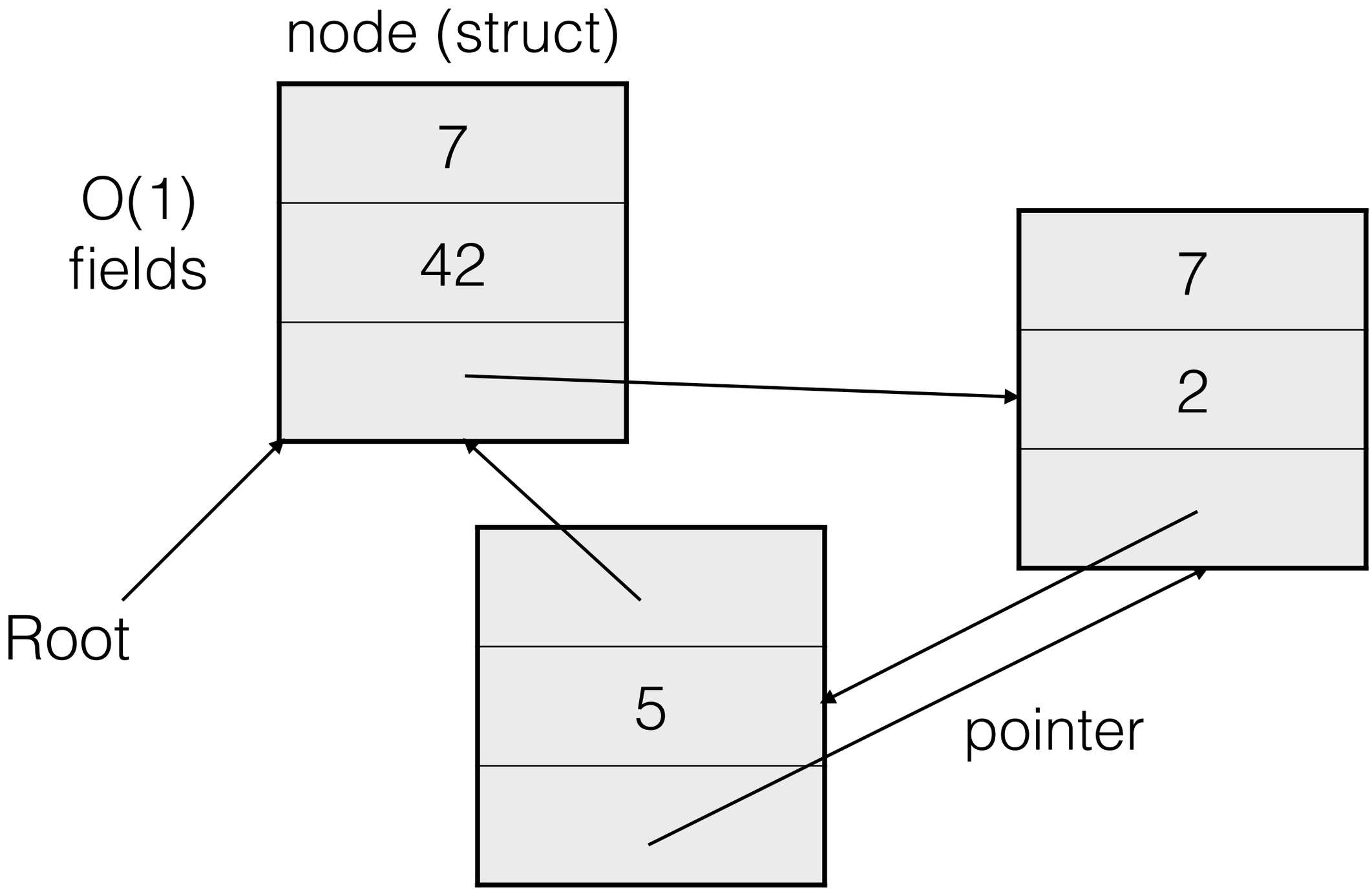
# Partial Persistent DS with O(1)-space/time

0116035 石奕心

# Pointer Machine



# Pointer Machine

- x = new node
- x = y.field
- x.field = y
- x = y + z  etc. (data operation)
- destroy x ( if no pointer points to it )

# Persistence

- keep all versions of DS
- DS operations relative to specified version
- update creates (& returns) new versions
- partial persistence — update only latest version
  => versions linearly ordered

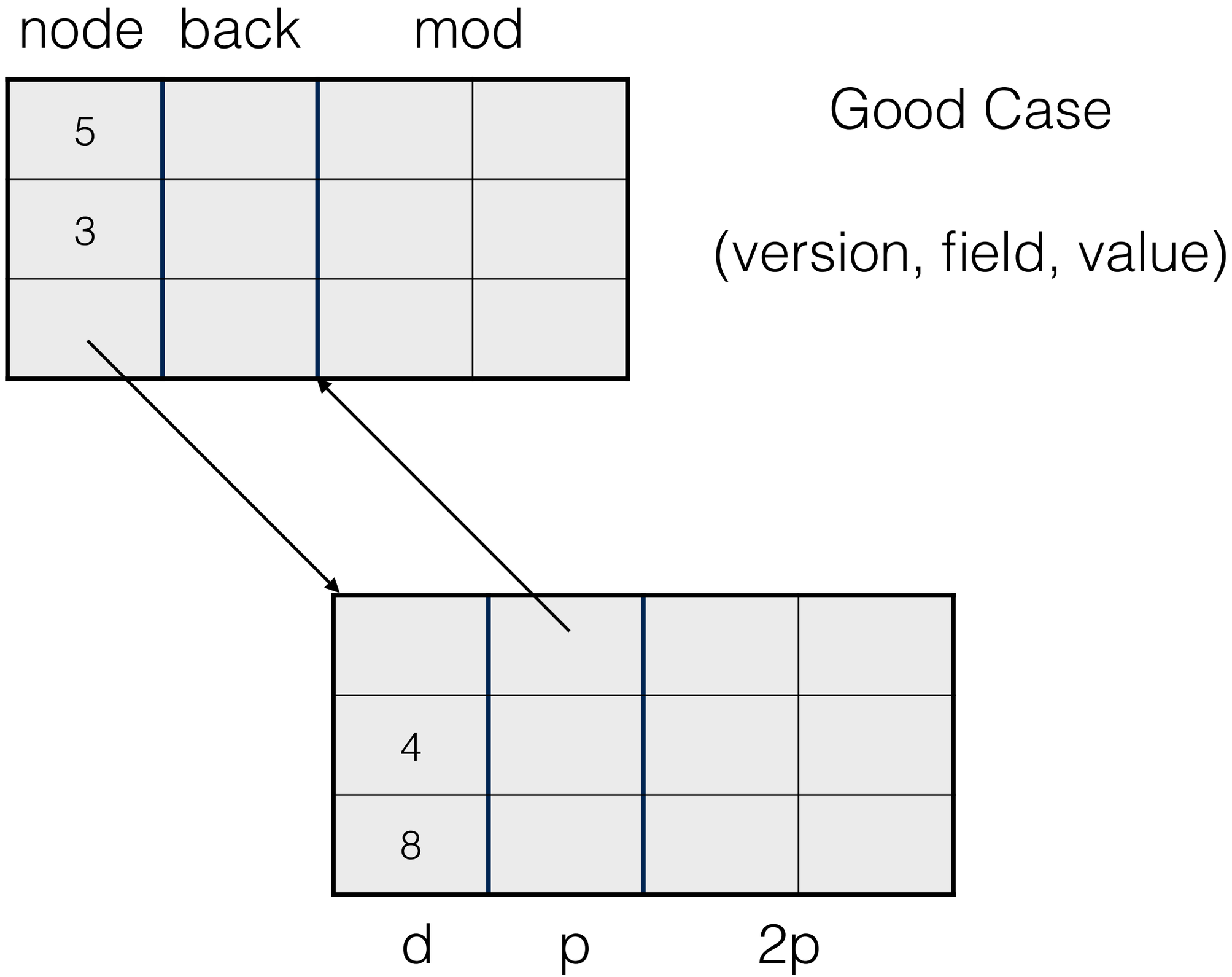
# Partial Persistence

- Any pointer-machine DS with ≤ p = O(1) pointers to any node can be made partially persistent with O(1) amortized multiplicative overhead & O(1) space per change

| node | back | | mod |
|---|---|---|---|
| 5 | | | |
| 3 | | | |
| | | | |

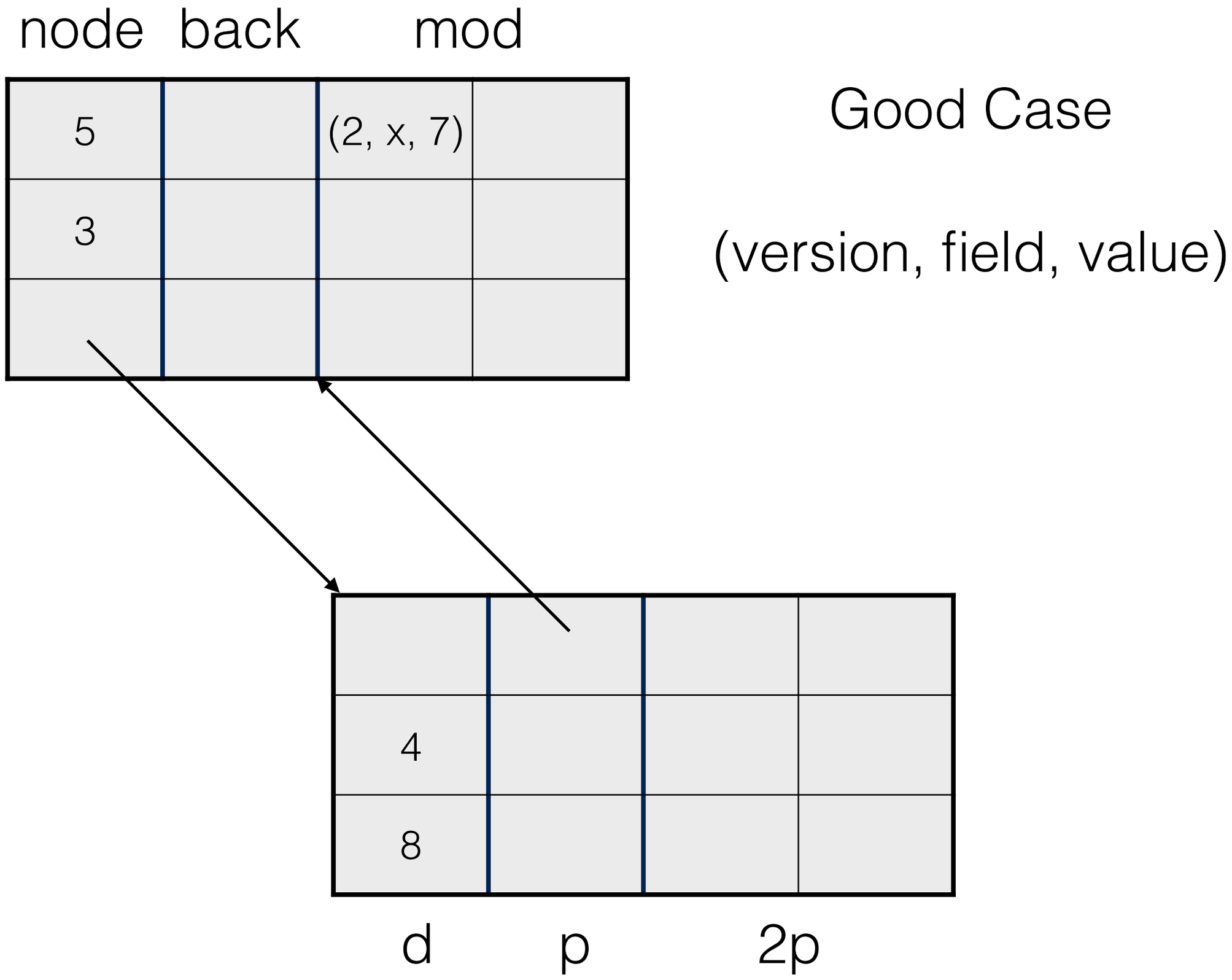
| | | | |
|---|---|---|---|
| | | | |
| 4 | | | |
| 8 | | | |

d p 2p




Good Case

(version, field, value)




Good Case

(version, field, value)

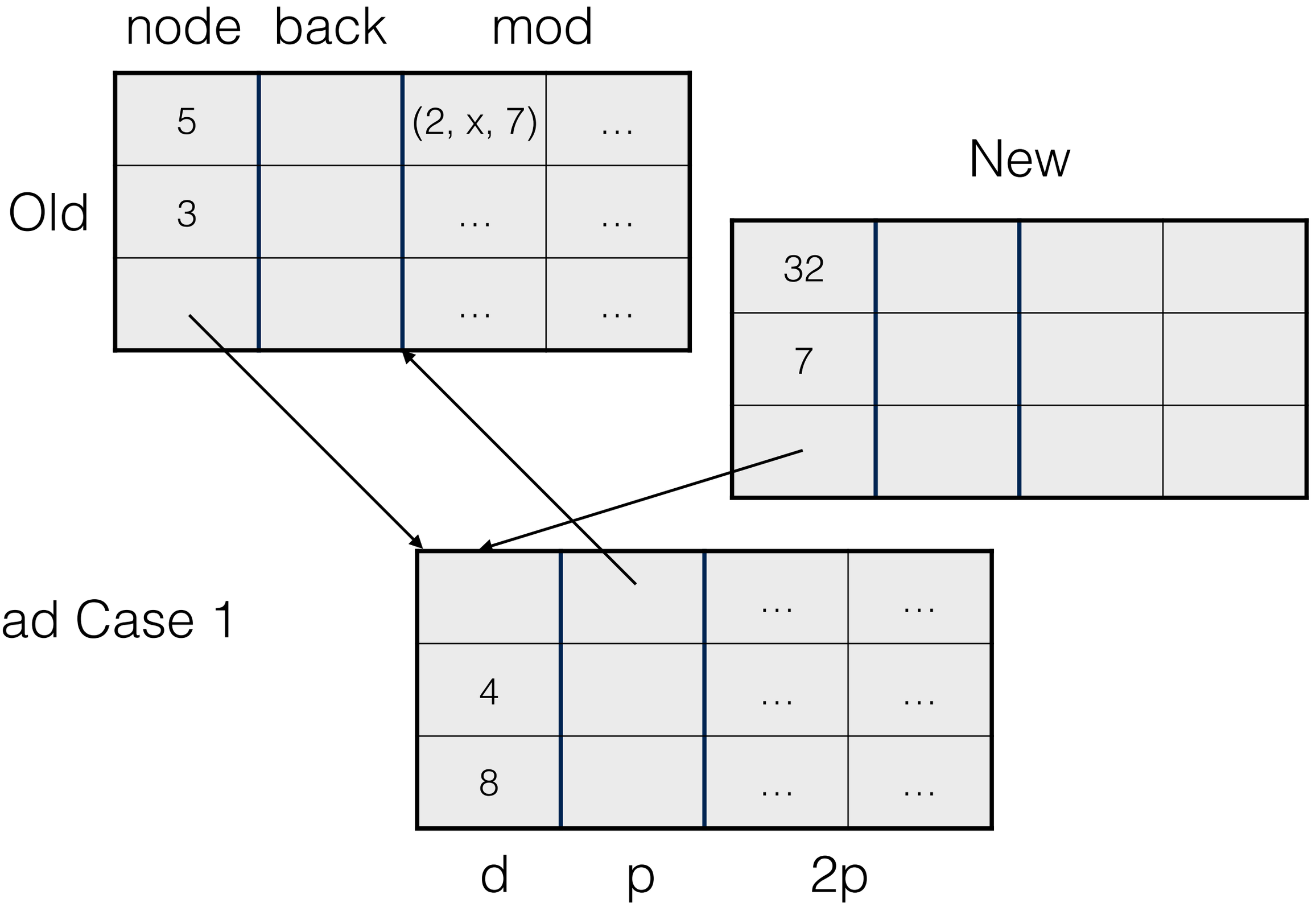
Bad Case

| node | back | mod | |
|---|---|---|---|
| 5 | | (2, x, 7) | … |
| 3 | | … | … |
| | | … | … |

| | | | |
|---|---|---|---|
| | | … | … |
| 4 | | … | … |
| 8 | | … | … |

d p 2p

Bad Case 1



| | node | back | mod | |
|---|---|---|---|---|
| Old | 5 | | (2, x, 7) | … |
| | 3 | | … | … |
| | | | … | … |

New

| | | | |
|---|---|---|---|
| 32 | | | |
| 7 | | | |
| | | | |

| | | | |
|---|---|---|---|
| | | … | … |
| 4 | | … | … |
| 8 | | … | … |

d p 2p

Bad Case 1

Directly Update Back Pointer

Bad Case

| node | back | mod | |
|---|---|---|---|
| 5 | | (2, x, 7) | … |
| 3 | | … | … |
| | | … | … |

| | | | |
|---|---|---|---|
| | | … | … |
| 4 | | … | … |
| 8 | | … | … |

d p 2p

Bad Case 2

???

| node | back | mod | |
|---|---|---|---|
| 5 | | (2, x, 7) | … |
| 3 | | … | … |
| | | … | … |

New

| | | | |
|---|---|---|---|
| | | | |
| 9 | | | |
| 10 | | | |

| | | | |
|---|---|---|---|
| | | … | … |
| 4 | | … | … |
| 8 | | … | … |

Old

d p 2p

???

| node | back | mod | |
|---|---|---|---|
| 5 | | (2, x, 7) | … |
| 3 | | … | … |
| | | … | … |

New

| | | | |
|---|---|---|---|
| | | | |
| 9 | | | |
| 10 | | | |

Bad Case 2

Recursively Update Pointer

| | | | |
|---|---|---|---|
| | | … | … |
| 4 | | … | … |
| 8 | | … | … |

Old

d p 2p

# Analysis

- if node_not_full: add mod good case
  else: - create node' = node with mods applied
  - change back pointer to node -> node'
  - recursively change pointers to node -> node'

- potential $\Phi = c * \Sigma$ (#mods)

  actual $\Delta(\Phi)$
- amortized_cost(n) $\leq c + c +$ [- 2cp + p * amortized_cost(x)]

  $\Delta(\Phi)$ update pointer
- recursion will end. When a recursion happens potential decrease at least cp = -2cp + cp

  empty node at most c recursion