10^{-4}$
s
NL-Join v.cid=h.cid
1
0
$|s| = 38000$
$b_s = 1900$
$sel_{vp} = 1.25 * 10^{-3}$
NL-Join p.id=v.taught_by
1
0
h
$|h| = 60000$
$b_h = 938$
Sel p.name=´Socrates´
v
pipl.
$|v| = 2000$
$b_v = 200$
Access(p)
$|p| = 800$
$b_p = 40$
I/O(pp): 40

Example 2

(Same query, different Join order)

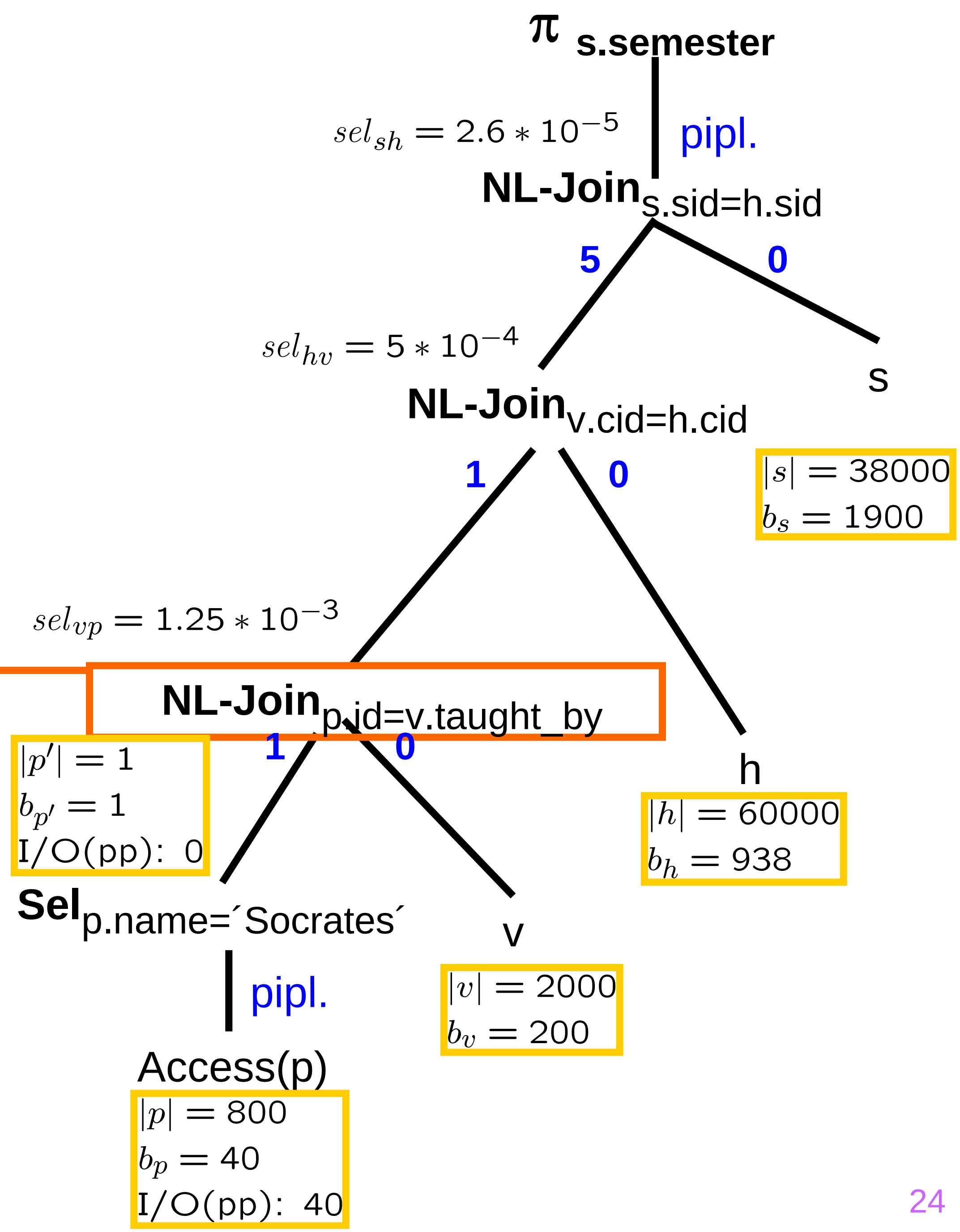


I/O(pp): 0 (pipelined evaluation)

#tuples: $|p'| = |p| * sel_p = 800 * 1.25 * 10^{-3} = 1$

#pages: $b_{p'} = 1$

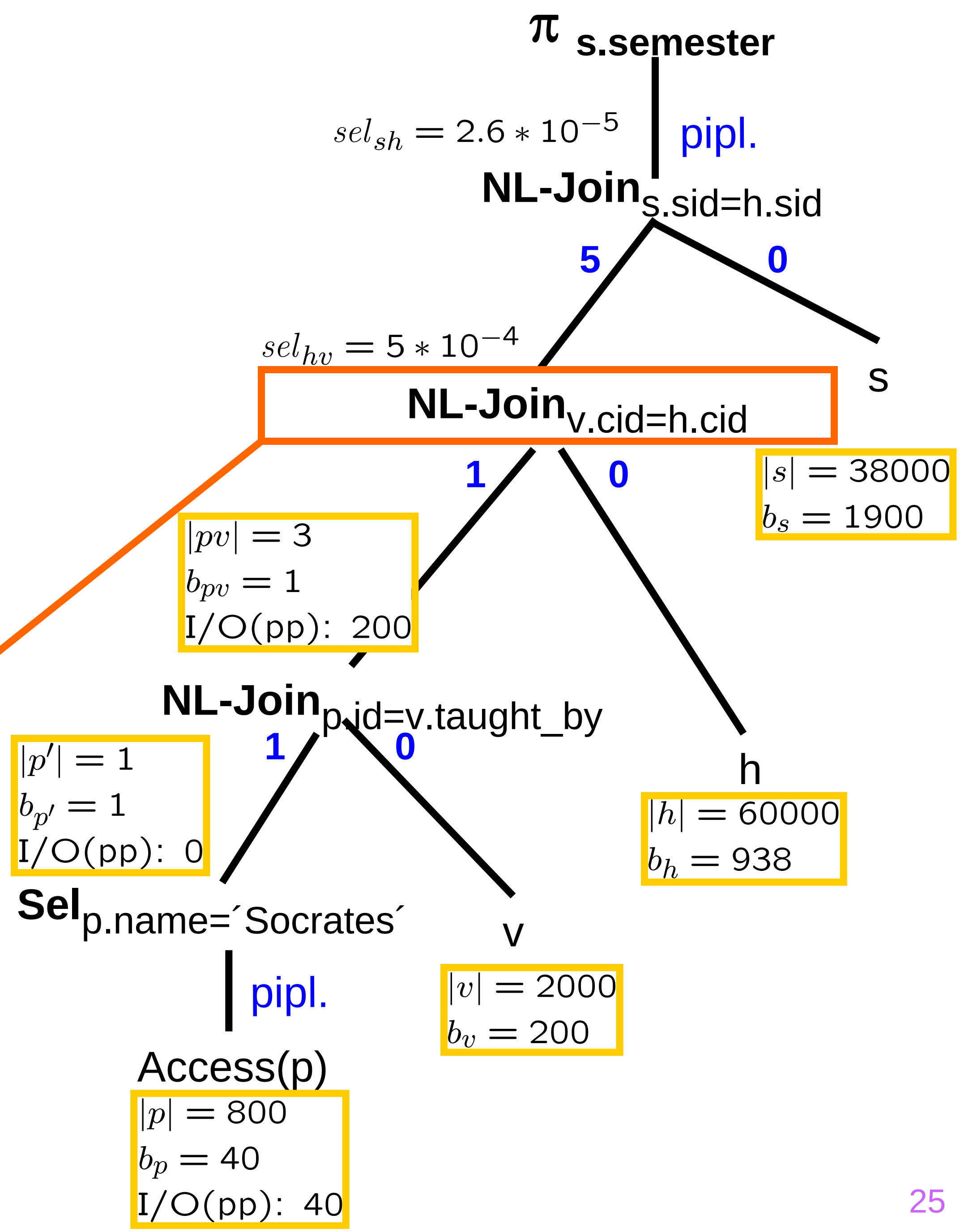
Example 2



I/O(pp): 200

#tuples: $|v| * |p| * sel_{vp} =$
$2000 * 1 * 1.25 * 10^{-3} = 3$
tuple size: 150 Bytes
#tuples/page: $\lfloor 1024/150 \rfloor = 6$
#pages: $\lceil 3/6 \rceil = 1$

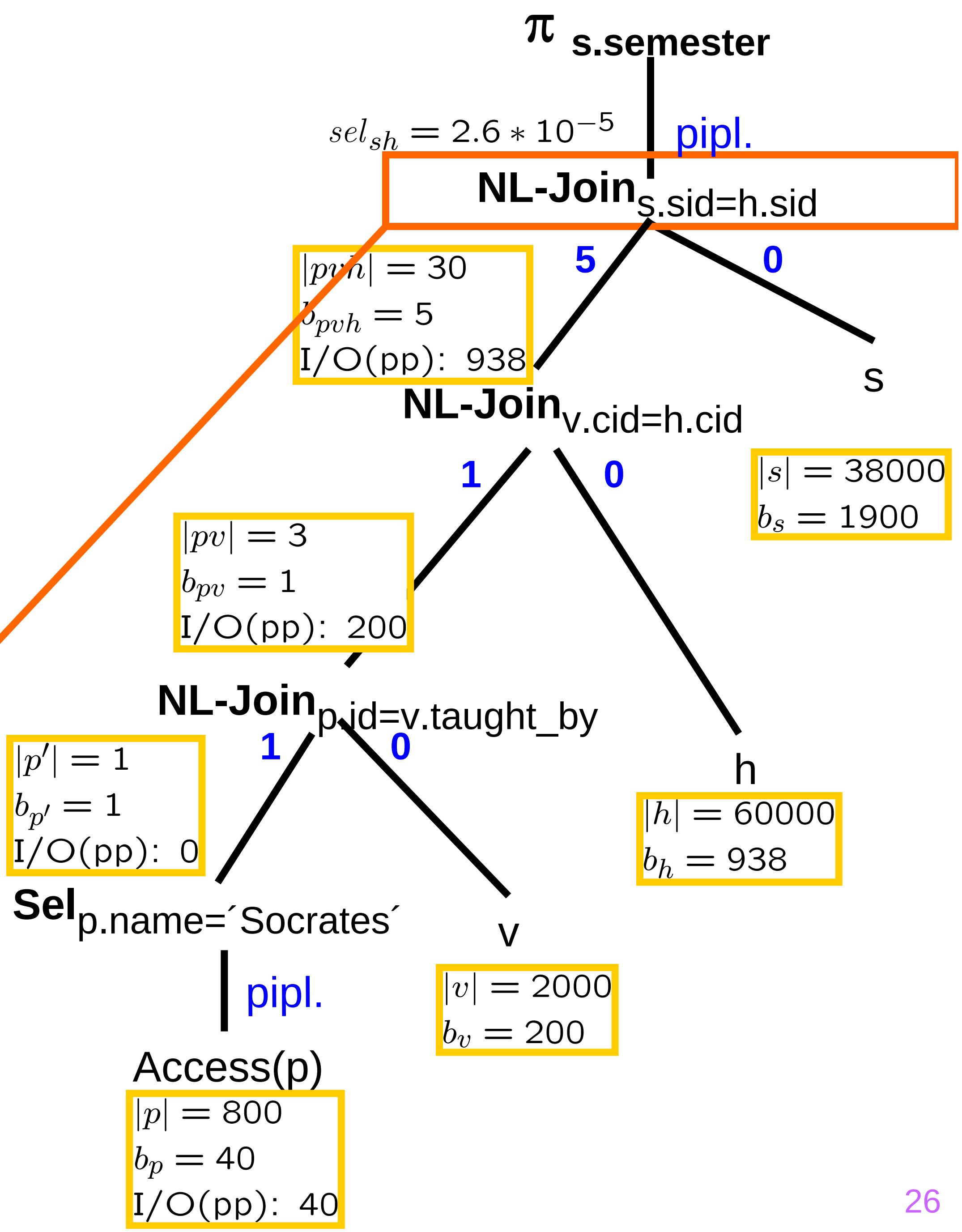
Example 2



I/O(pp): $b_h = 938$

#tuples: $|pv| * |h| * sel_{hv} =$
$1 * 60000 * 5 * 10^{-4} = 30$
tuple size: 166 Bytes
#tuples/page: $\lfloor 1024/166 \rfloor = 6$
#pages: $\lceil 30/6 \rceil = 5$

Example 2



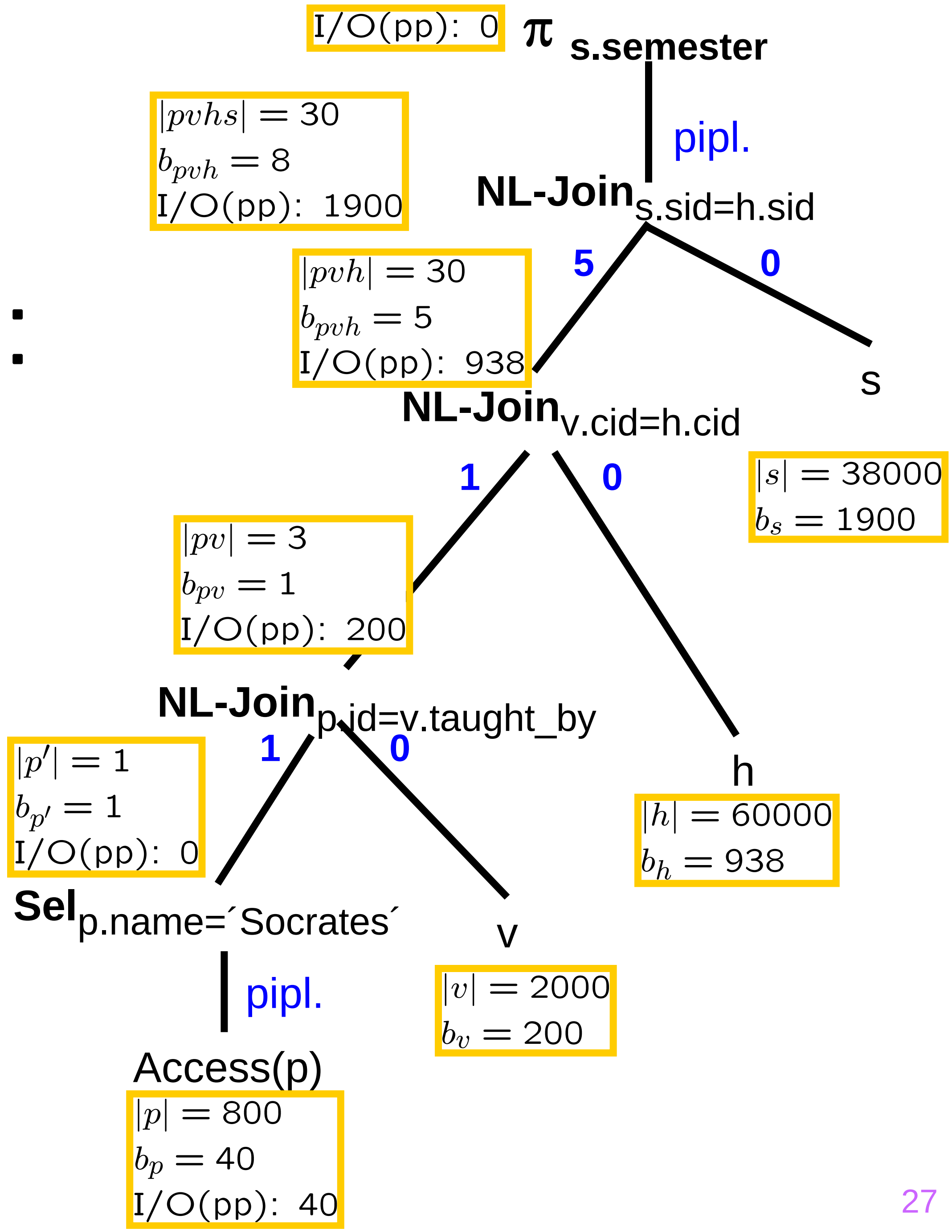
I/O(pp): $b_s = 1900$

#tuples: $|pvh| * |s| * sel_{sh} =$
$30 * 38000 * 2.6 * 10^{-5} = 30$
tuple size: 216 Bytes
#tuples/page: $\lfloor 1024/216 \rfloor = 4$
#pages: $\lceil 30/4 \rceil = 8$

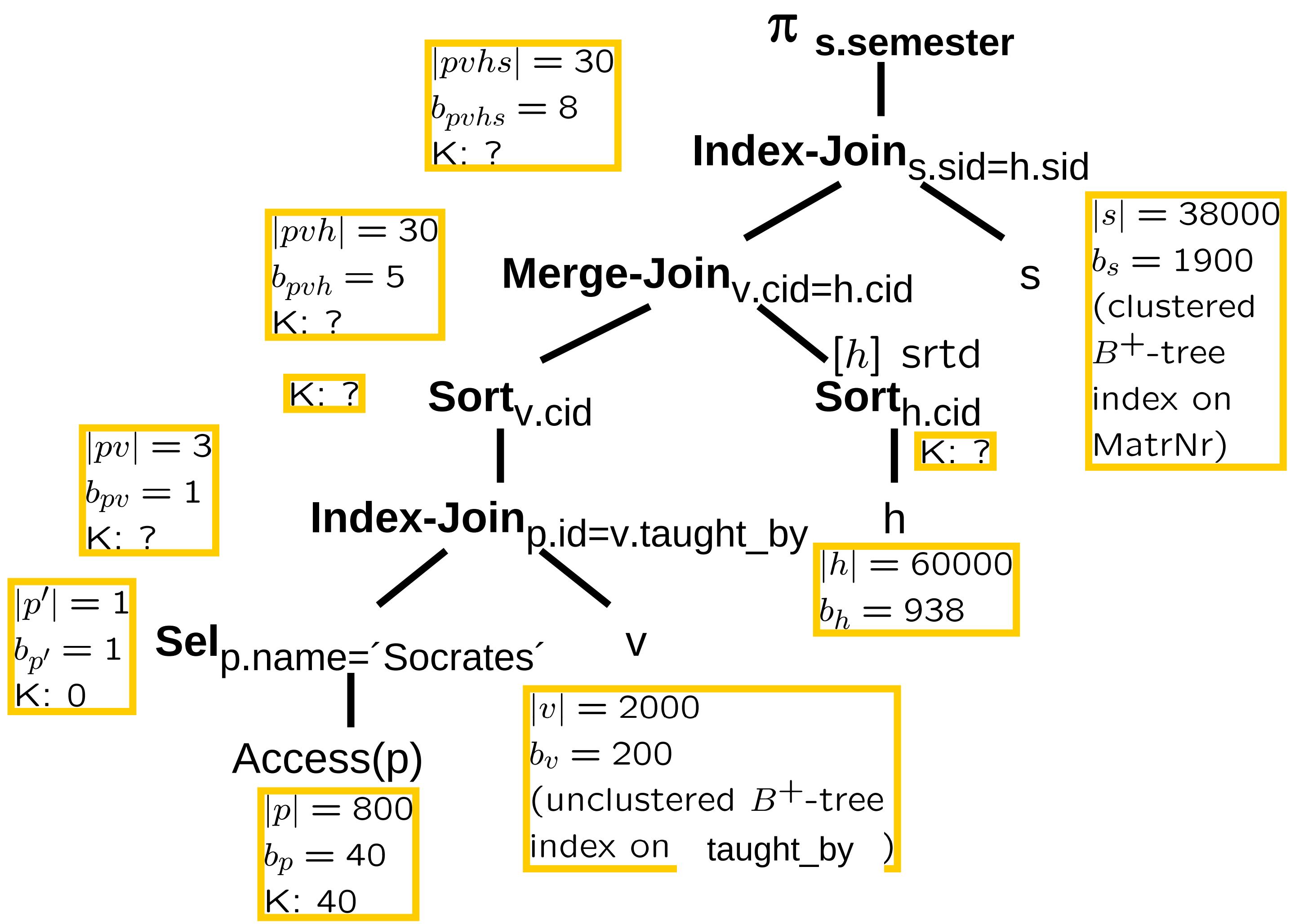
Example 2

Total cost:

3078

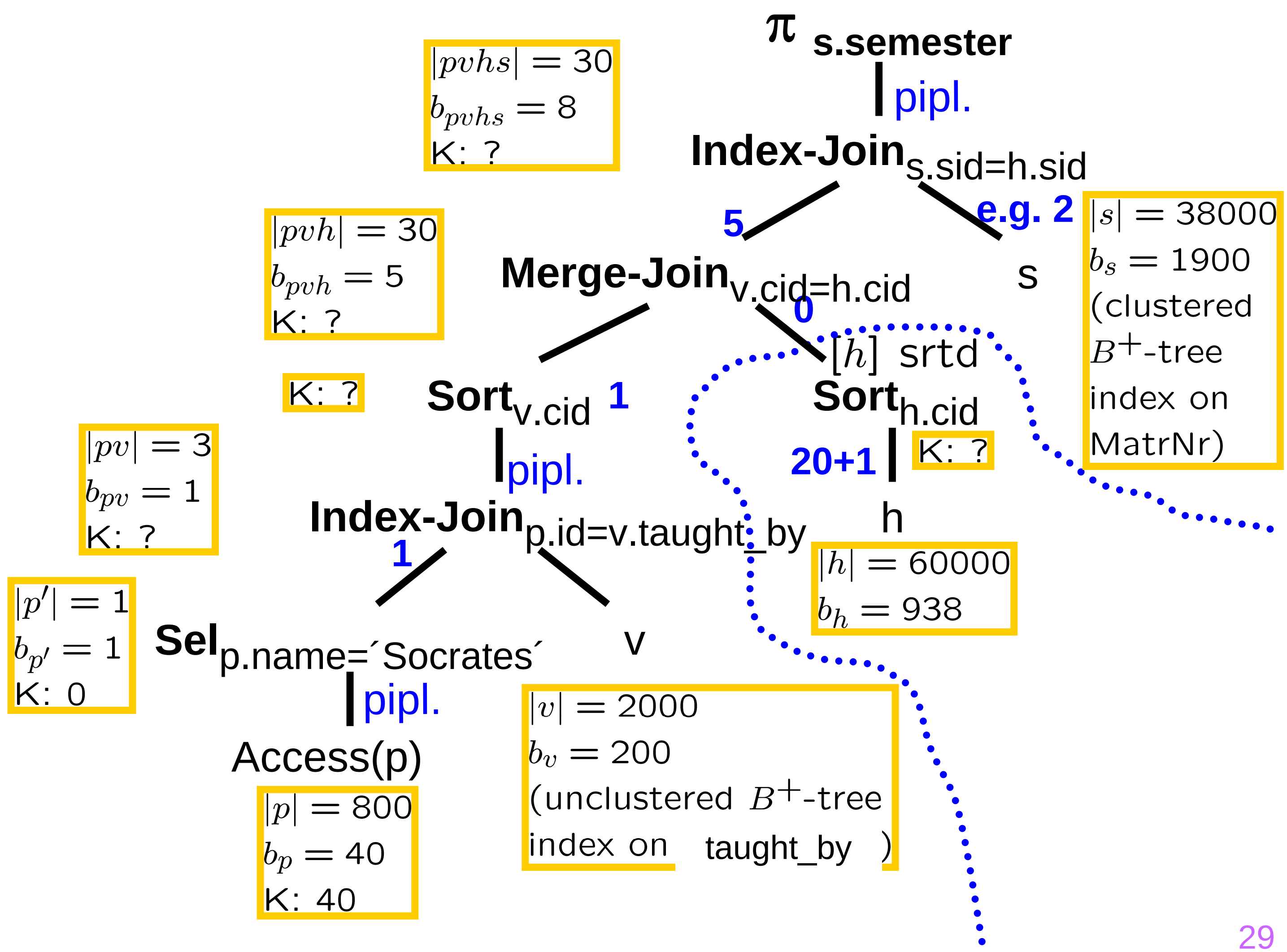


Example 3a



Shortcut K
for I/O(pp)

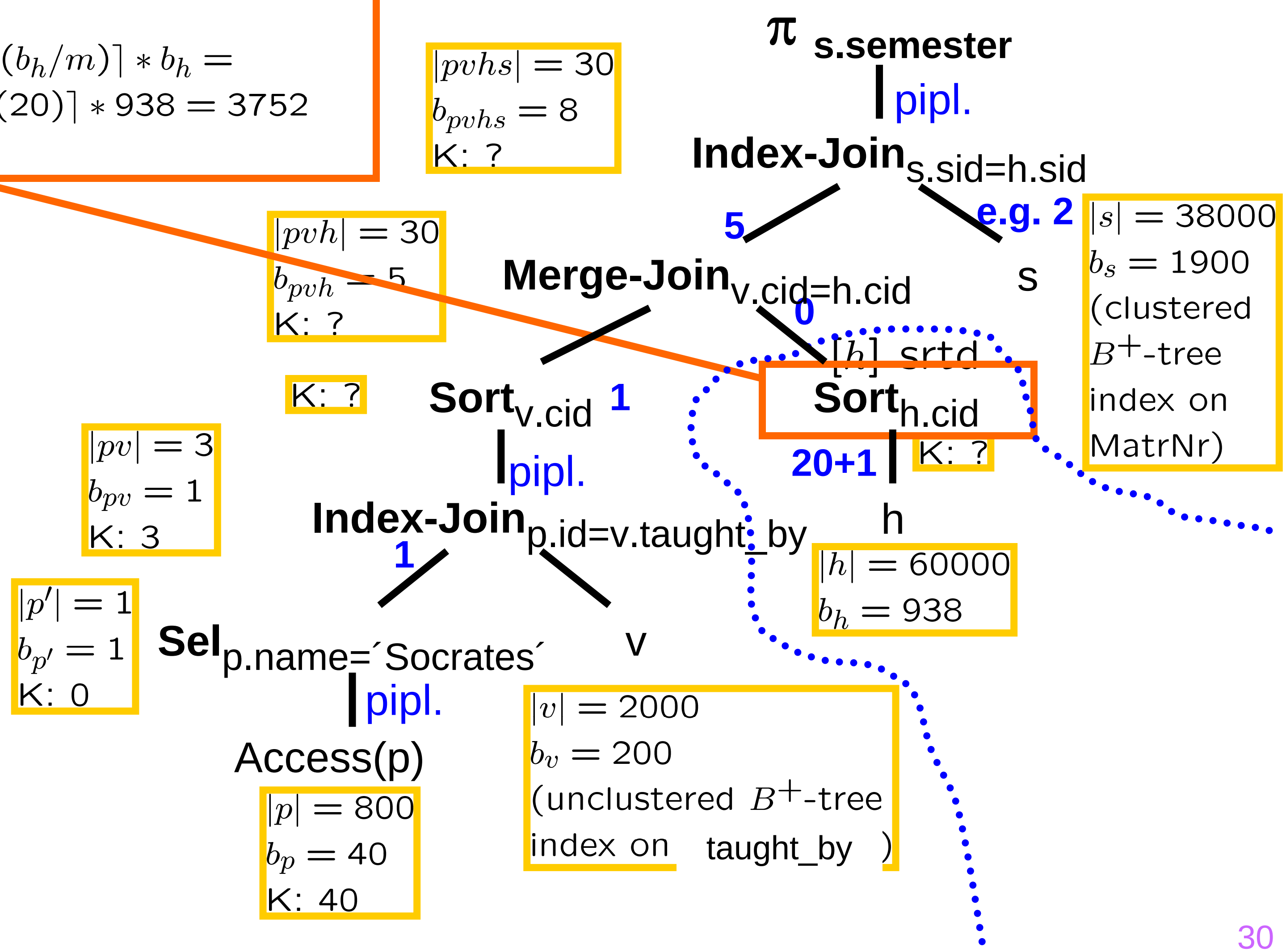
Example 3a



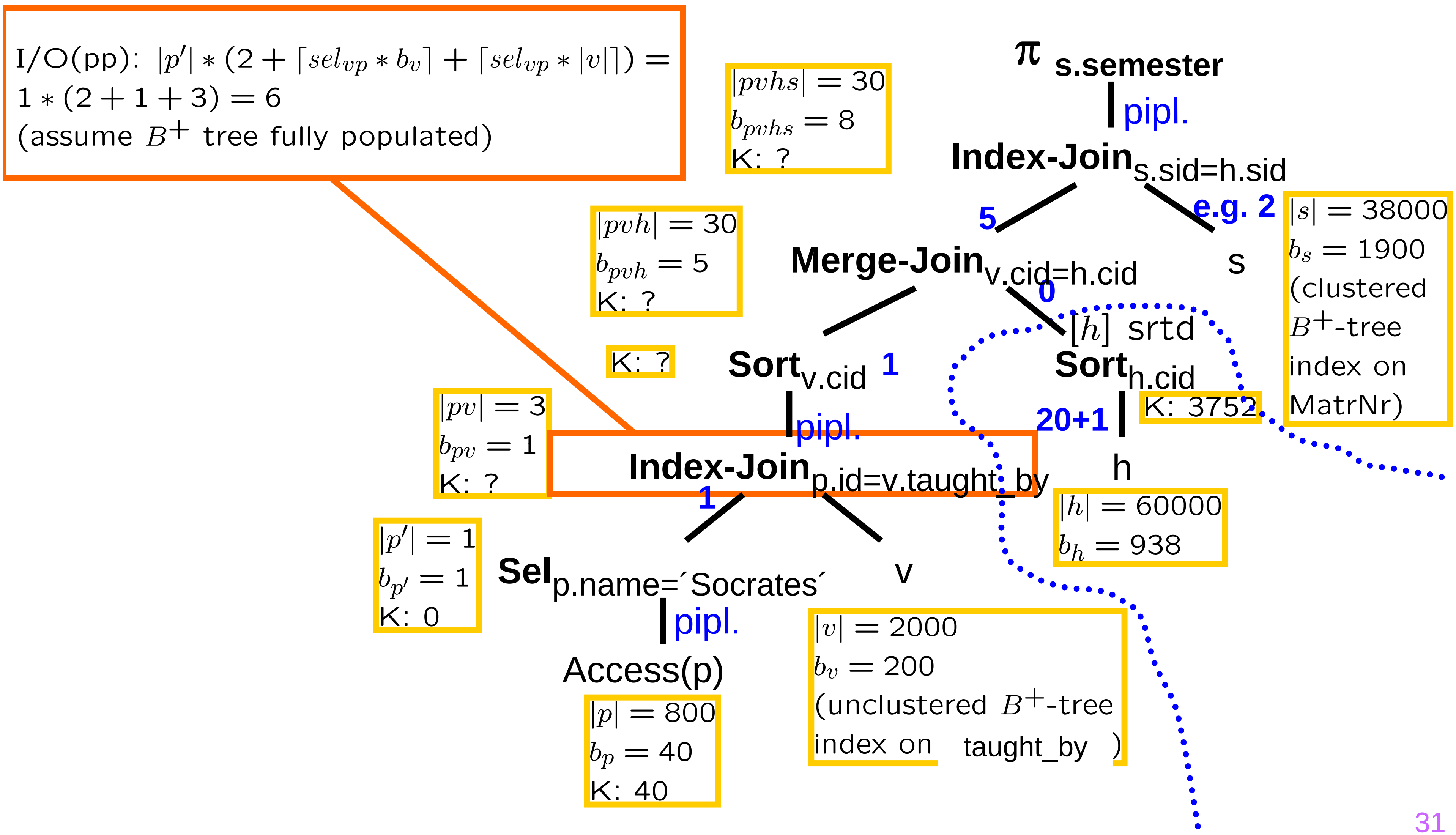
π s.semester
pipl.
Index-Join s.sid=h.sid
5
e.g. 2
Merge-Join v.cid=h.cid
0
s
[h] srtd
Sort v.cid
1
Sort h.cid
20+1
pipl.
Index-Join p.id=v.taught_by
1
h
Sel p.name=´Socrates´
v
pipl.
Access(p)
|pvhs| = 30
b_pvhs = 8
K: ?
|pvh| = 30
b_pvh = 5
K: ?
K: ?
|pv| = 3
b_pv = 1
K: ?
|p'| = 1
b_p' = 1
K: 0
|p| = 800
b_p = 40
K: 40
|v| = 2000
b_v = 200
(unclustered B+-tree
index on taught_by)
|h| = 60000
b_h = 938
K: ?
|s| = 38000
b_s = 1900
(clustered
B+-tree
index on
MatrNr)

Example 3a

I/O(pp): $2 * \lceil \log_{m-1}(b_h/m) \rceil * b_h =$
$2 * \lceil \log(938/21)/\log(20) \rceil * 938 = 3752$

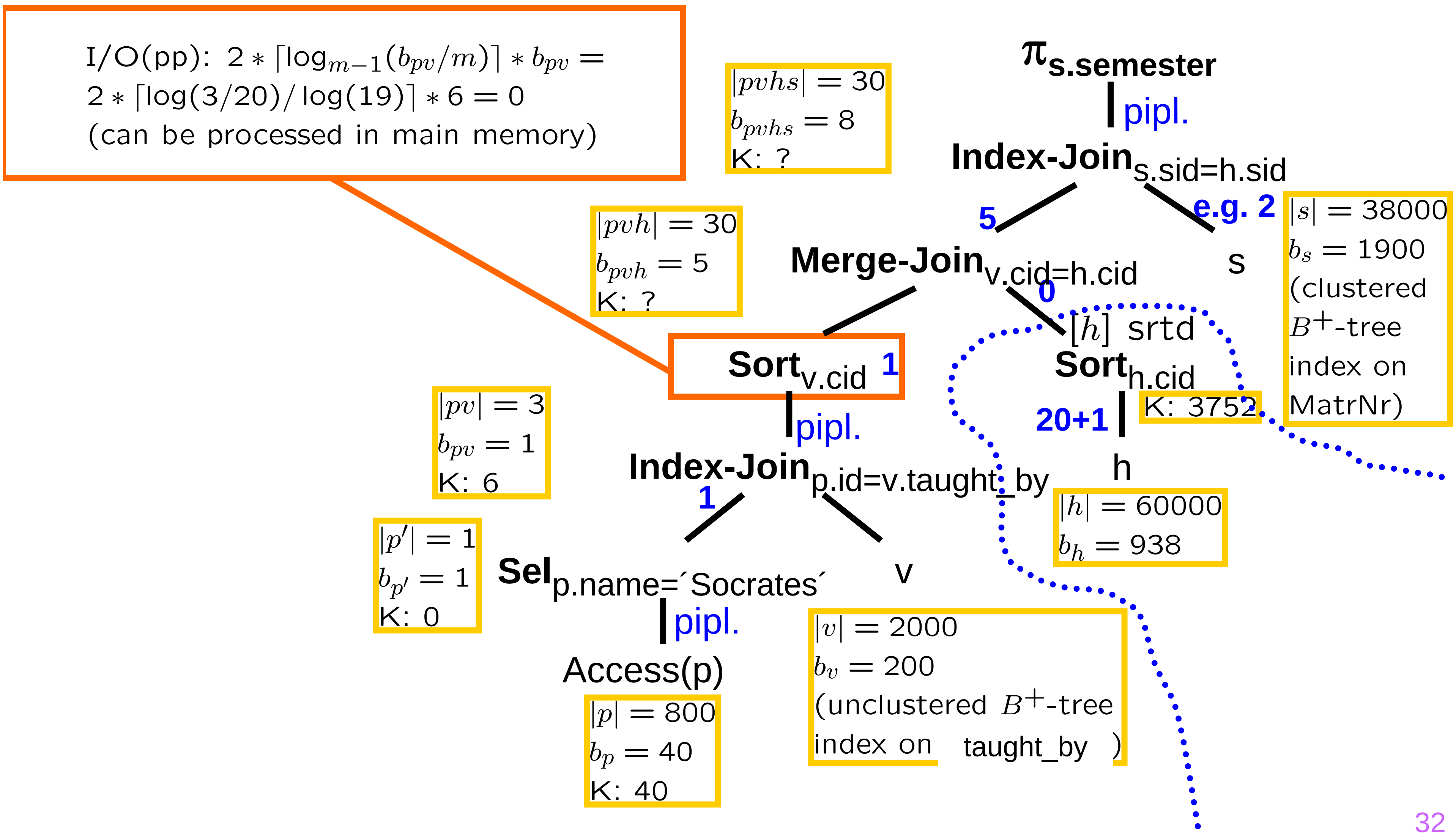


Example 3a



I/O(pp): $|p'| * (2 + \lceil sel_{vp} * b_v \rceil + \lceil sel_{vp} * |v| \rceil) =$
$1 * (2 + 1 + 3) = 6$
(assume B^+ tree fully populated)
π s.semester
pipl.
Index-Join s.sid=h.sid
5
e.g. 2
$|pvhs| = 30$
$b_{pvhs} = 8$
K: ?
$|pvh| = 30$
$b_{pvh} = 5$
K: ?
Merge-Join v.cid=h.cid
0
s
$|s| = 38000$
$b_s = 1900$
(clustered B^+-tree index on MatrNr)
K: ?
Sort v.cid
1
$[h]$ srtd
Sort h.cid
K: 3752
20+1
$|pv| = 3$
$b_{pv} = 1$
K: ?
pipl.
Index-Join p.id=v.taught_by
h
1
$|h| = 60000$
$b_h = 938$
$|p'| = 1$
$b_{p'} = 1$
K: 0
Sel p.name=´Socrates´
v
pipl.
Access(p)
$|p| = 800$
$b_p = 40$
K: 40
$|v| = 2000$
$b_v = 200$
(unclustered B^+-tree index on taught_by)

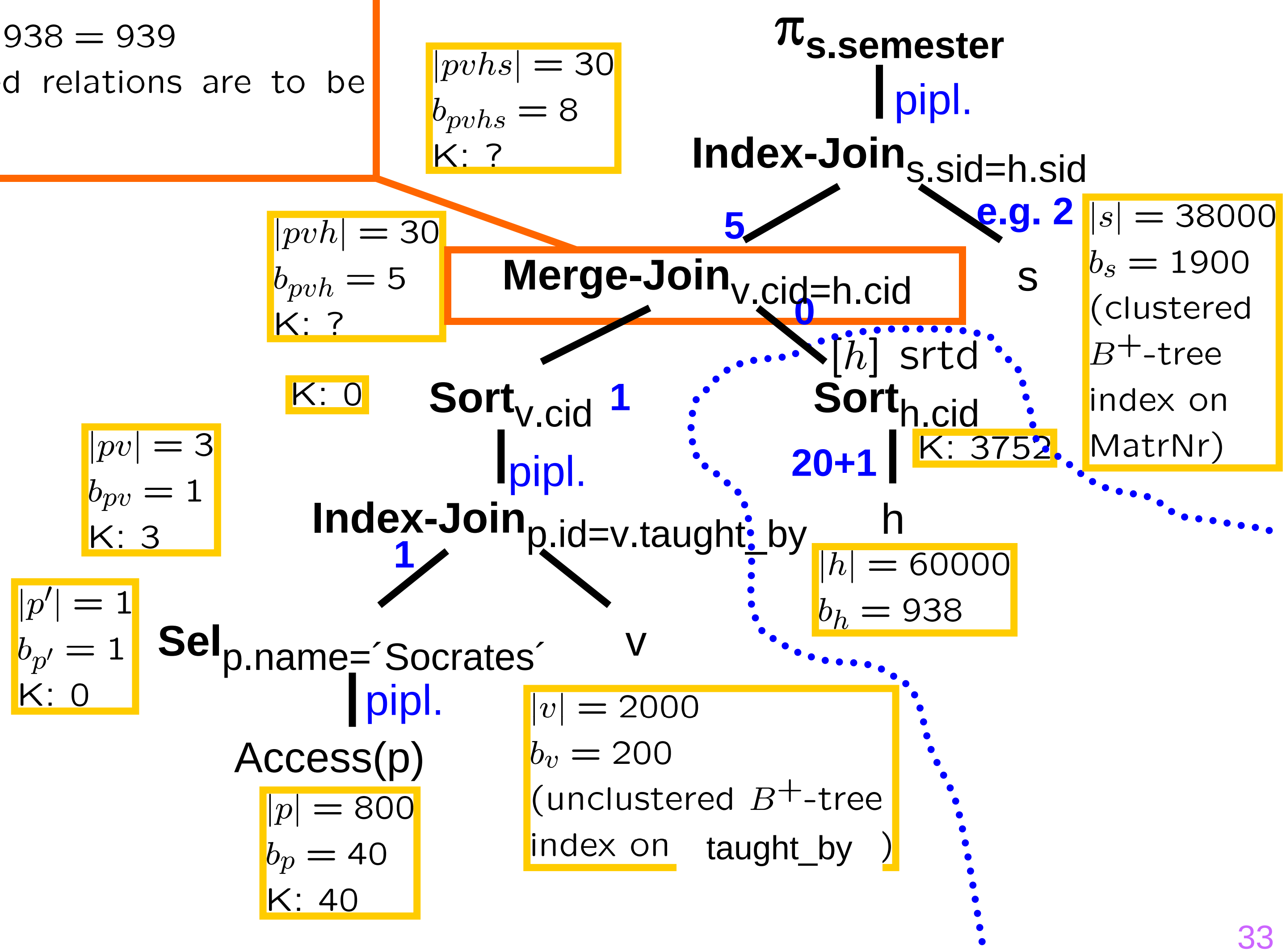
Example 3a



I/O(pp): $2 * \lceil \log_{m-1}(b_{pv}/m) \rceil * b_{pv} =$
$2 * \lceil \log(3/20)/\log(19) \rceil * 6 = 0$
(can be processed in main memory)
$|pvhs| = 30$
$b_{pvhs} = 8$
K: ?
$\pi_{s.semester}$
pipl.
Index-Join$_{s.sid=h.sid}$
5
e.g. 2
$|s| = 38000$
$b_s = 1900$
(clustered B^+-tree index on MatrNr)
s
$|pvh| = 30$
$b_{pvh} = 5$
K: ?
Merge-Join$_{v.cid=h.cid}$
0
$[h]$ srtd
Sort$_{v.cid}$
1
Sort$_{h.cid}$
K: 3752
20+1
h
$|pv| = 3$
$b_{pv} = 1$
K: 6
pipl.
Index-Join$_{p.id=v.taught_by}$
1
$|h| = 60000$
$b_h = 938$
$|p'| = 1$
$b_{p'} = 1$
K: 0
Sel$_{p.name=´Socrates´}$
v
pipl.
Access(p)
$|v| = 2000$
$b_v = 200$
(unclustered B^+-tree index on taught_by)
$|p| = 800$
$b_p = 40$
K: 40

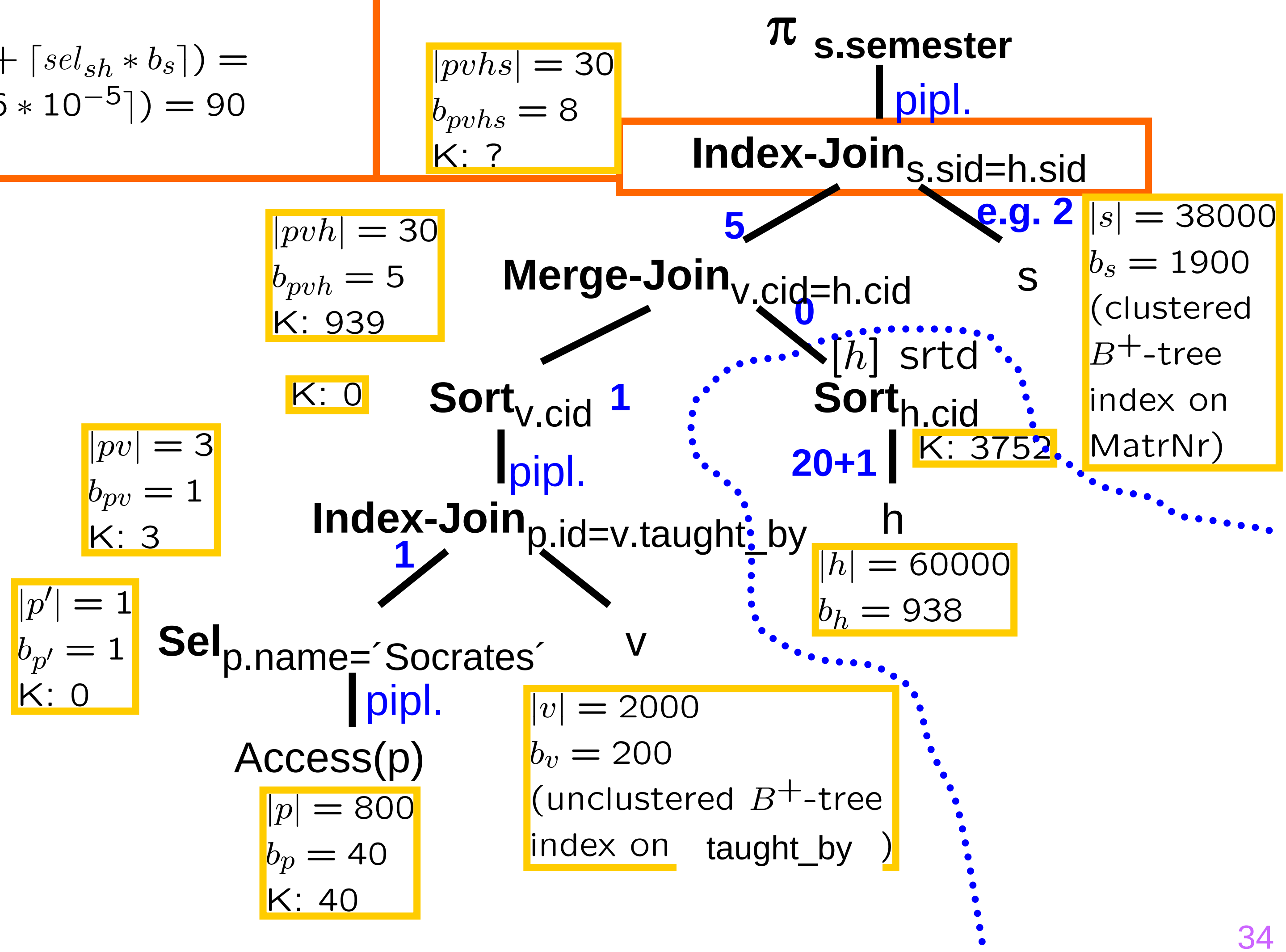
Example 3a

I/O(pp): $b_{pv} + b_h = 1 + 938 = 939$
(After sorting the sorted relations are to be read from disk.)



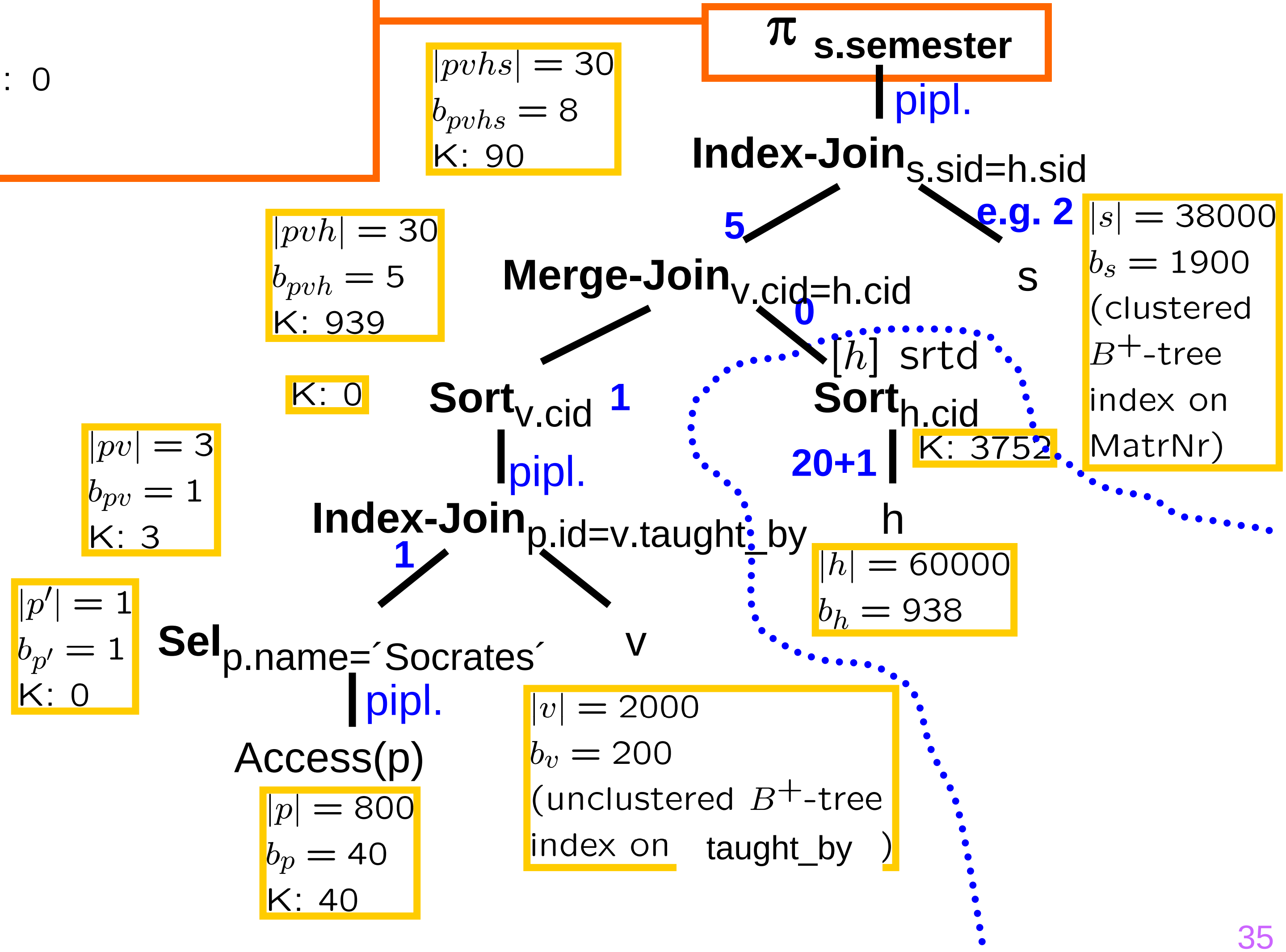
Example 3a

I/O(pp): $|pvh| * (2 + \lceil sel_{sh} * b_s \rceil) =$
$30 * (2 + \lceil 1900 * 2.6 * 10^{-5} \rceil) = 90$



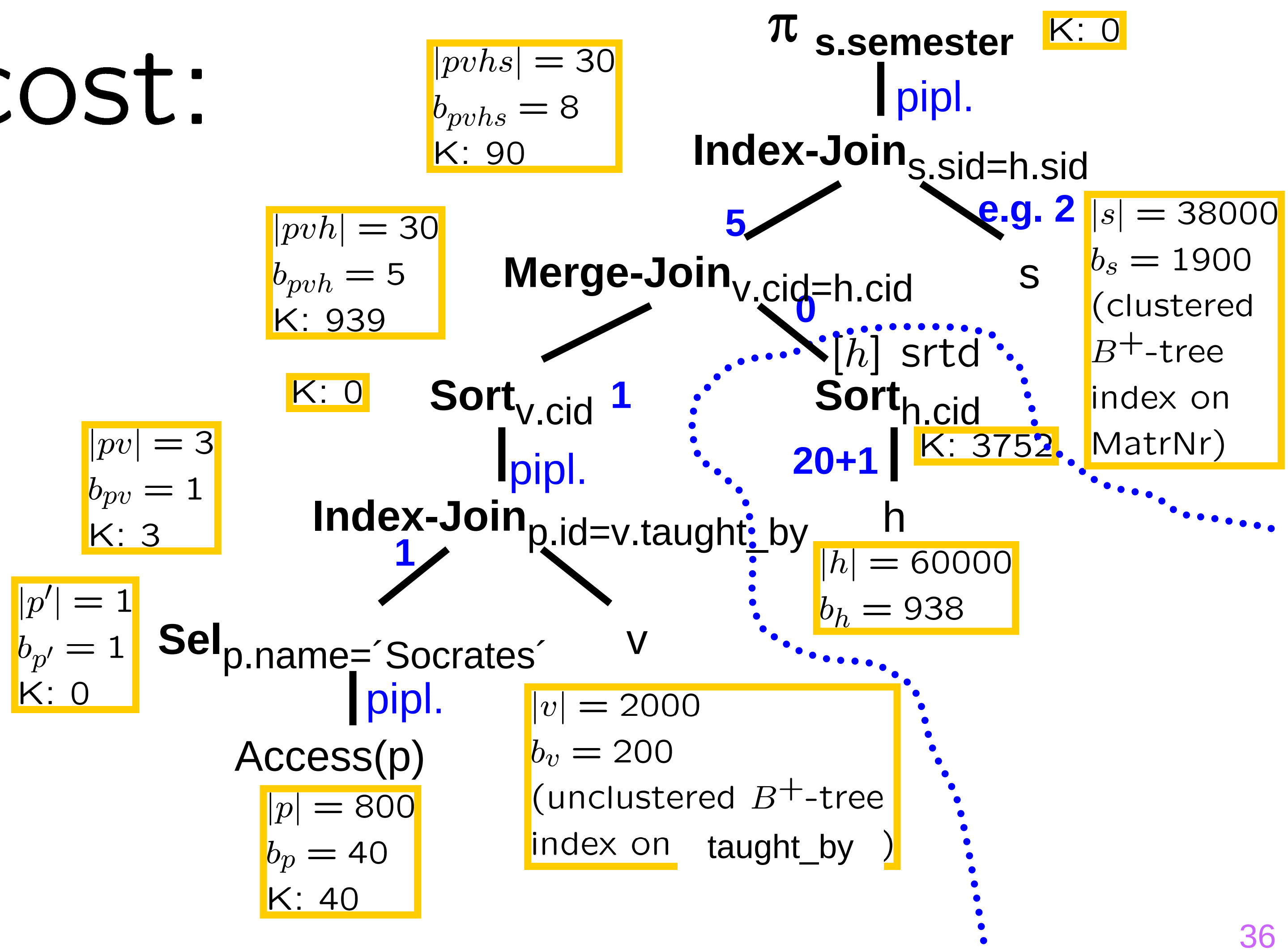
Example 3a

I/O(pp): 0

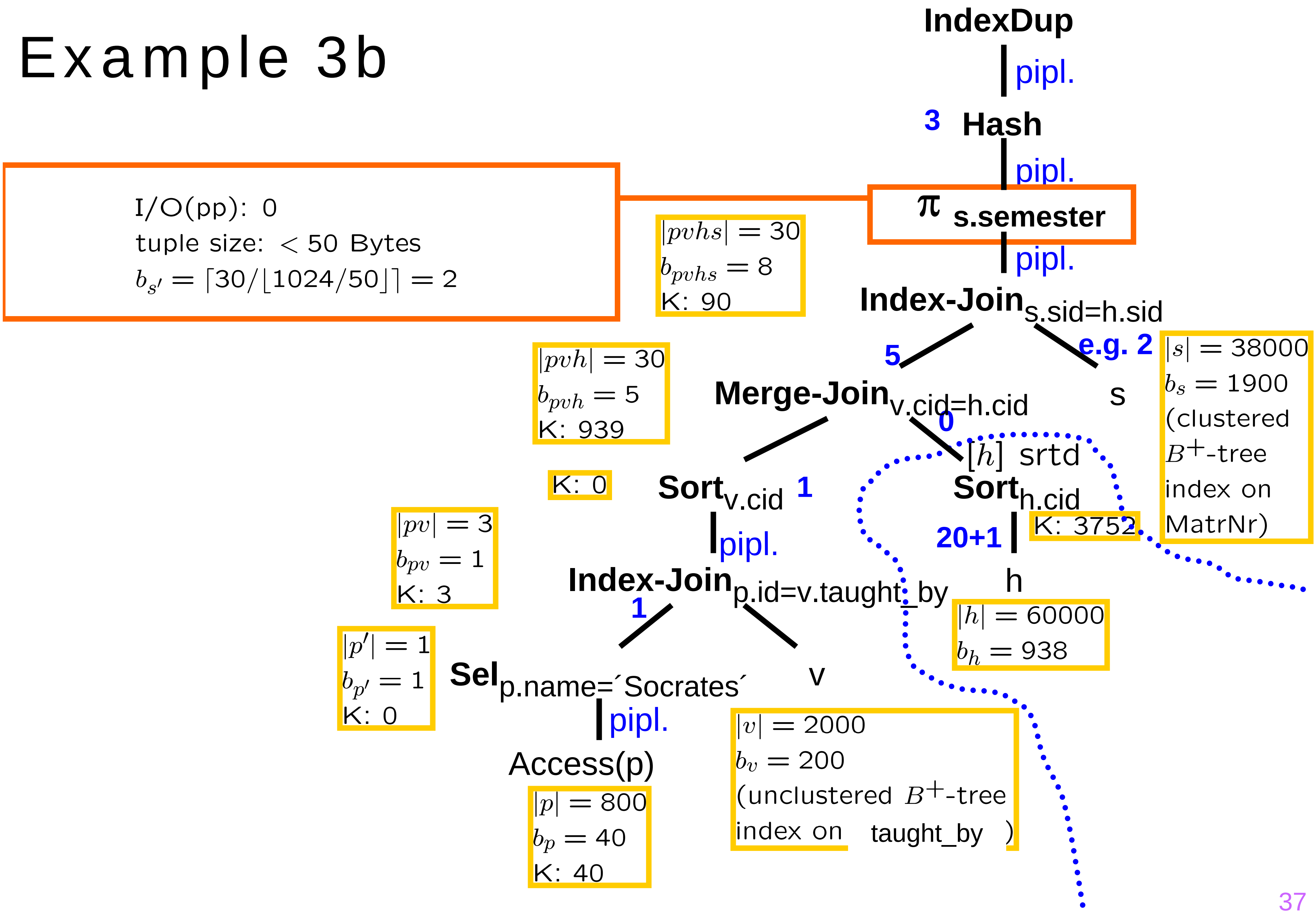


Example 3a

Total cost: 4824



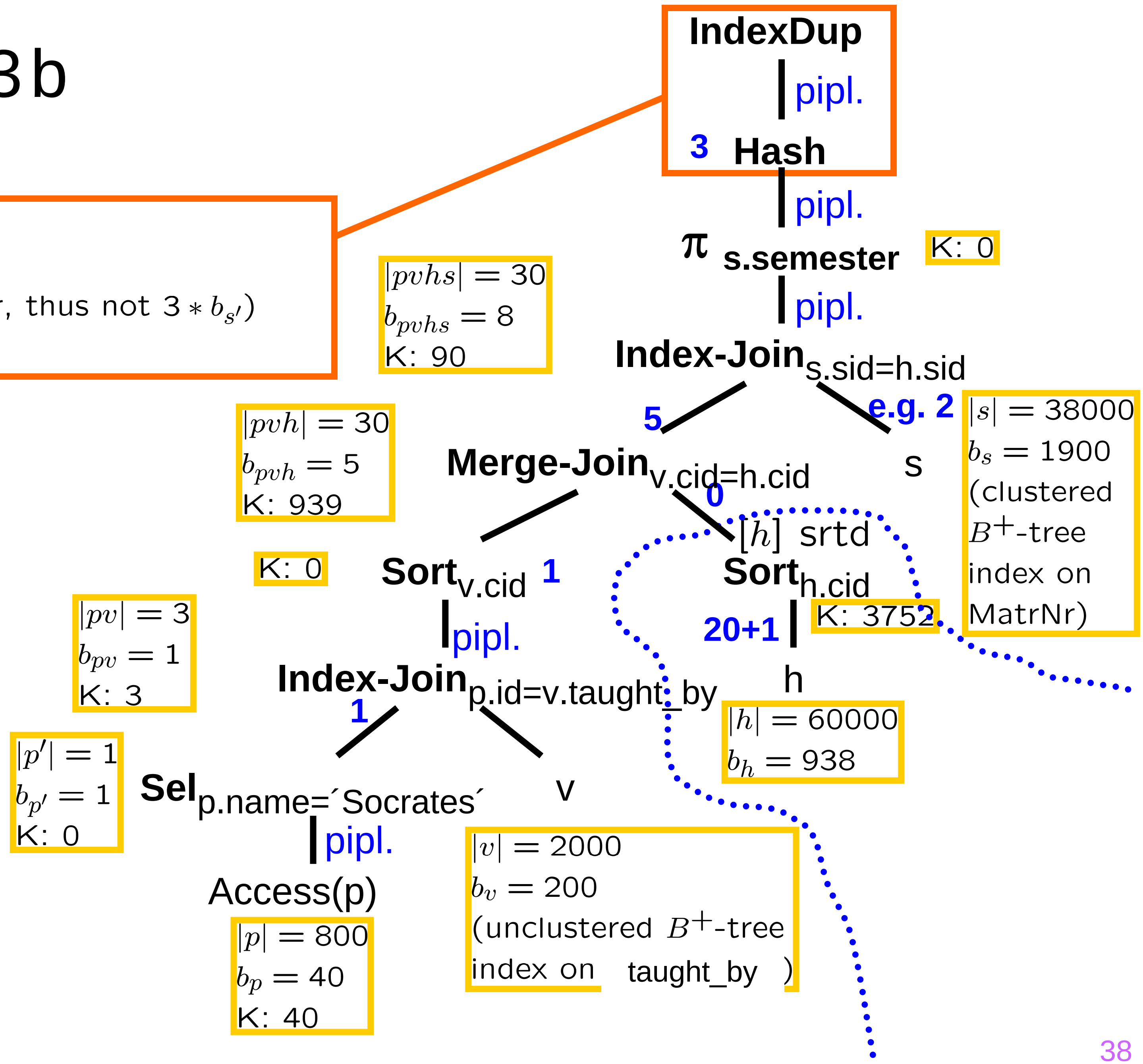
Example 3b



I/O(pp): 0
tuple size: < 50 Bytes
$b_{s'} = \lceil 30/\lfloor 1024/50 \rfloor \rceil = 2$
IndexDup
pipl.
3 Hash
pipl.
π s.semester
pipl.
Index-Join s.sid=h.sid
5
e.g. 2
s
$|s| = 38000$
$b_s = 1900$
(clustered B^+-tree index on MatrNr)
$|pvhs| = 30$
$b_{pvhs} = 8$
K: 90
$|pvh| = 30$
$b_{pvh} = 5$
K: 939
Merge-Join v.cid=h.cid
0
[h] srtd
K: 0
Sort v.cid
1
pipl.
Sort h.cid
20+1
K: 3752
h
$|h| = 60000$
$b_h = 938$
$|pv| = 3$
$b_{pv} = 1$
K: 3
Index-Join p.id=v.taught_by
1
v
$|v| = 2000$
$b_v = 200$
(unclustered B^+-tree index on taught_by)
$|p'| = 1$
$b_{p'} = 1$
K: 0
Sel p.name=´Socrates´
pipl.
Access(p)
$|p| = 800$
$b_p = 40$
K: 40

Example 3b

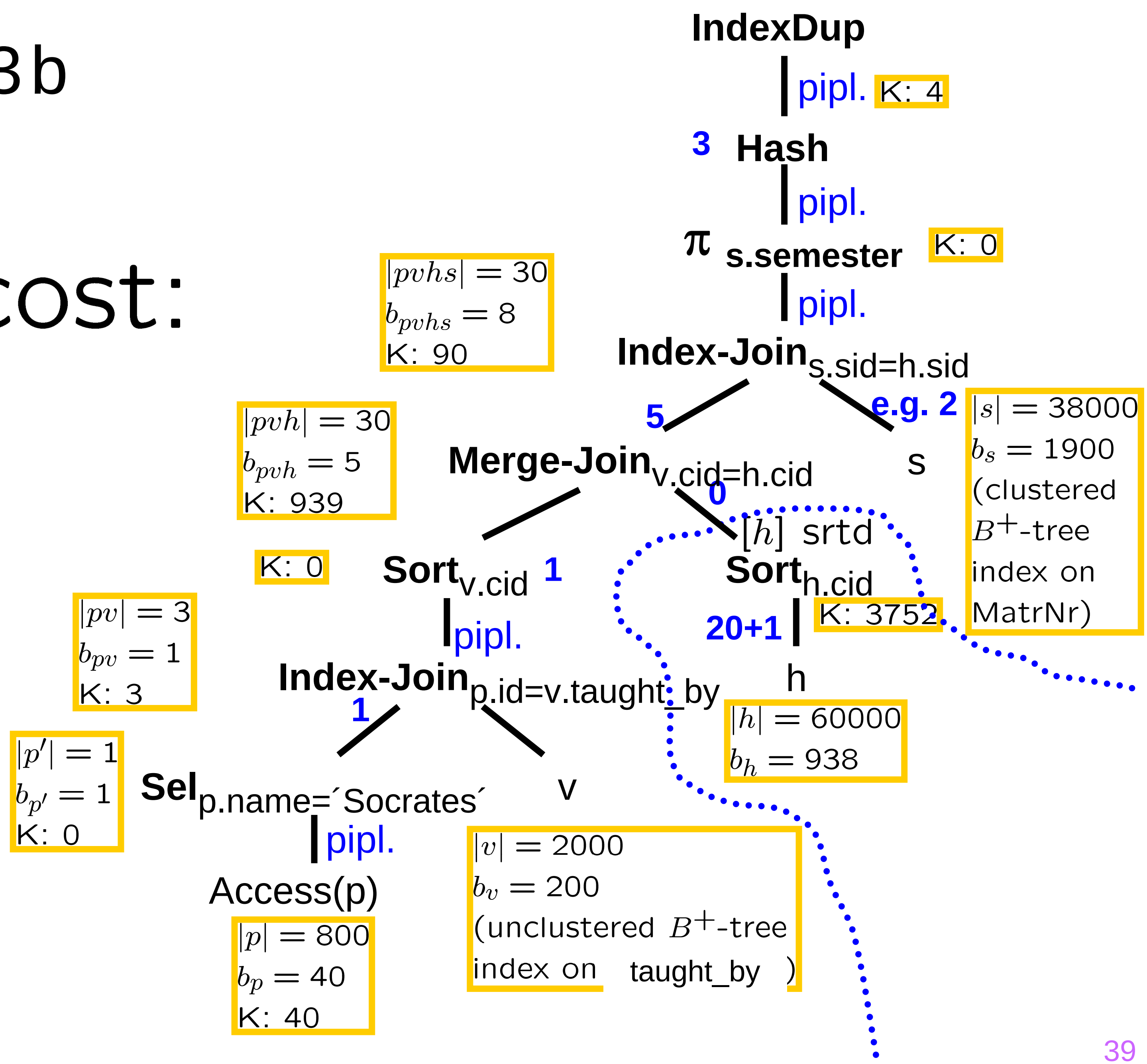
I/O(pp): $2 * b_{s'} = 4$
(reads from iterator, thus not $3 * b_{s'}$)



Example 3b

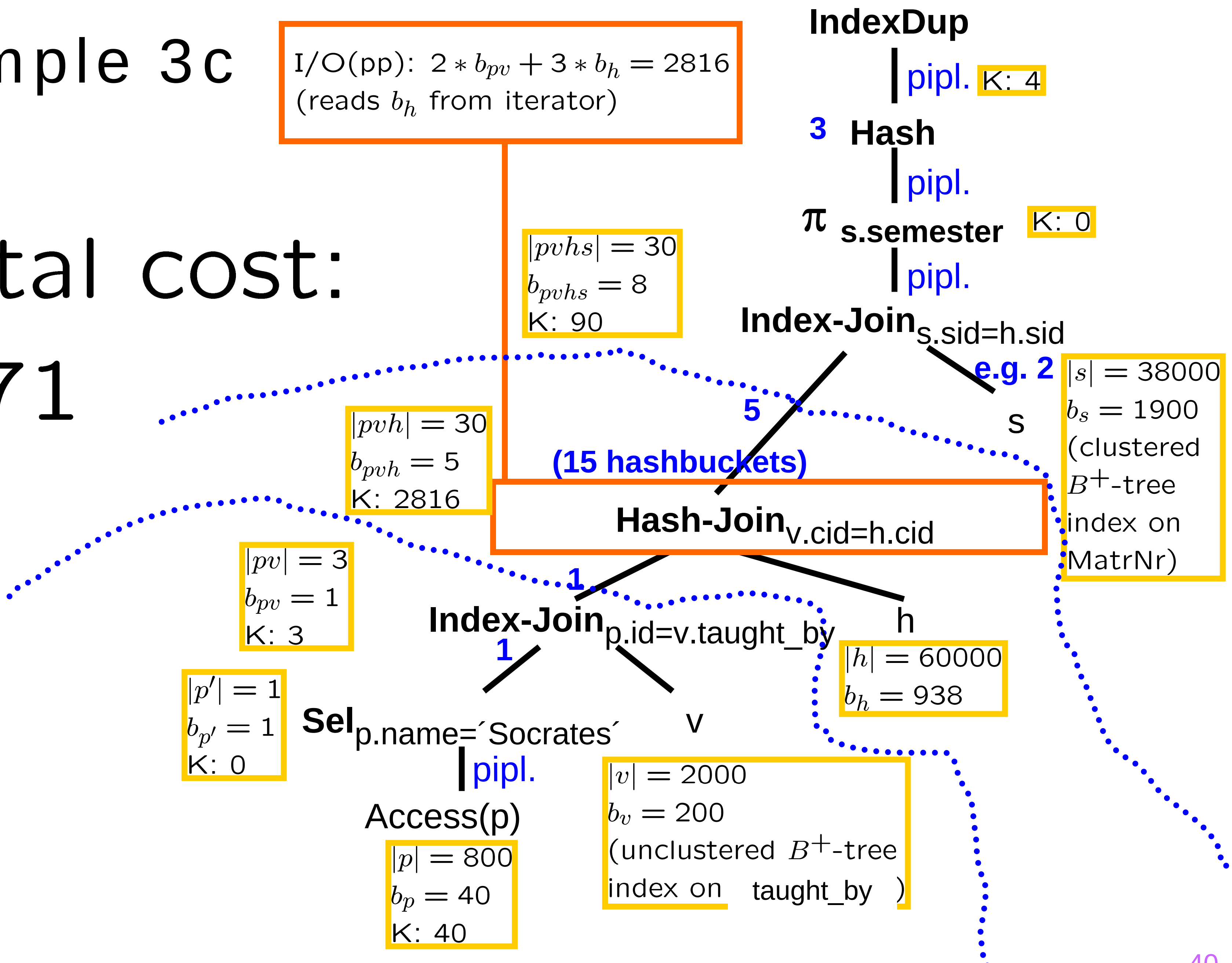
Total cost:

4898



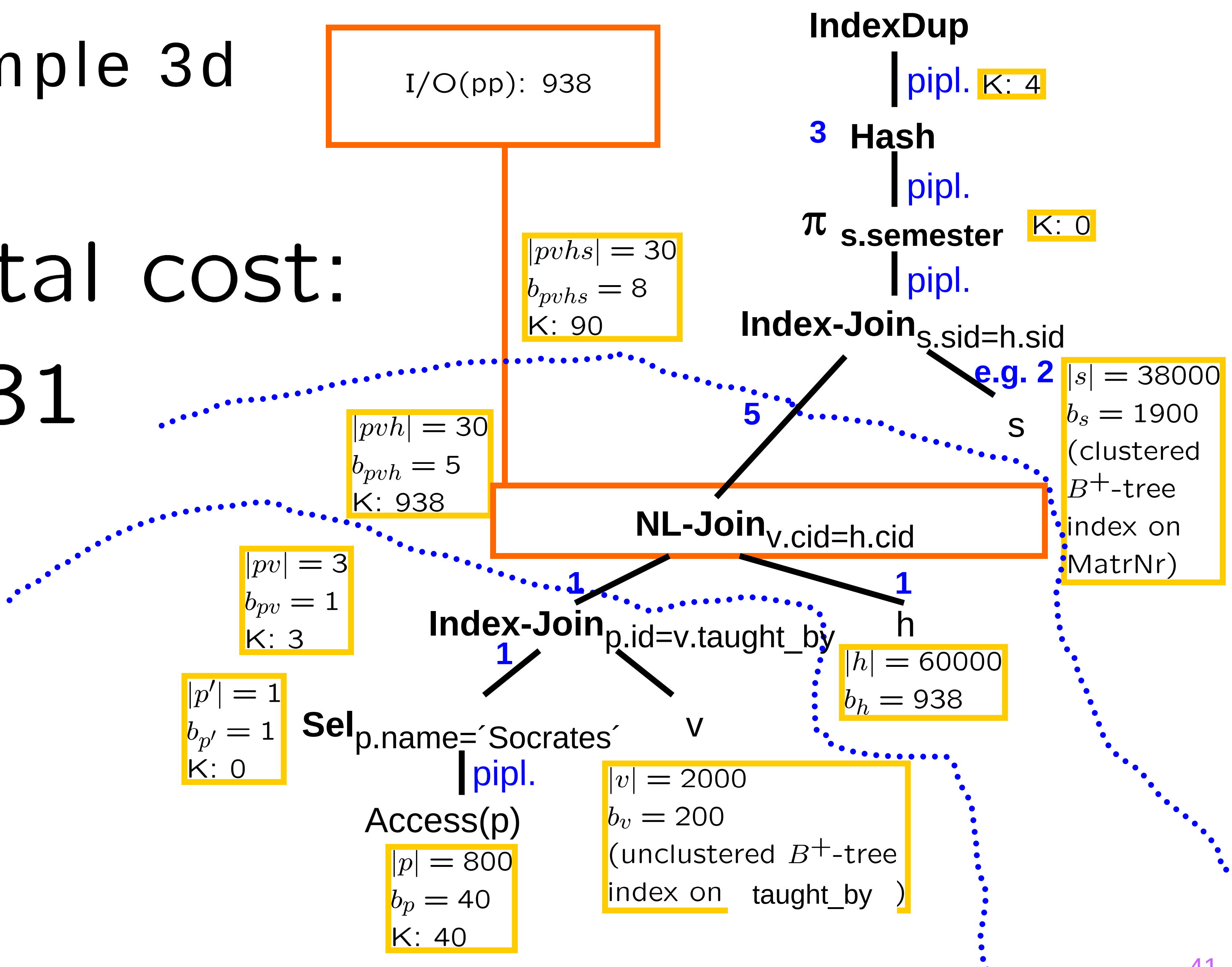
Example 3c

Total cost: 2071



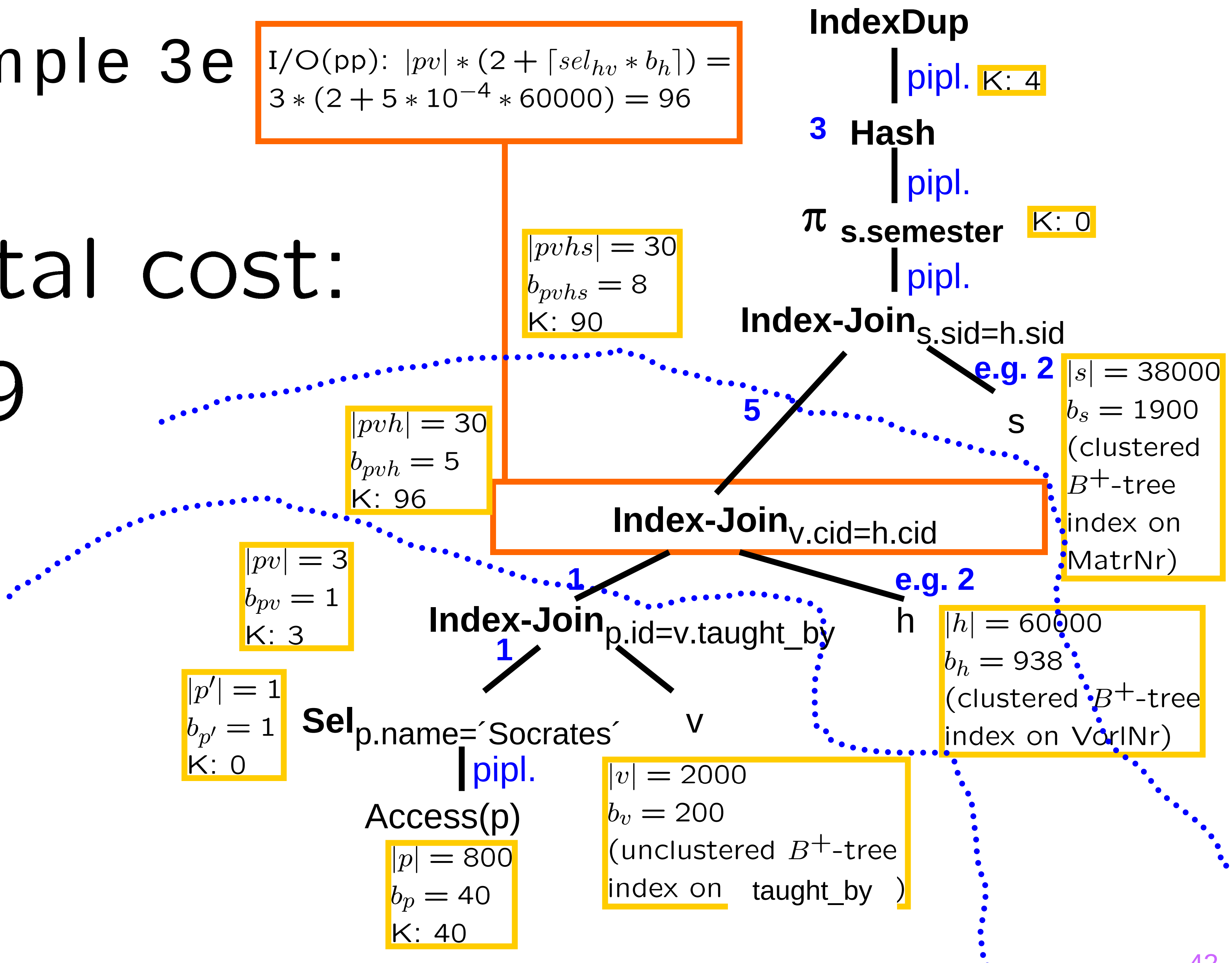
Example 3d

Total cost: 1131



Example 3e

Total cost:
289



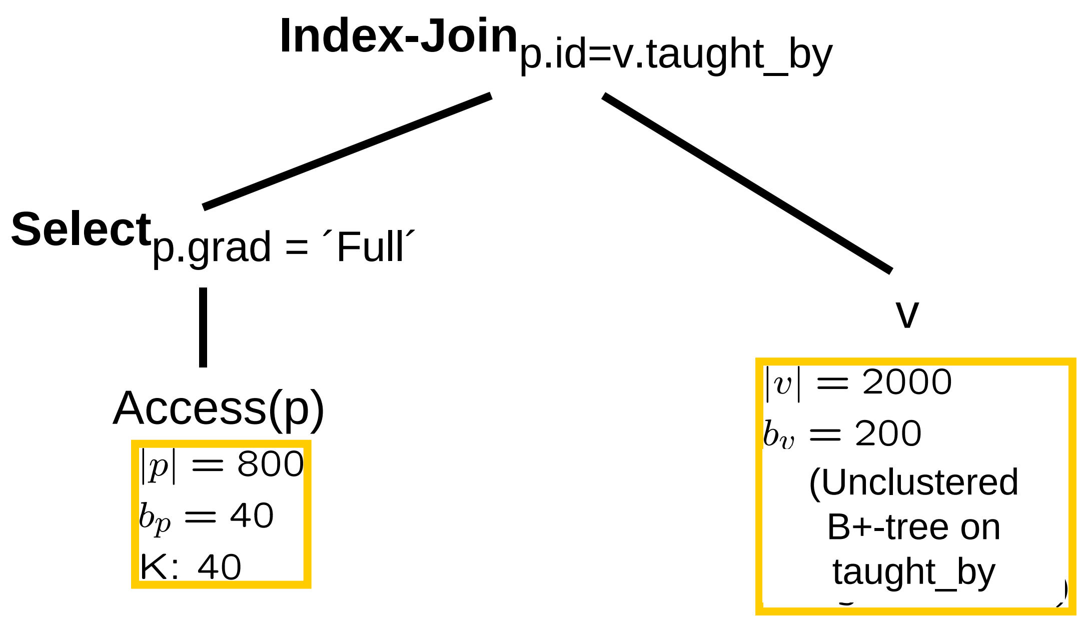
Observations

- Join order has BIG impact on query evaluation time!
- Cost of Operations and size of intermediate result can be computed separately.
  - The selection of a join operator has only local influence on the runtime of the query, except if it destroys properties of the data (such as sort order) that are important for operations later on in the pipeline or if it requires main memory buffers that are allocated to other operators.

Example 4: Selectivity estimates

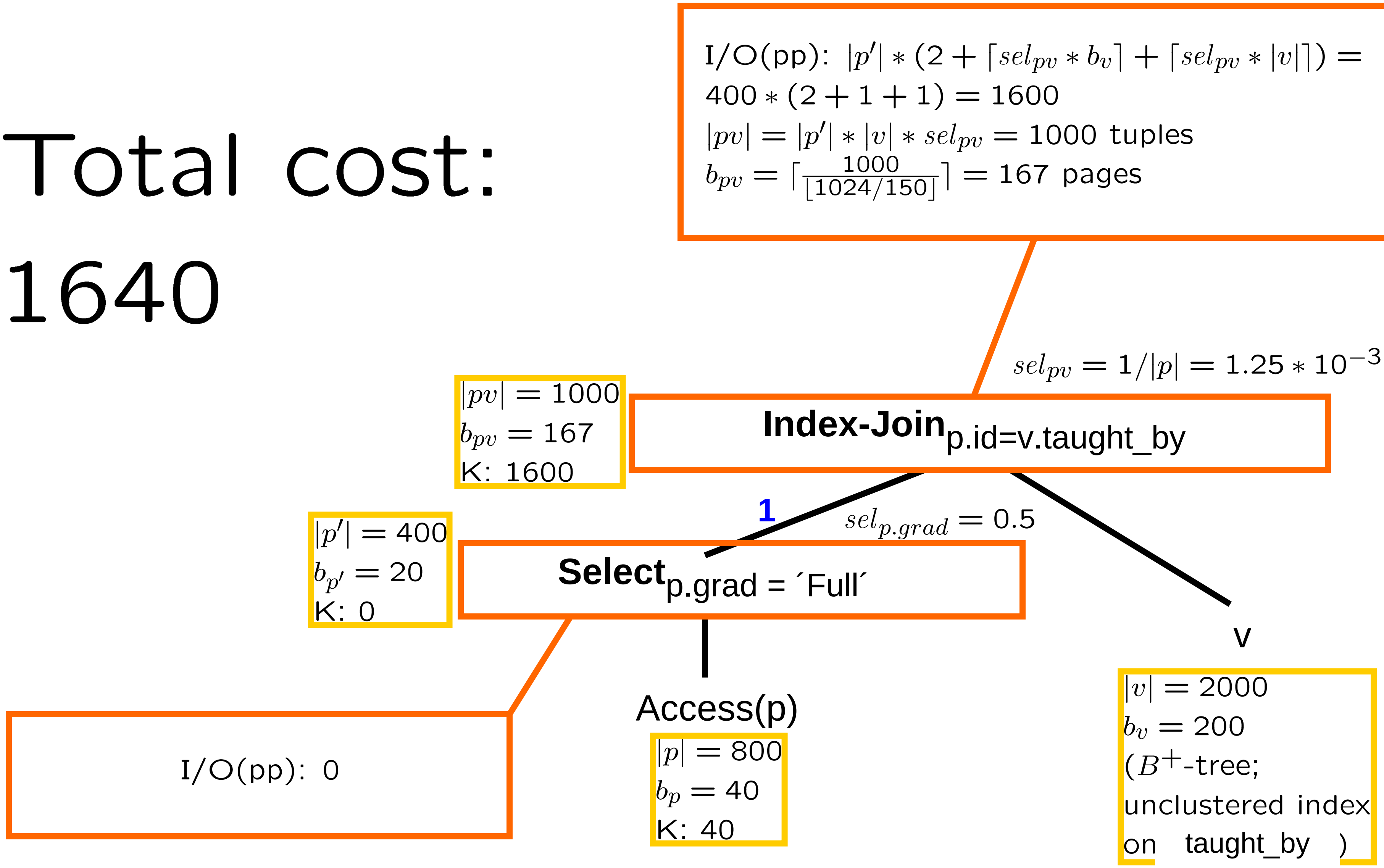
Now no selectivities are given, but we know that

- Professors (p) are in a 1:n relationship with Courses (v) (via foreign key taught_by).
- Each Professor has either rank „Full“ or „Assoc“.



Example 4a: Selectivity estimates

Total cost:
1640



Example 4b: Selectivity estimates

Total cost: 441

Cost: $b_{p'} - (m-2) + 1 + \lceil b_{p'}/(m-1) \rceil * (b_v - 1) =$
$20 - 18 + 1 + \lceil 20/19 \rceil * (200-1) = 401$

$sel_{pv} = 1/|p| = 1.25 * 10^{-3}$

$|pv| = 1000$
$b_{pv} = 167$
K: 401

NL-Join$_{\text{p.id=v.taught\_by}}$

$|p'| = 400$
$b_{p'} = 20$
K: 0

$sel_{p.grad} = 0.5$

Select$_{\text{p.grad = ´Full´}}$

v

Access(p)

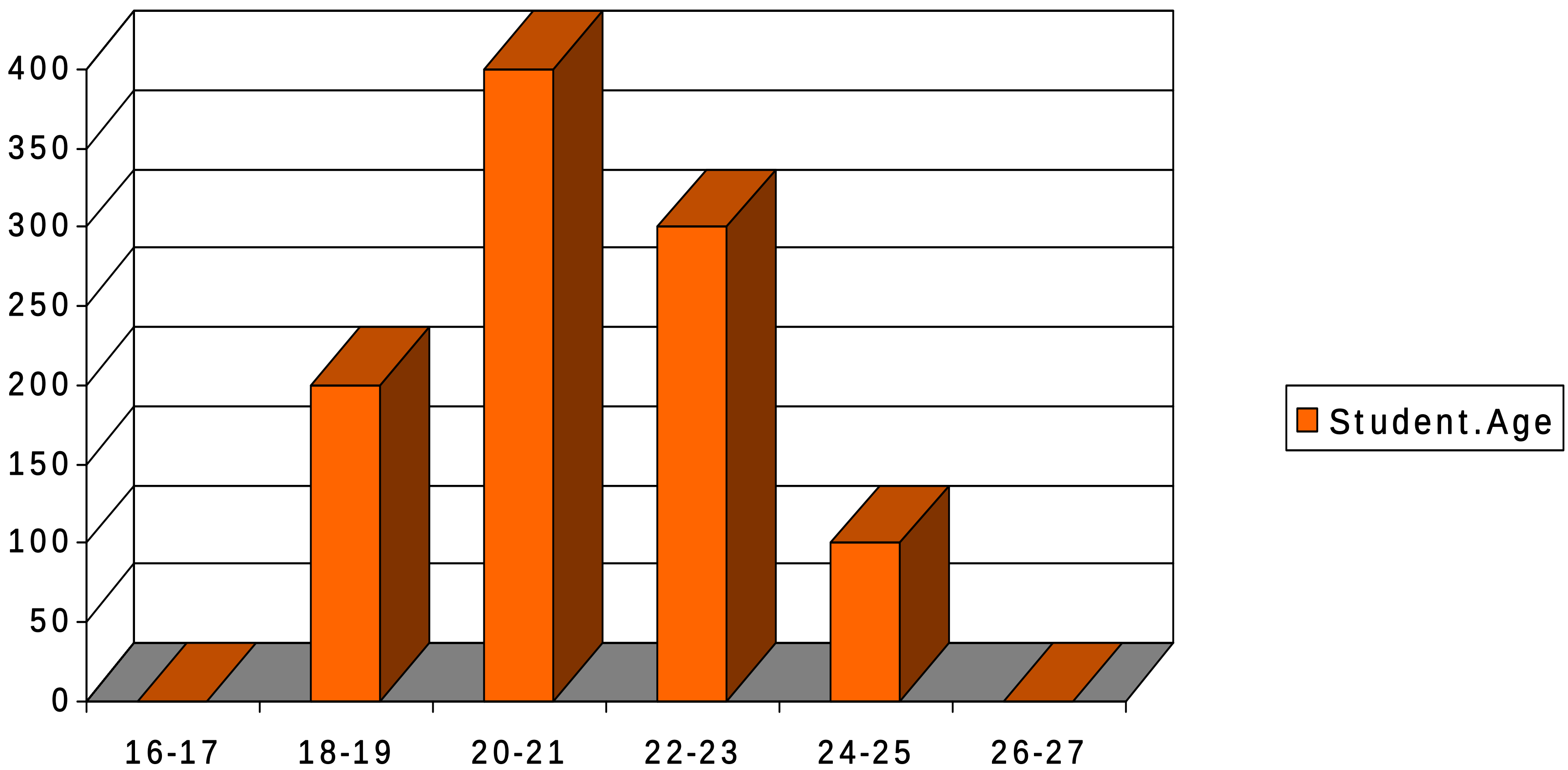
$|p| = 800$
$b_p = 40$
K: 40

$|v| = 2000$
$b_v = 200$
(Unclustered B+-tree on taught_by)

It is better NOT to Use the index!

Using Histograms

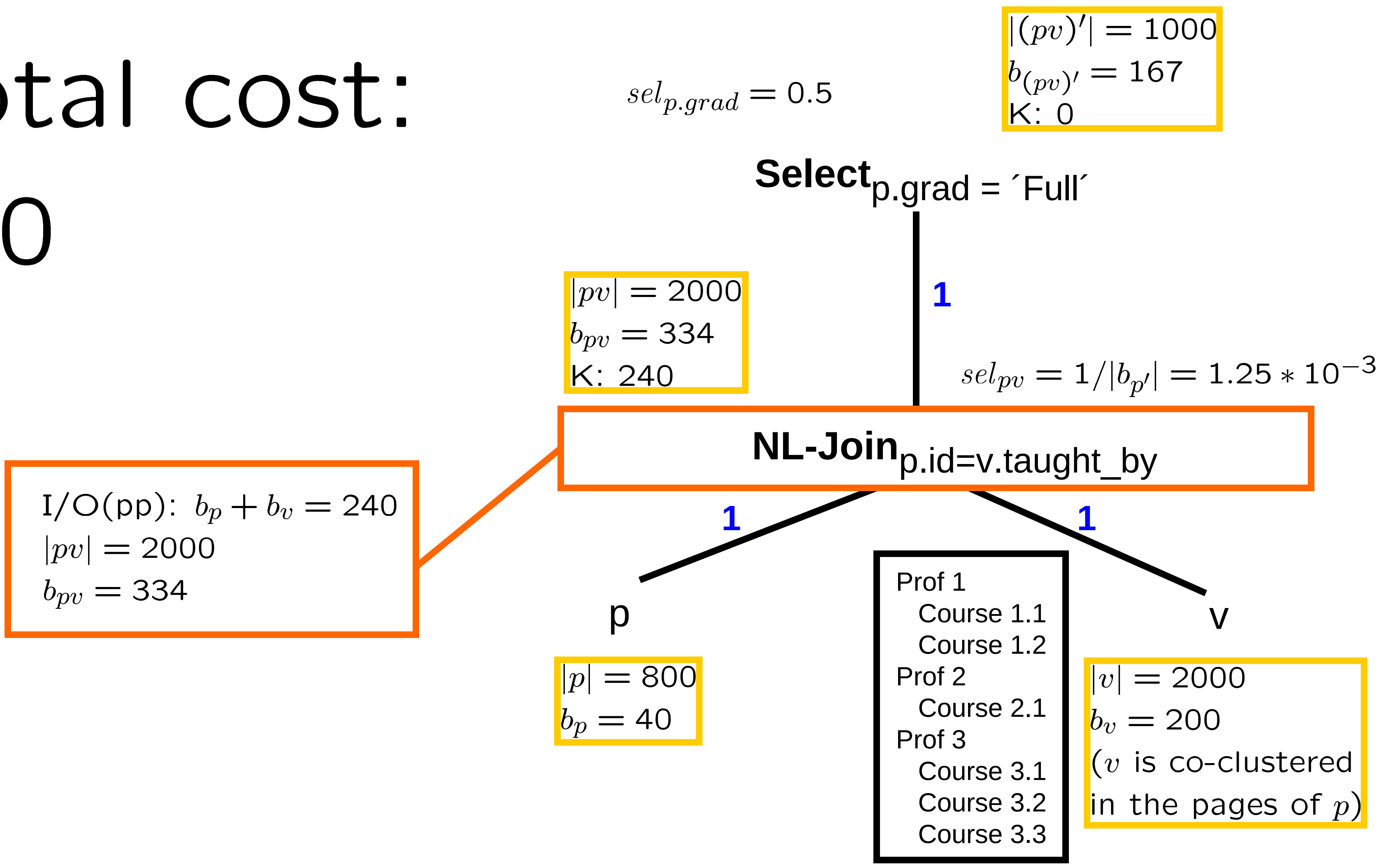
Histogram computed by „analyze table“ statement



$$sel_{student.age=22} = (300/2)/(0+200+400+300+100+0) = 0.15$$

Example 5: (co-)Clusters

Total cost: 240



Example 6: Seeks

Seek+latency time: 8ms

Transfer time: 0.067 ms/page (15 MB/s)

Total cost:

$441 * 0.067 + 5 * 8 = 69.547$ ms