

Single and Bulk Updates in Stratified Trees

Eljas Soisalon-Soininen and Peter Widmayer

Search trees

- Important part of IT
 - Fast web & database searches
- Rebalancing
- Logarithmical access cost

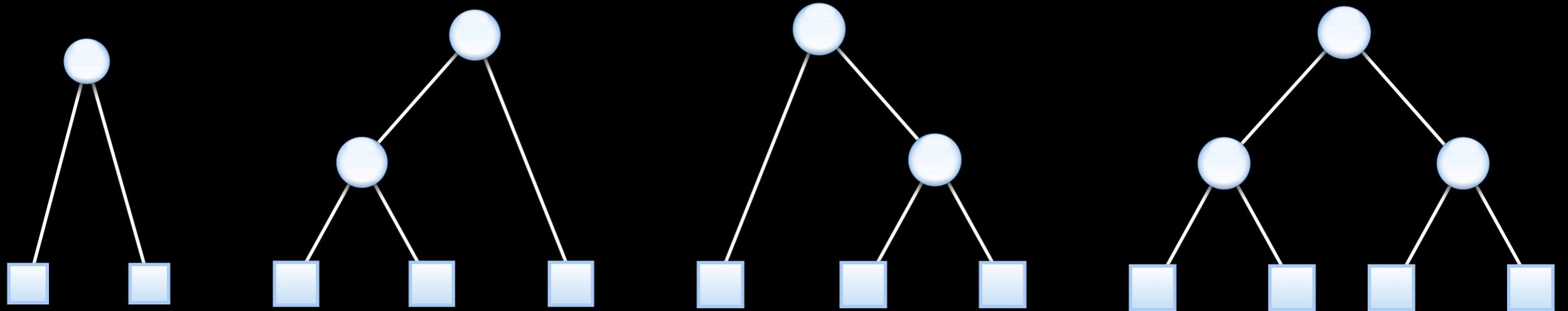
Relaxed rebalancing

- Concurrent systems
 - Problem during link changes
 - Locked nodes
 - Cache misses & Disk accesses
- Rebalancing occurs when there is time
 - Updates may cancel each other

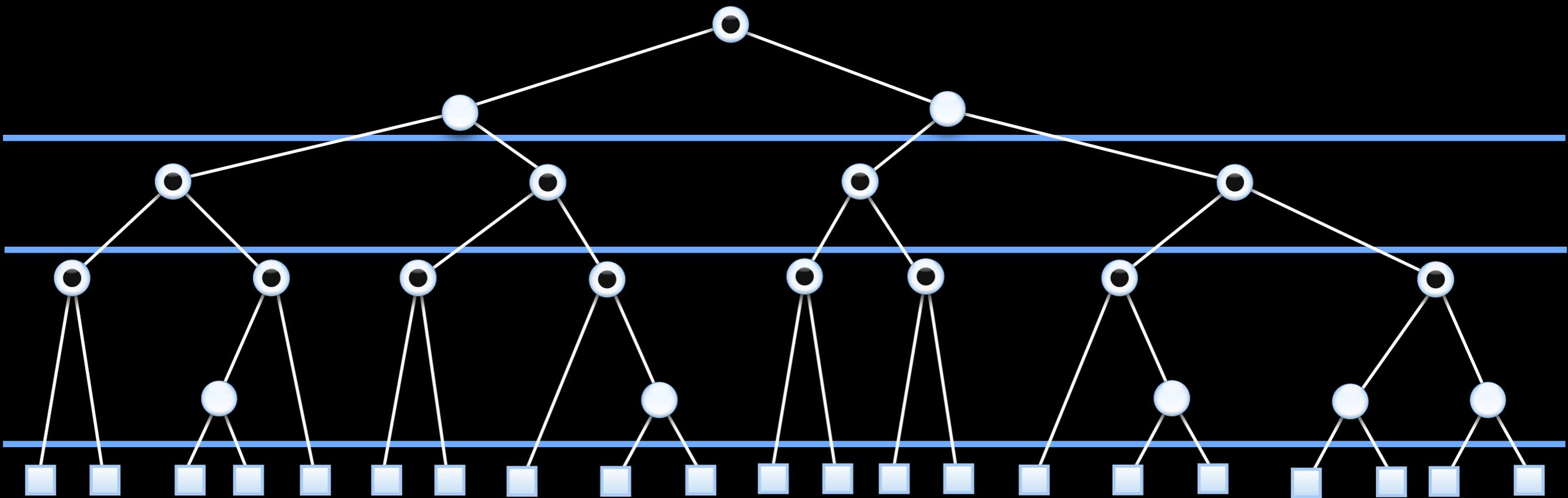
Stratified Trees

- One class of stratified trees
 - Ottman & Wood
 - Rebalancing with constant linkage cost
- Basically consists of:
 - Component trees
 - Layers

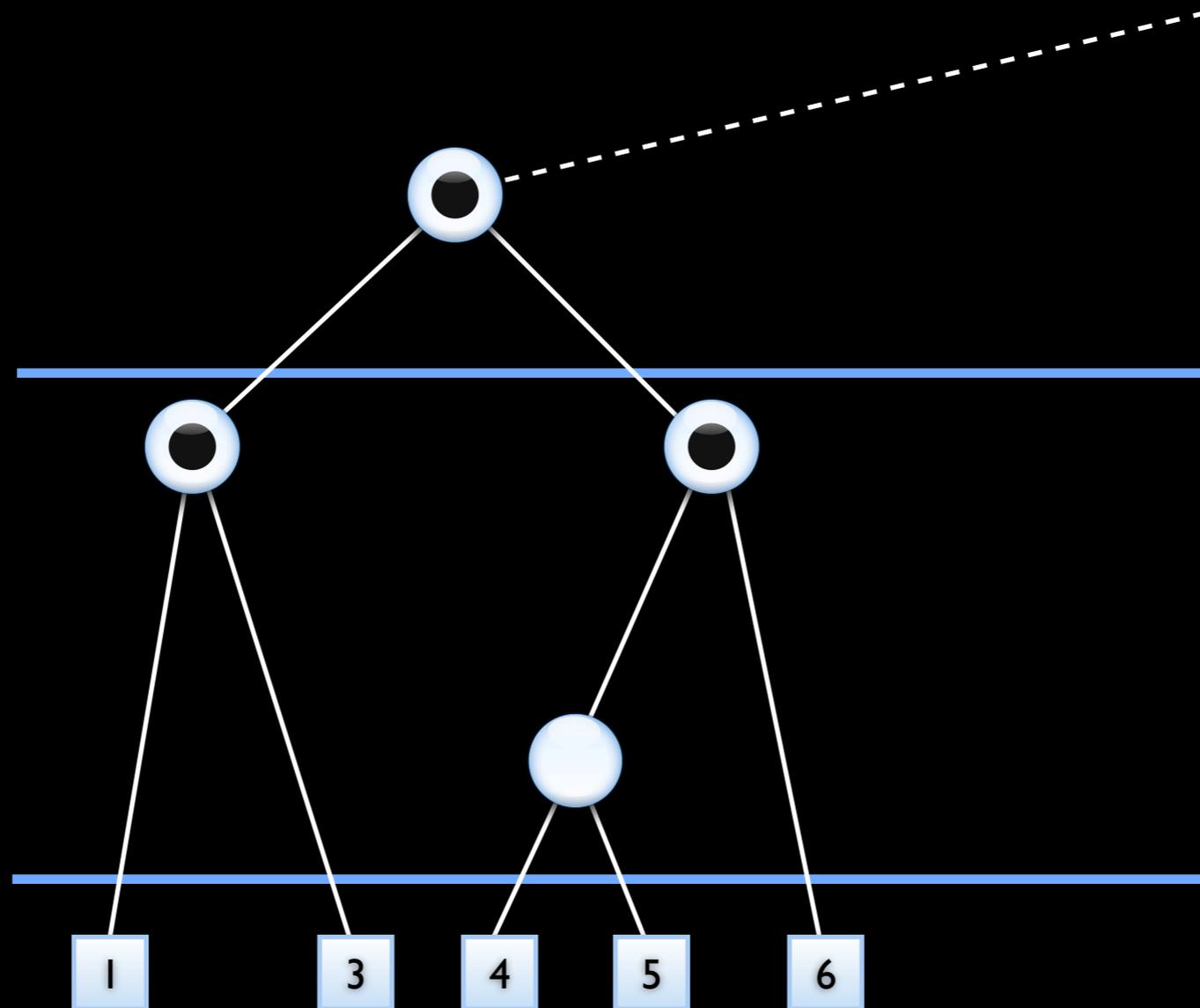
Component trees



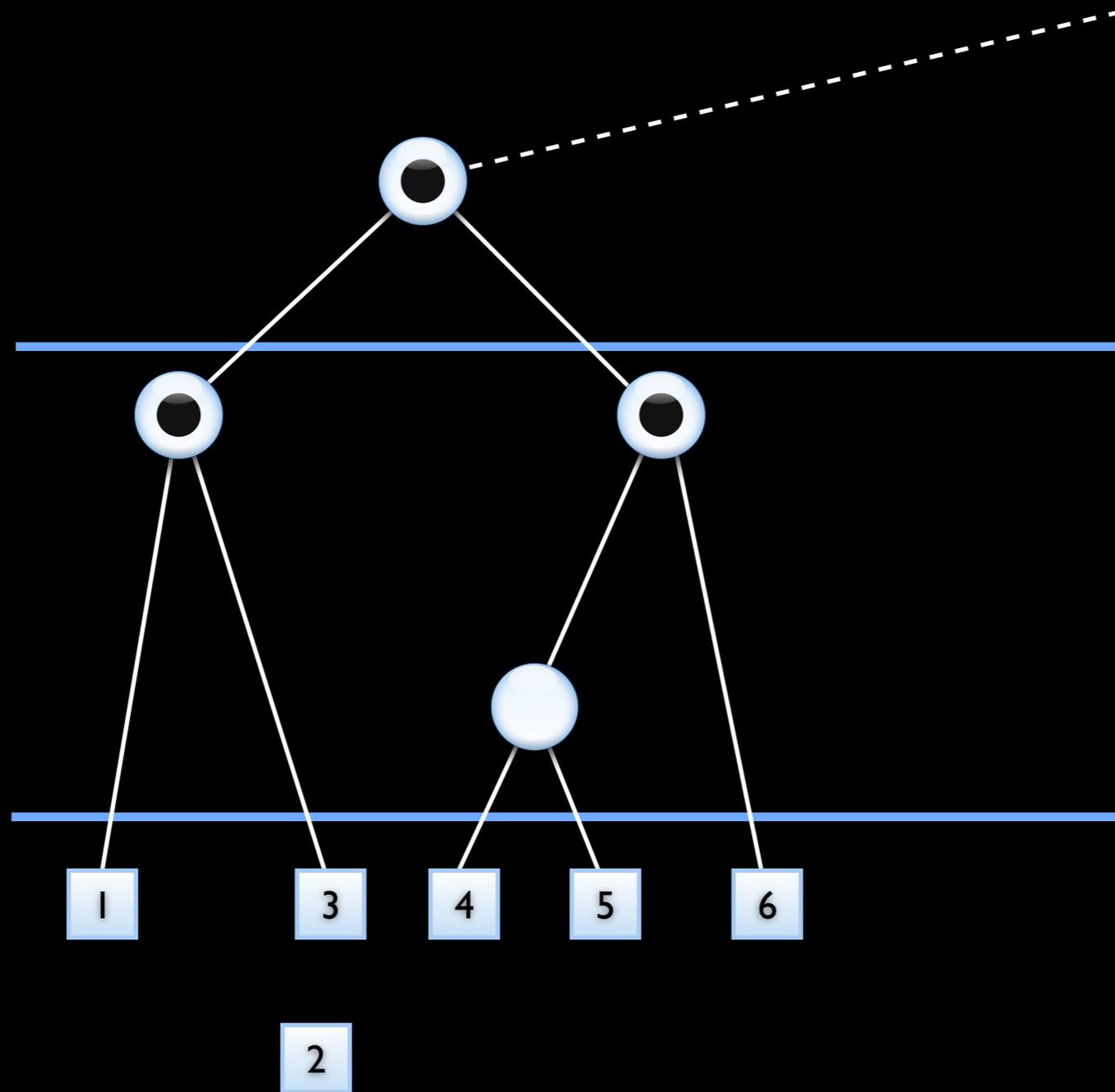
Stratified Tree



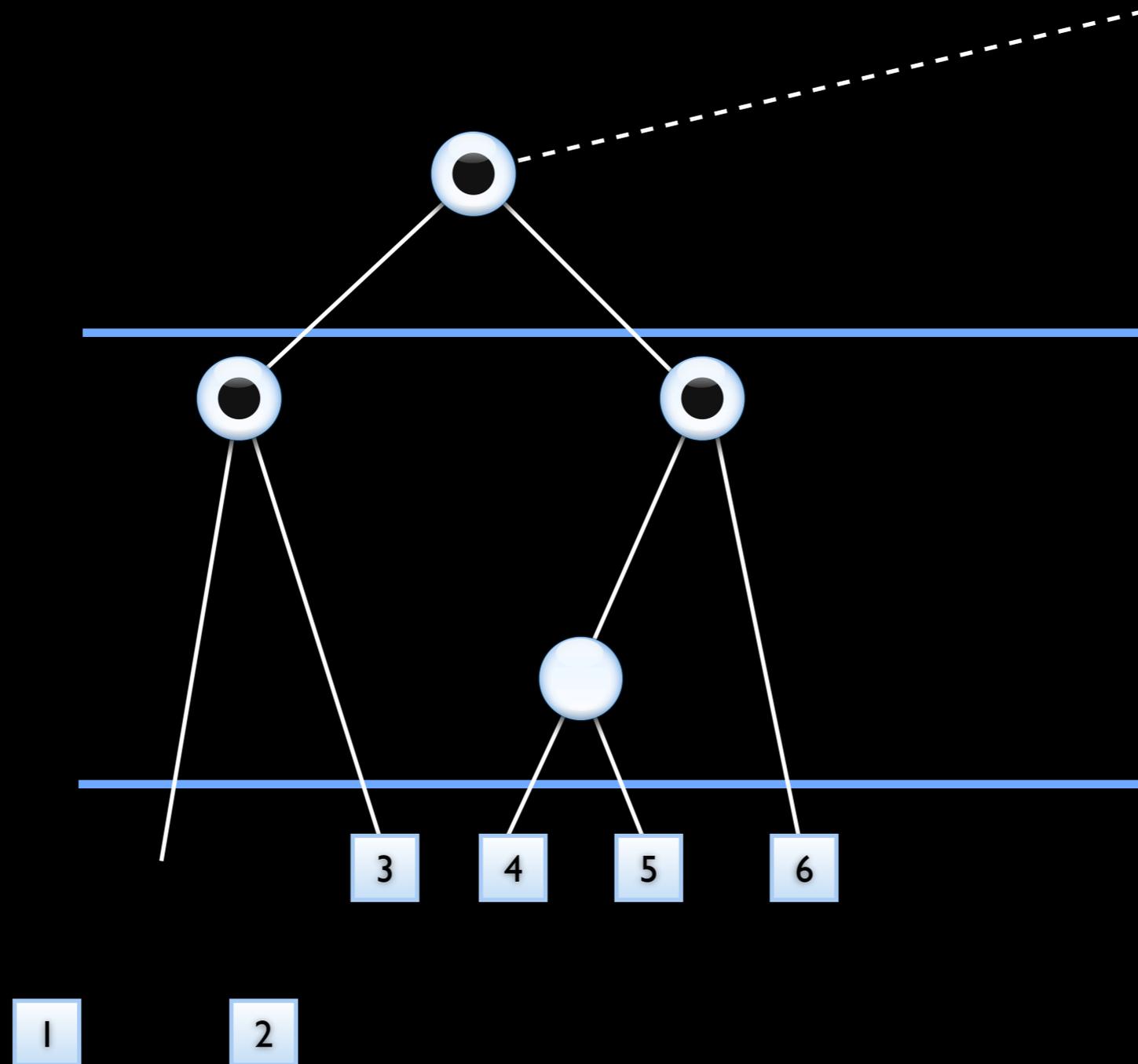
Insertion in Stratified Trees



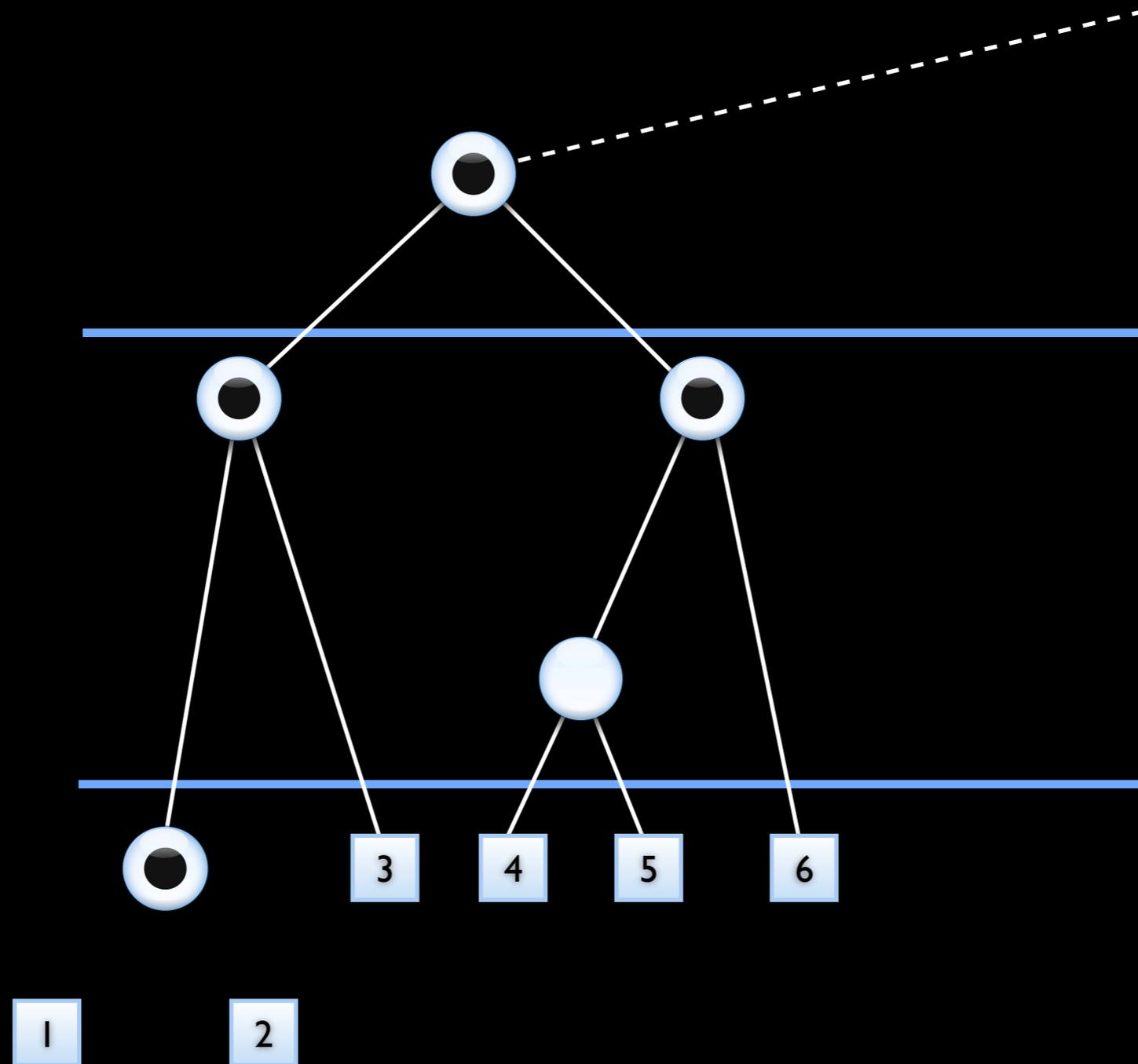
Insertion in Stratified Trees



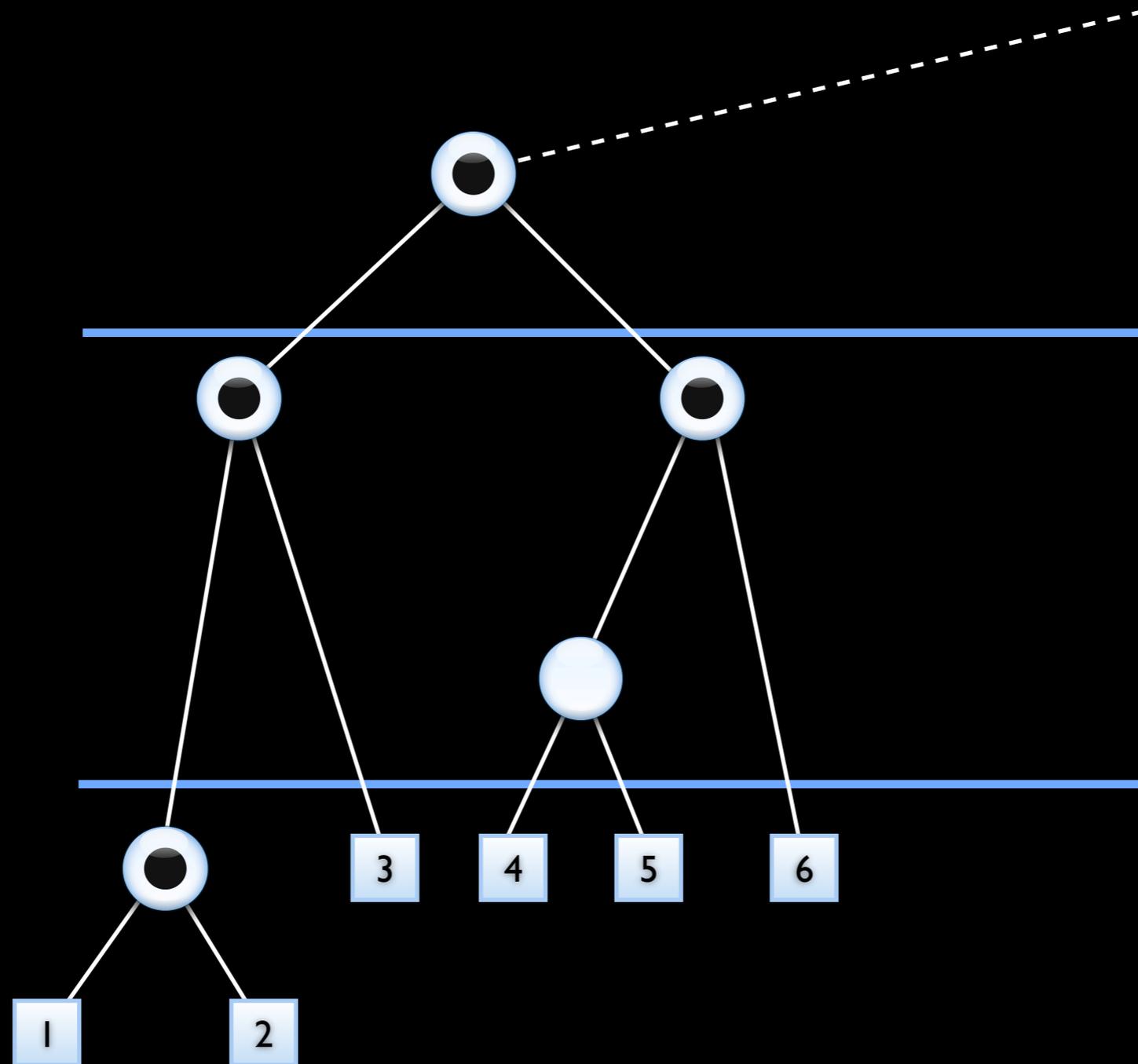
Insertion in Stratified Trees



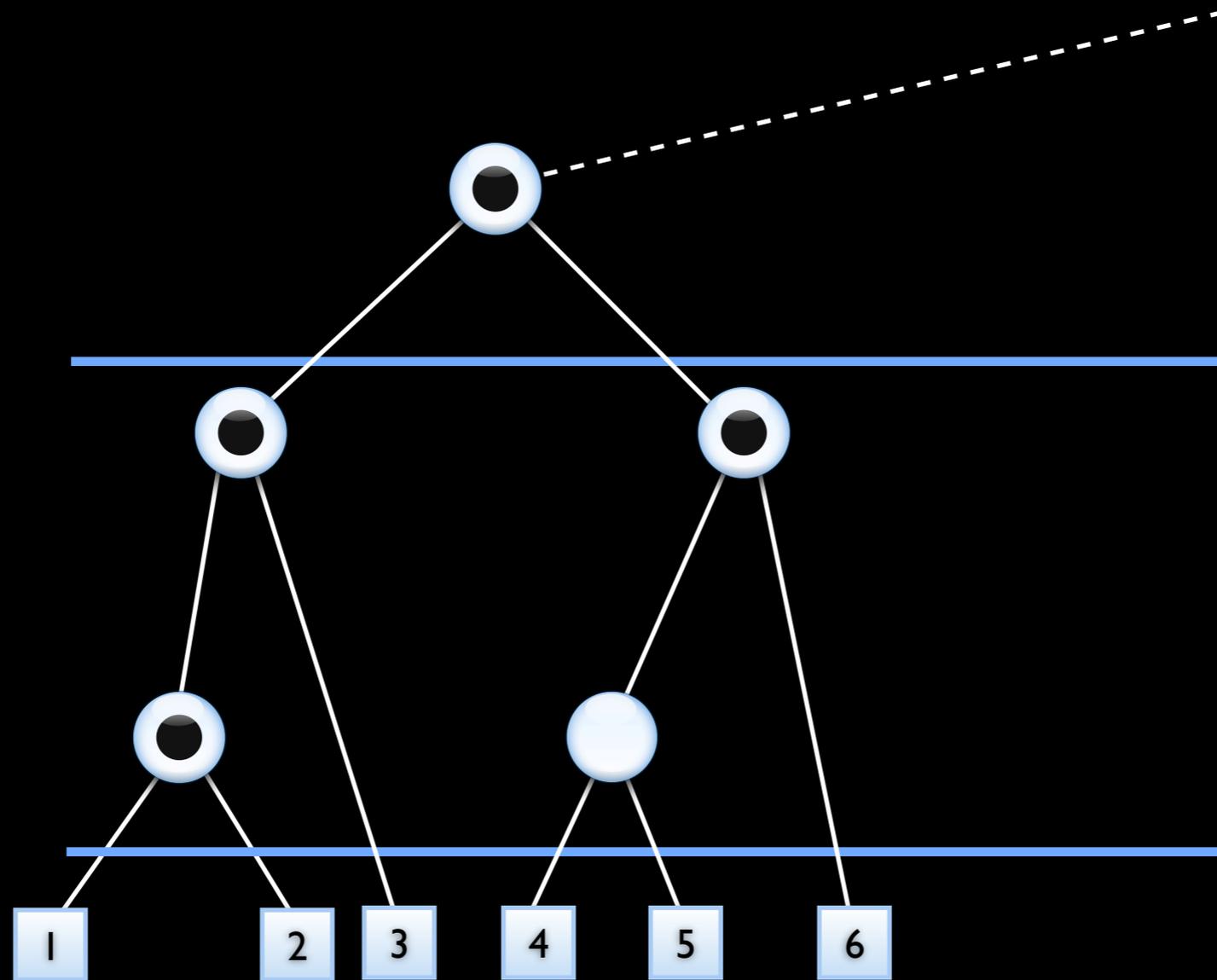
Insertion in Stratified Trees



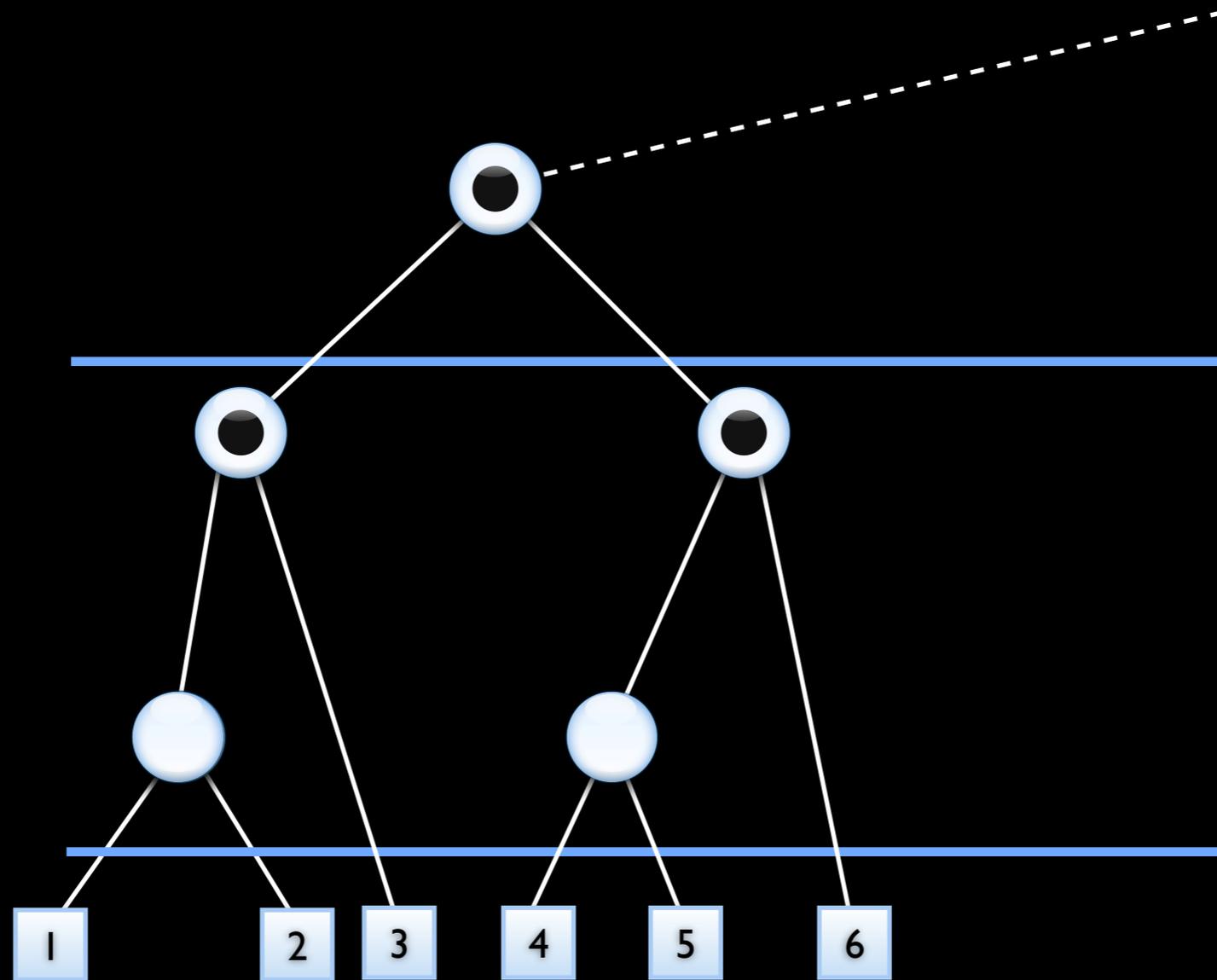
Insertion in Stratified Trees



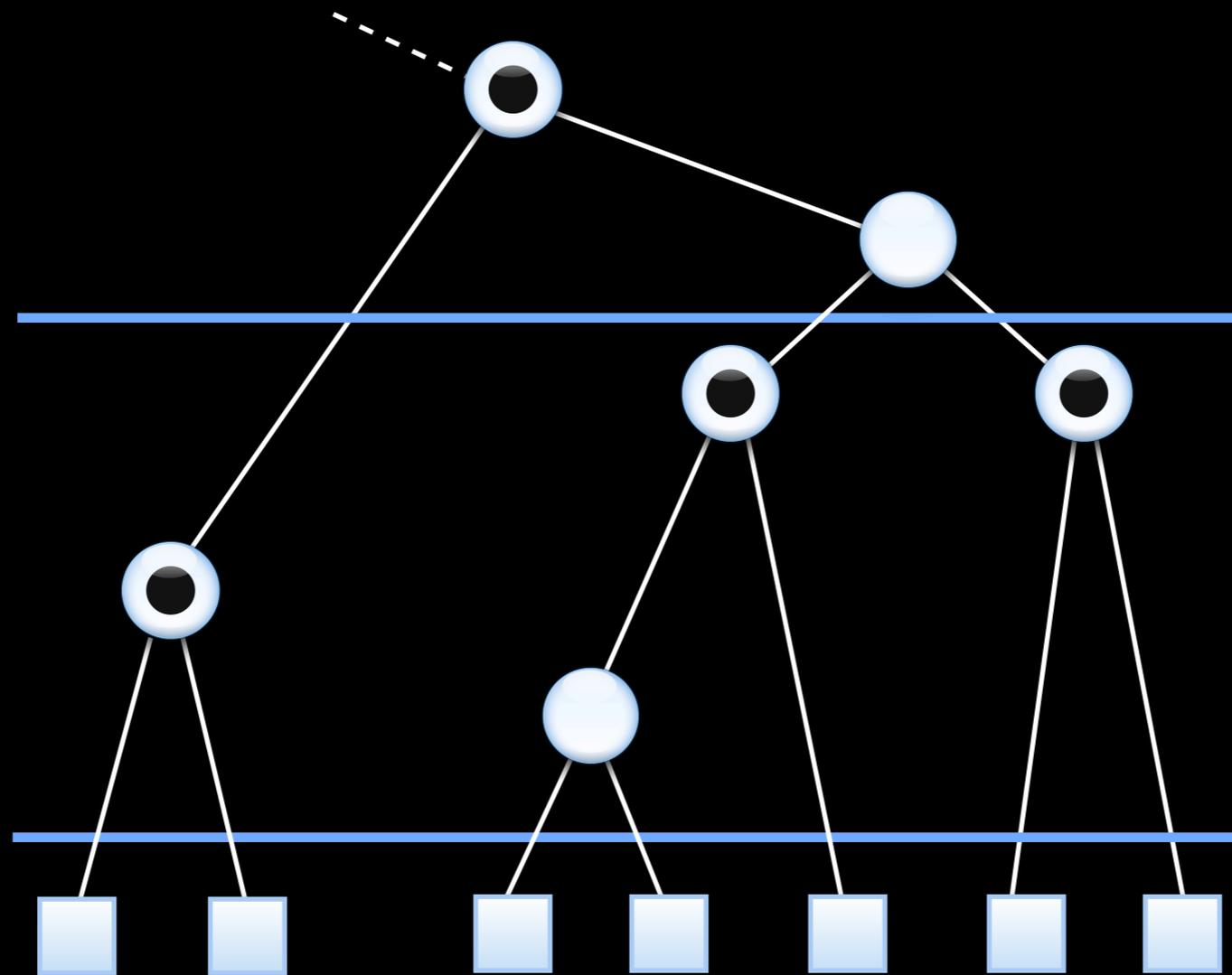
Insertion in Stratified Trees



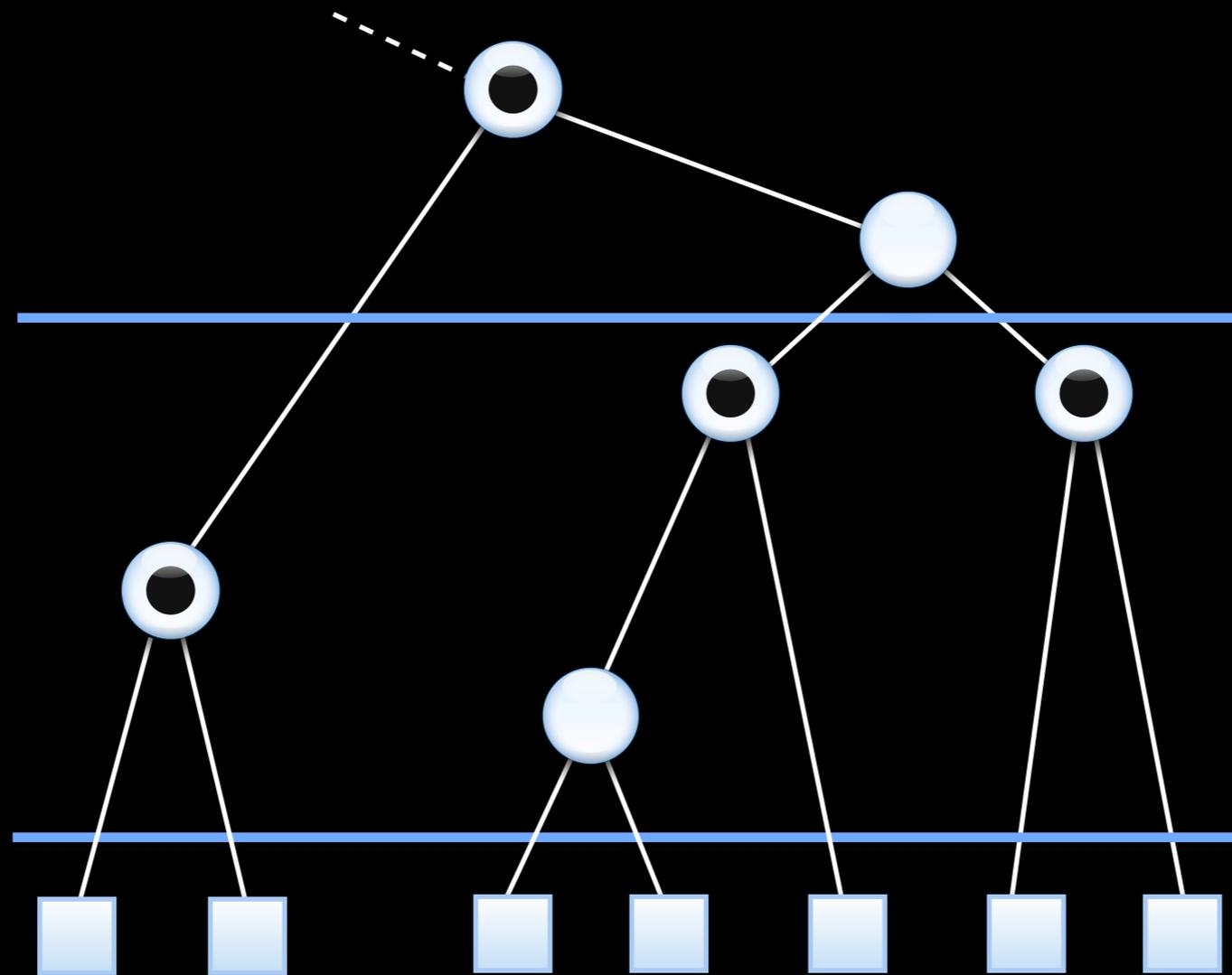
Insertion in Stratified Trees



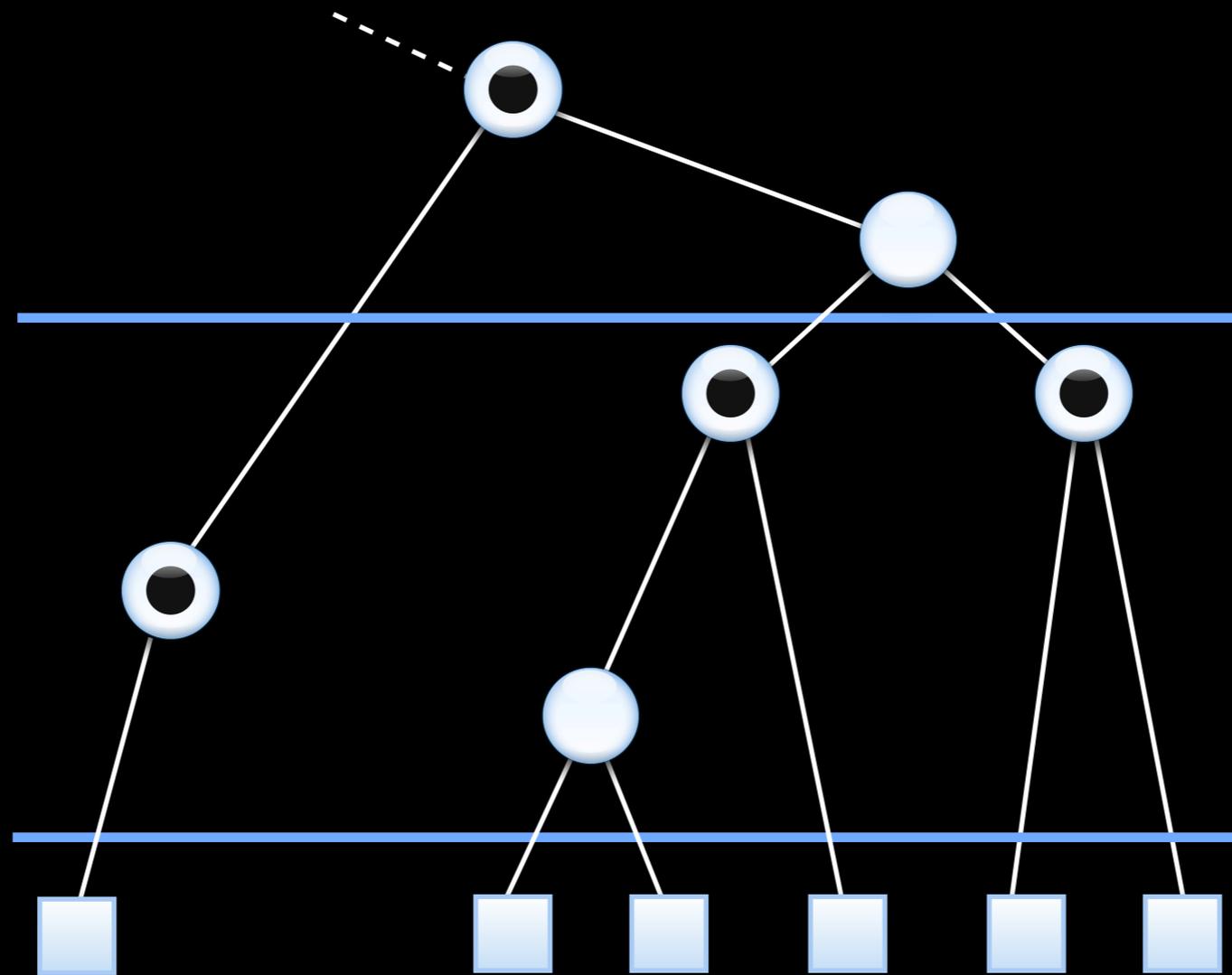
Deletion Case I



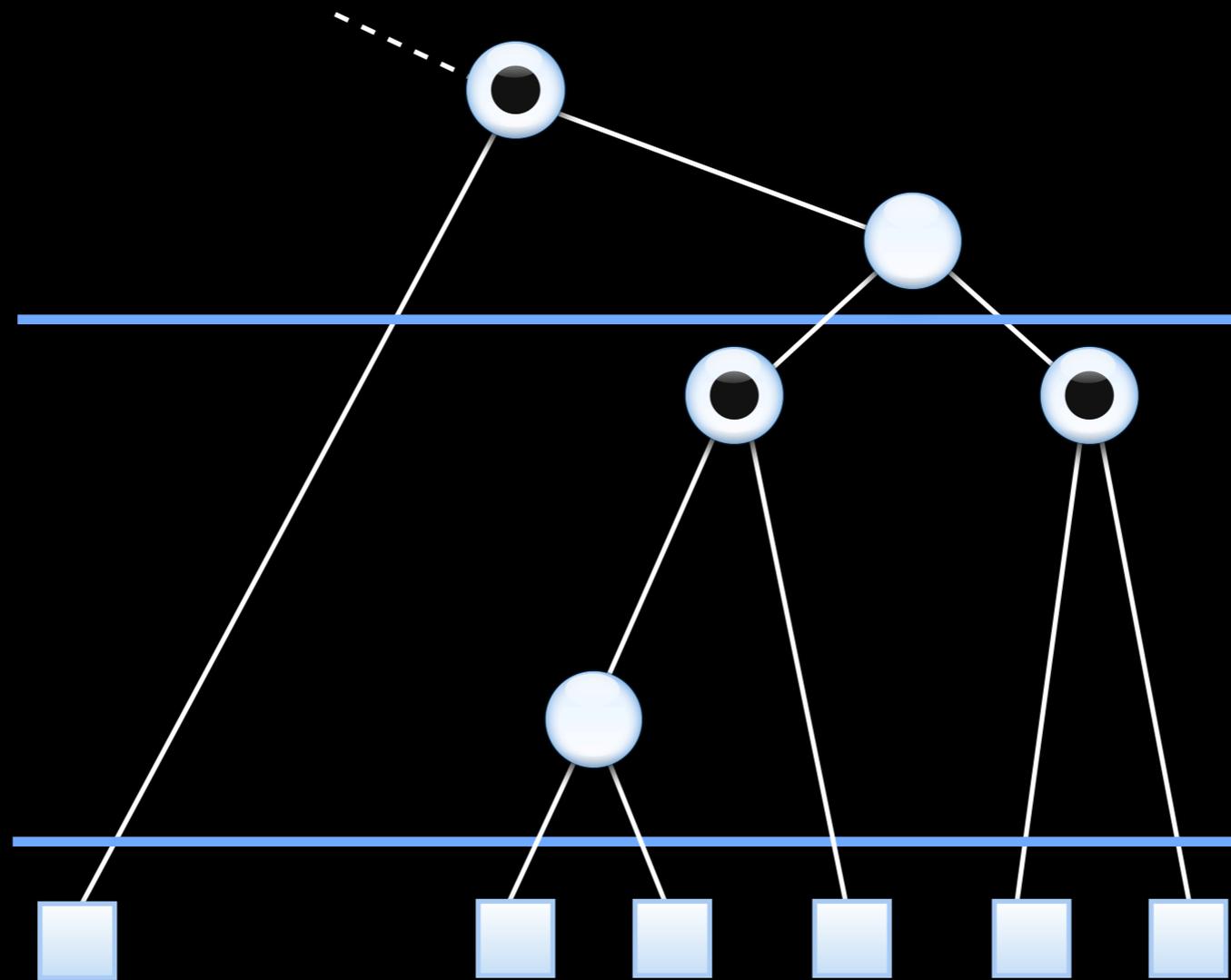
Deletion Case I



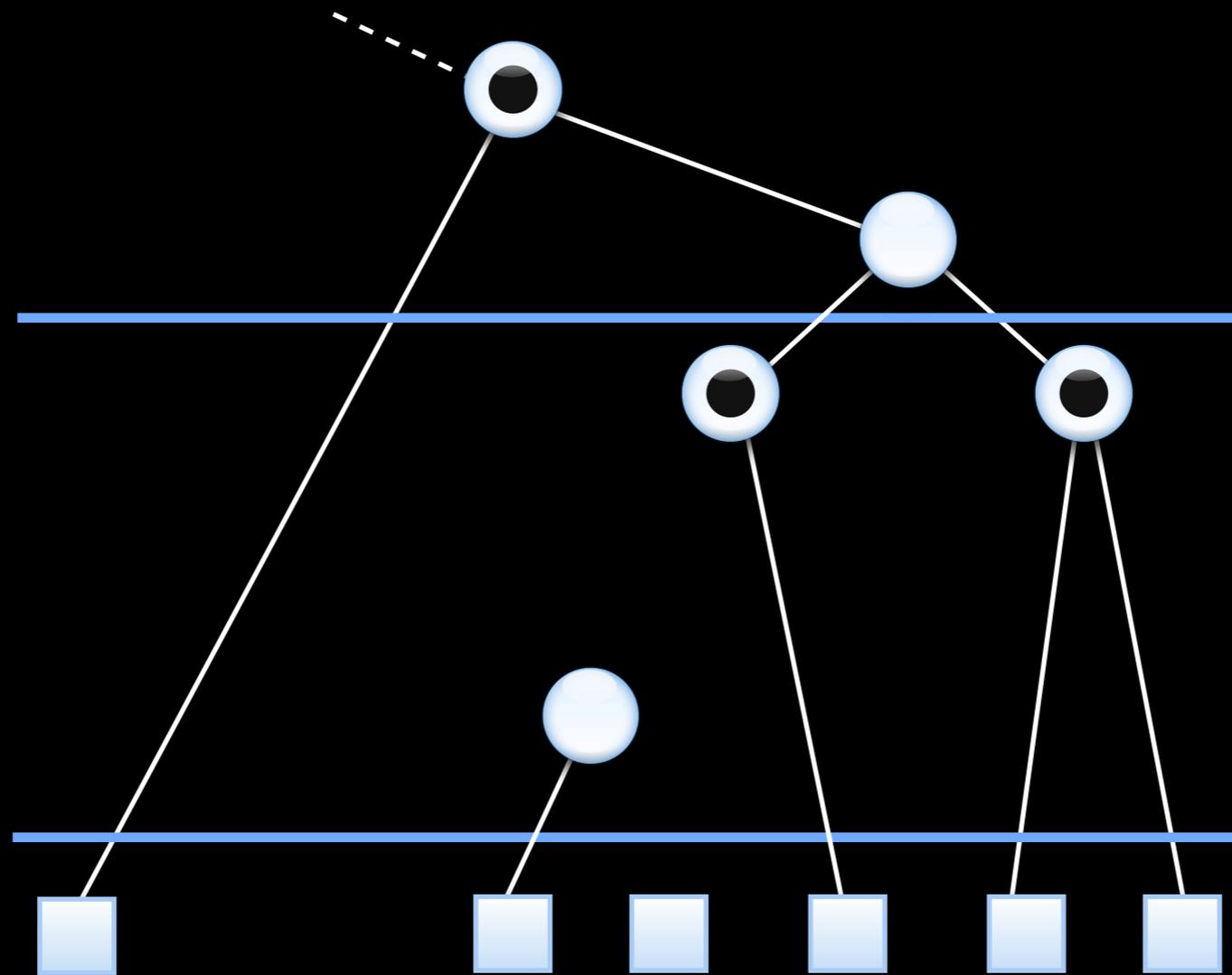
Deletion Case I



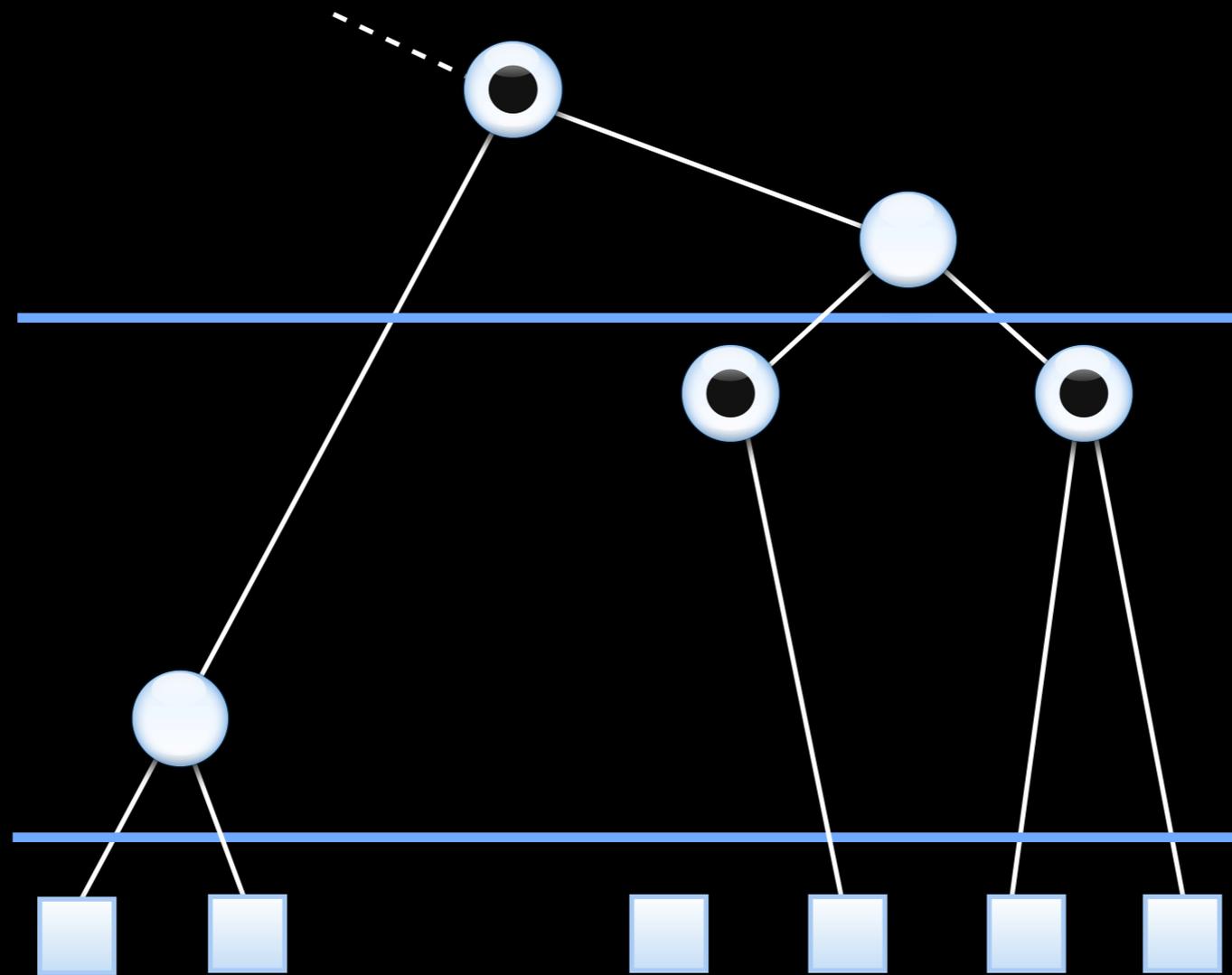
Deletion Case I



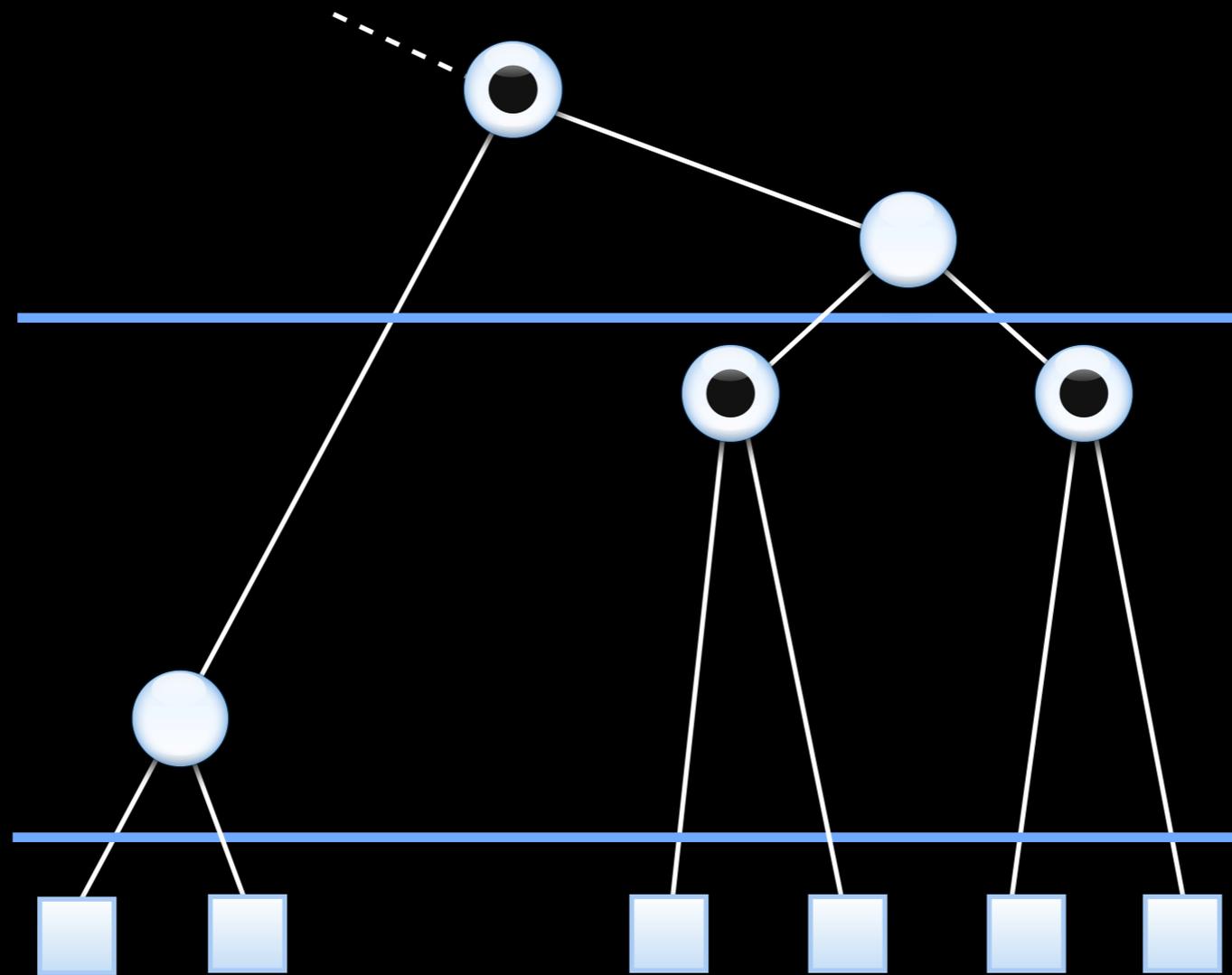
Deletion Case I



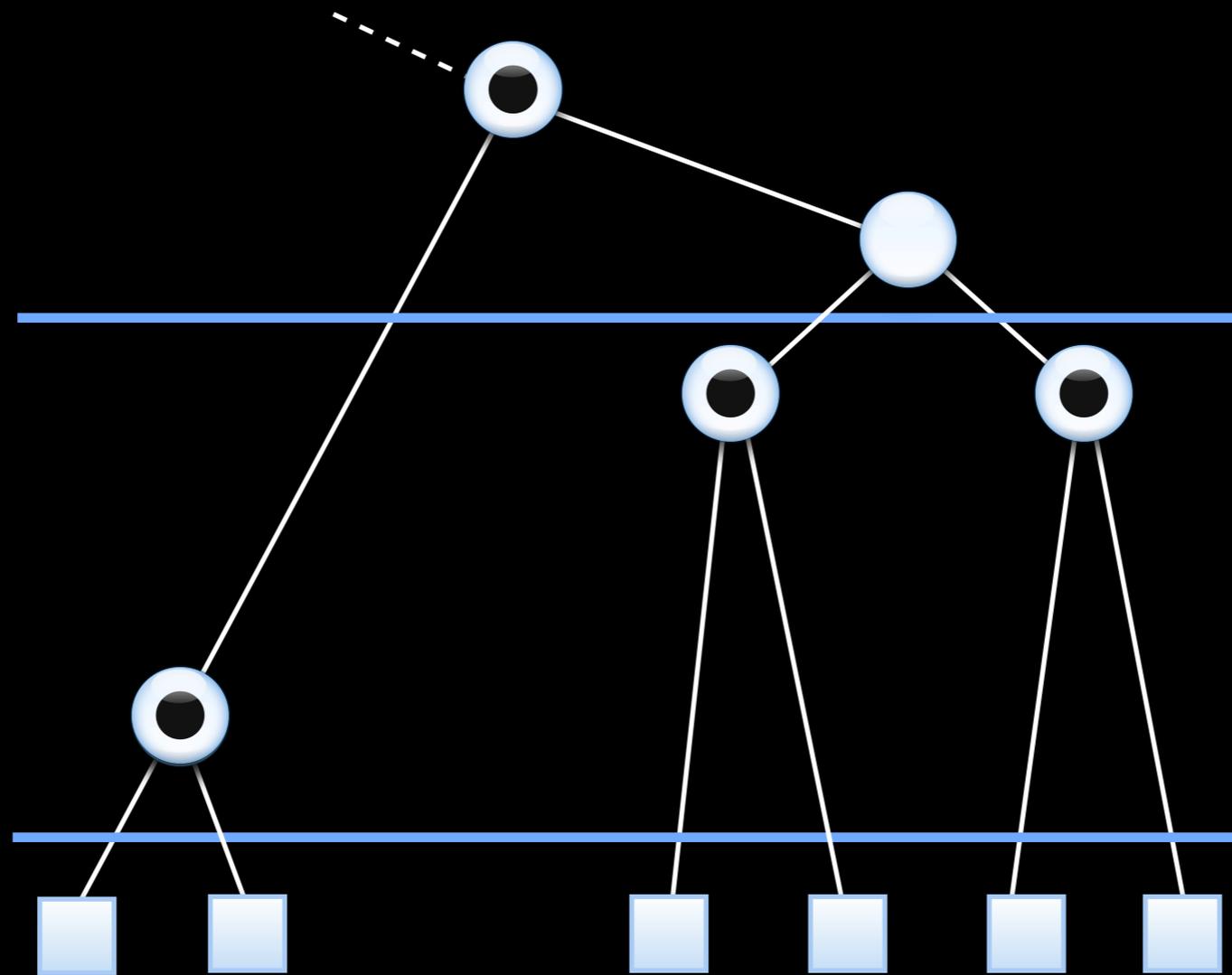
Deletion Case I



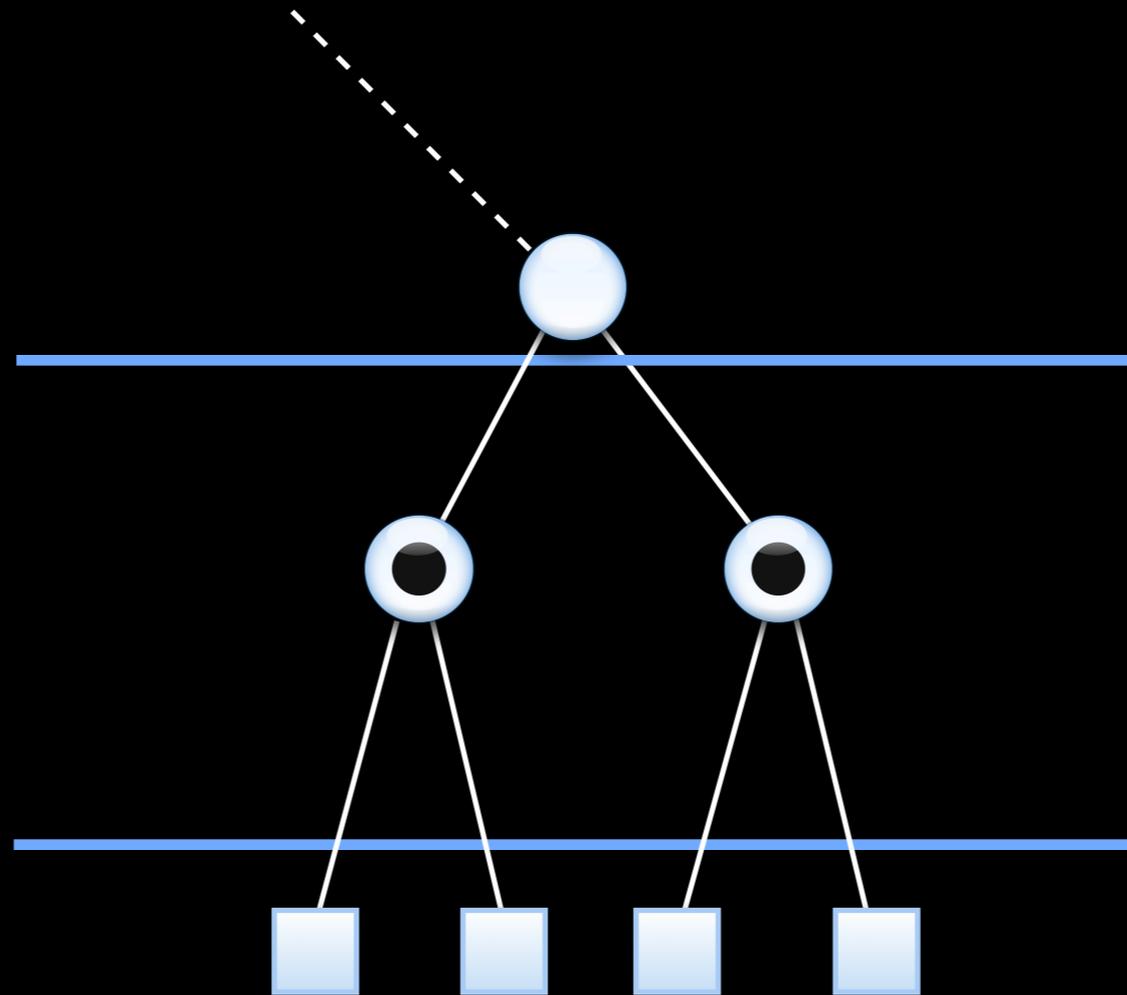
Deletion Case I



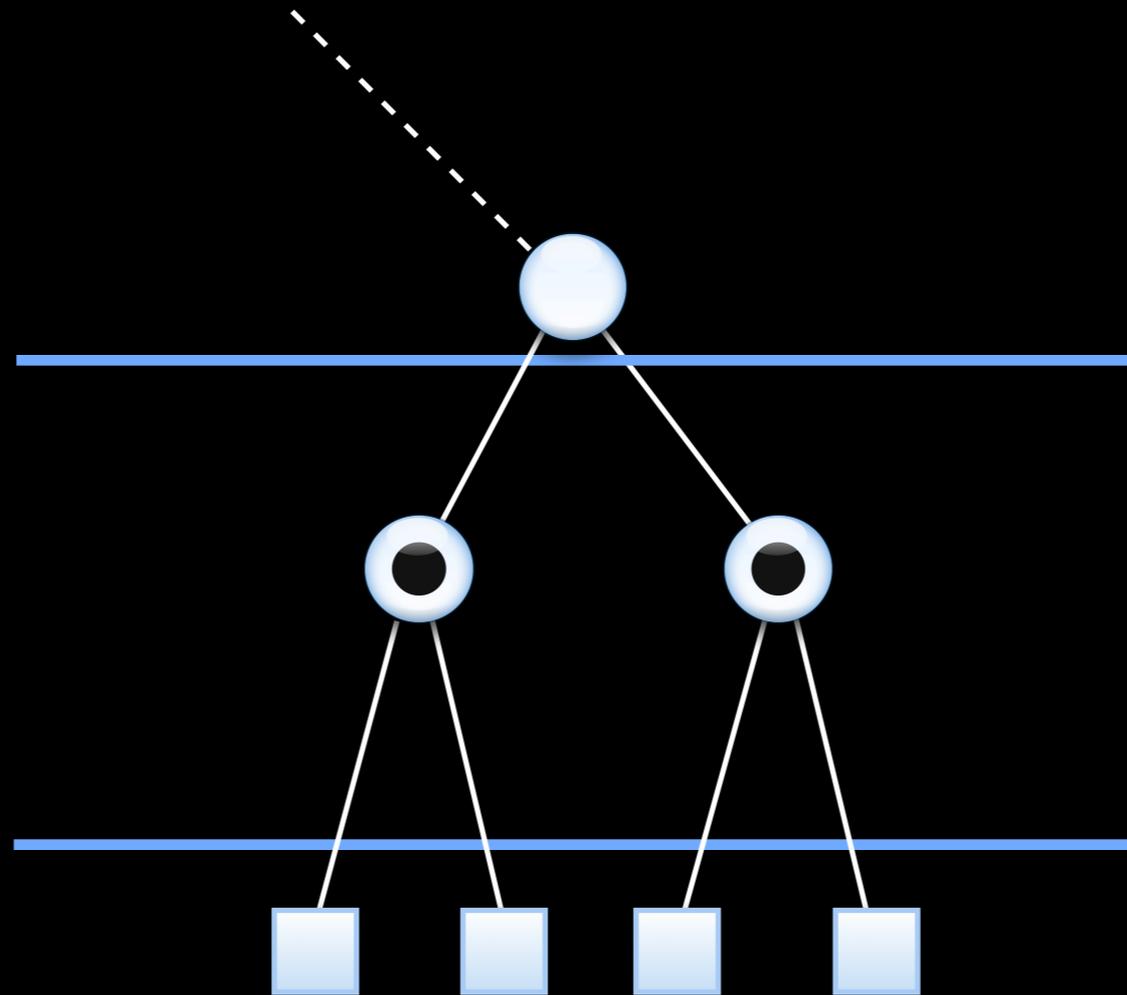
Deletion Case I



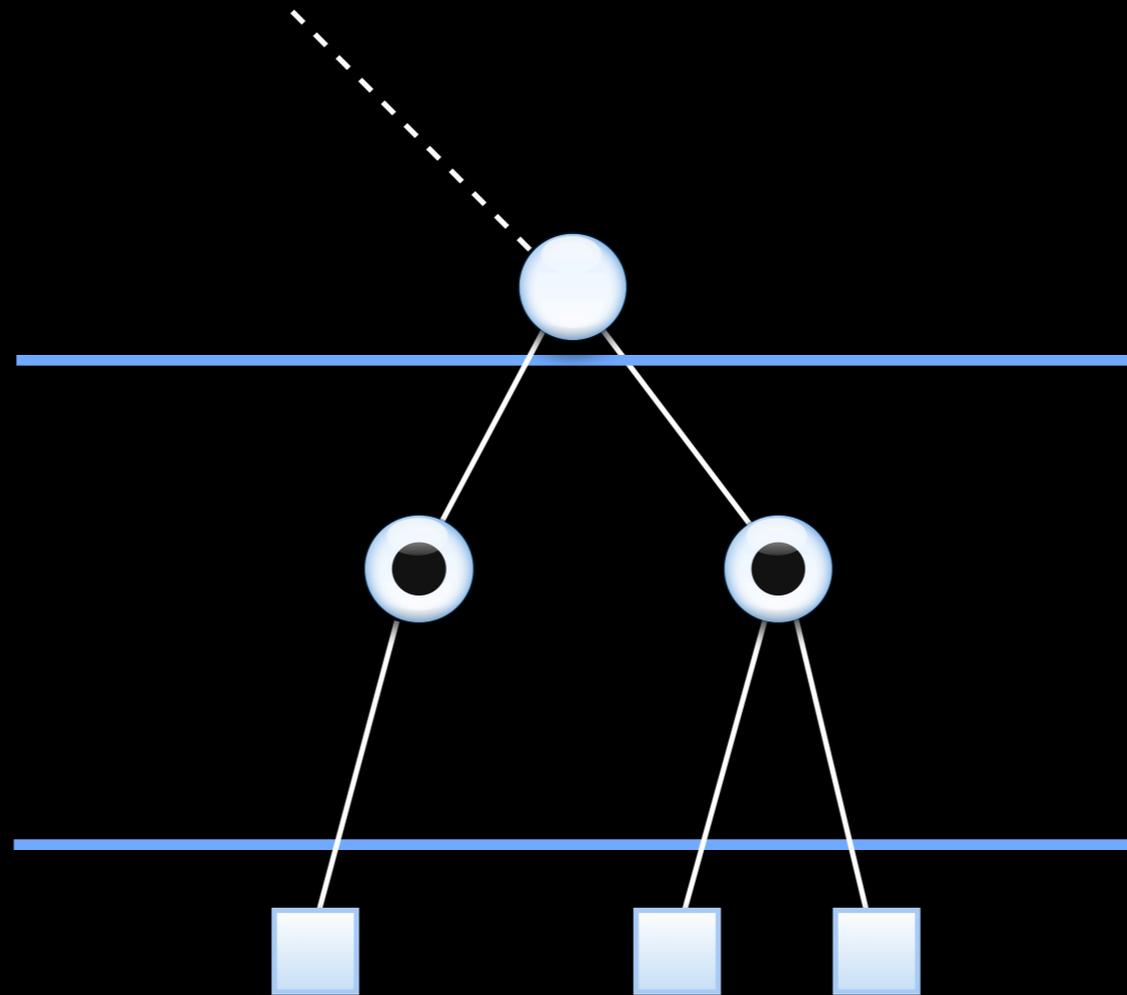
Deletion Case 2



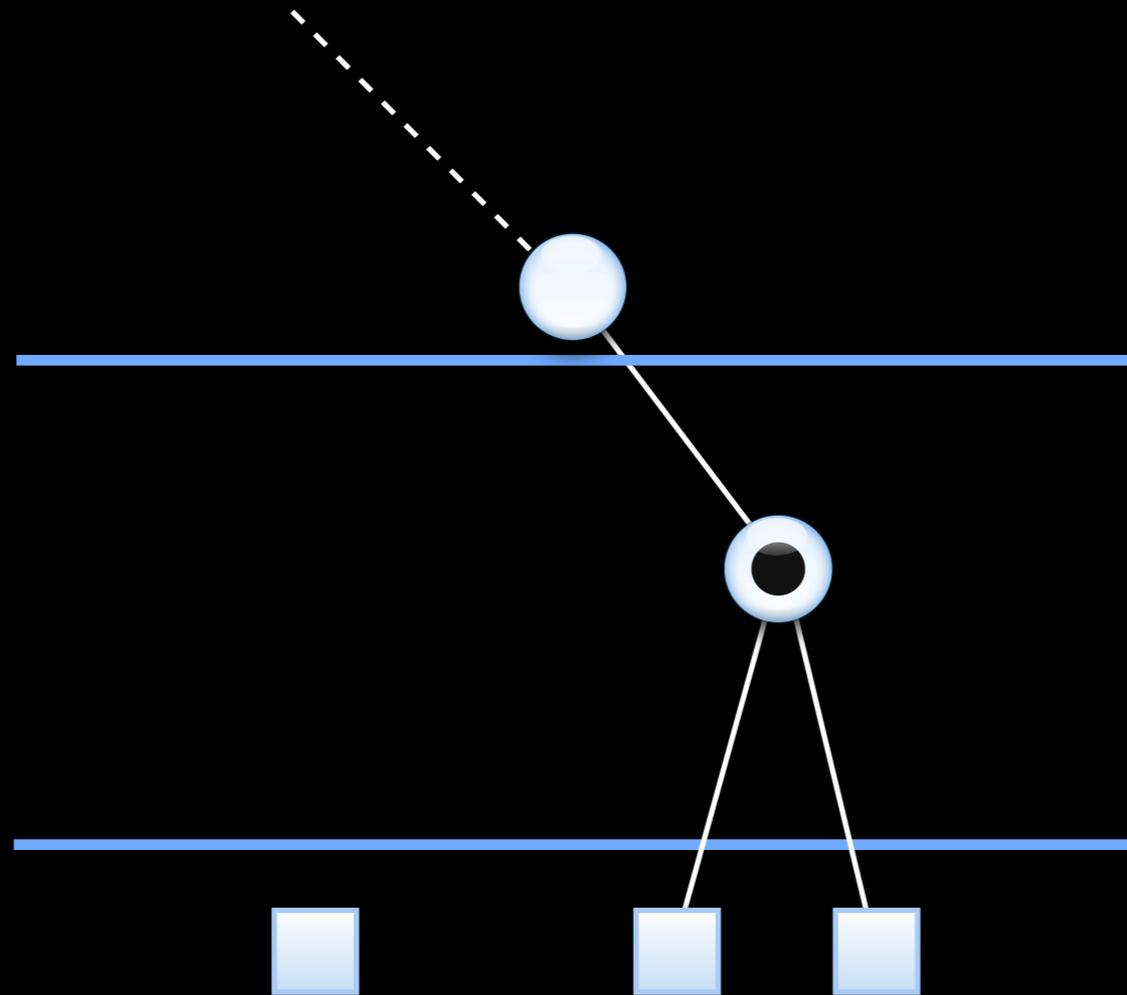
Deletion Case 2



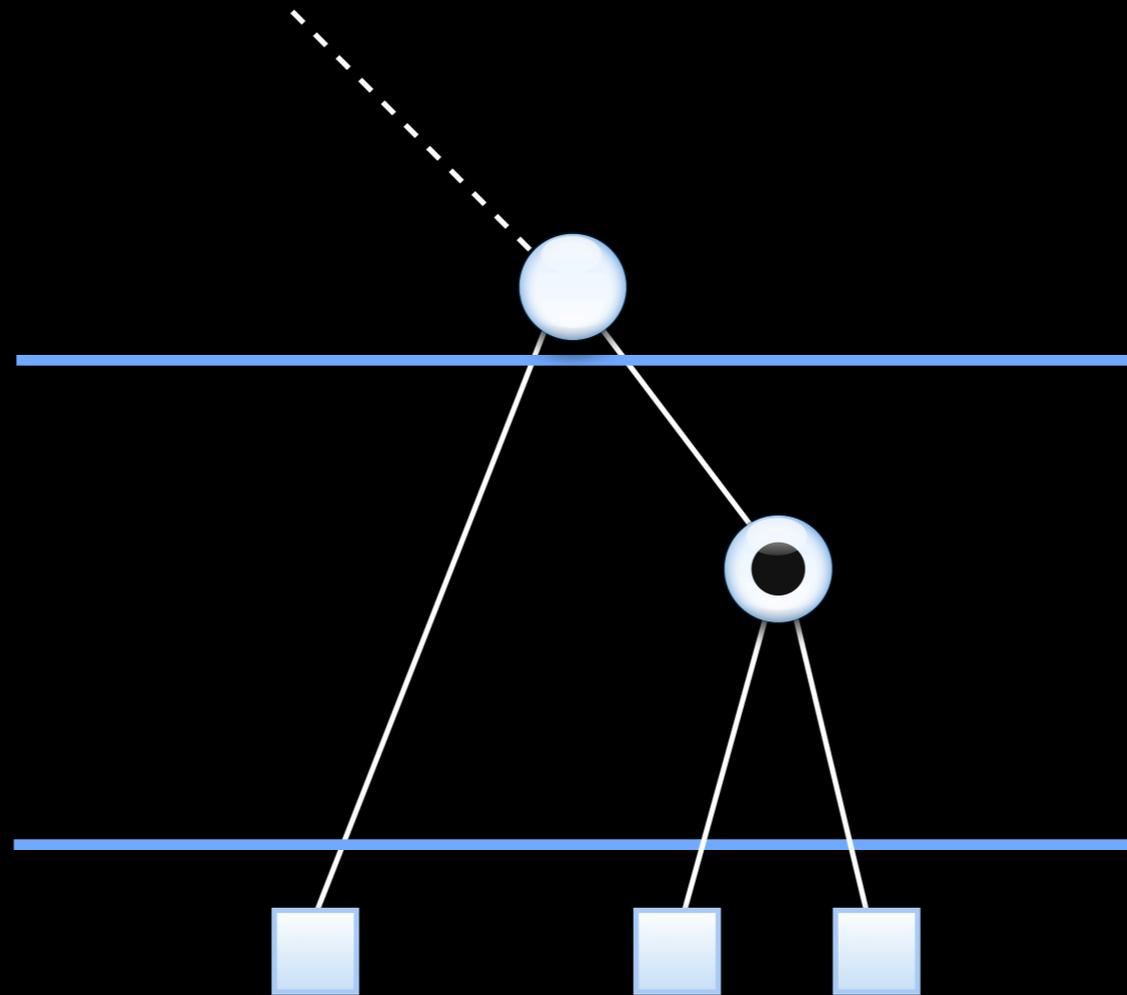
Deletion Case 2



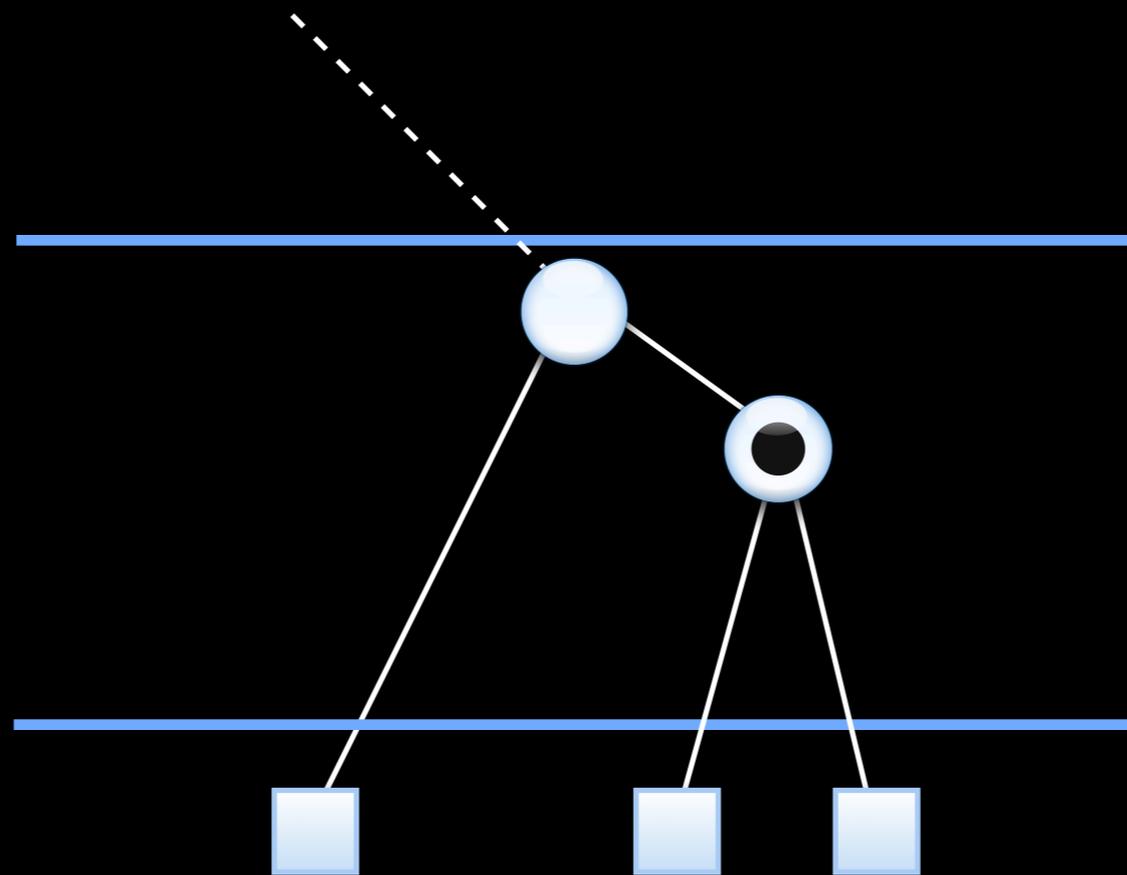
Deletion Case 2



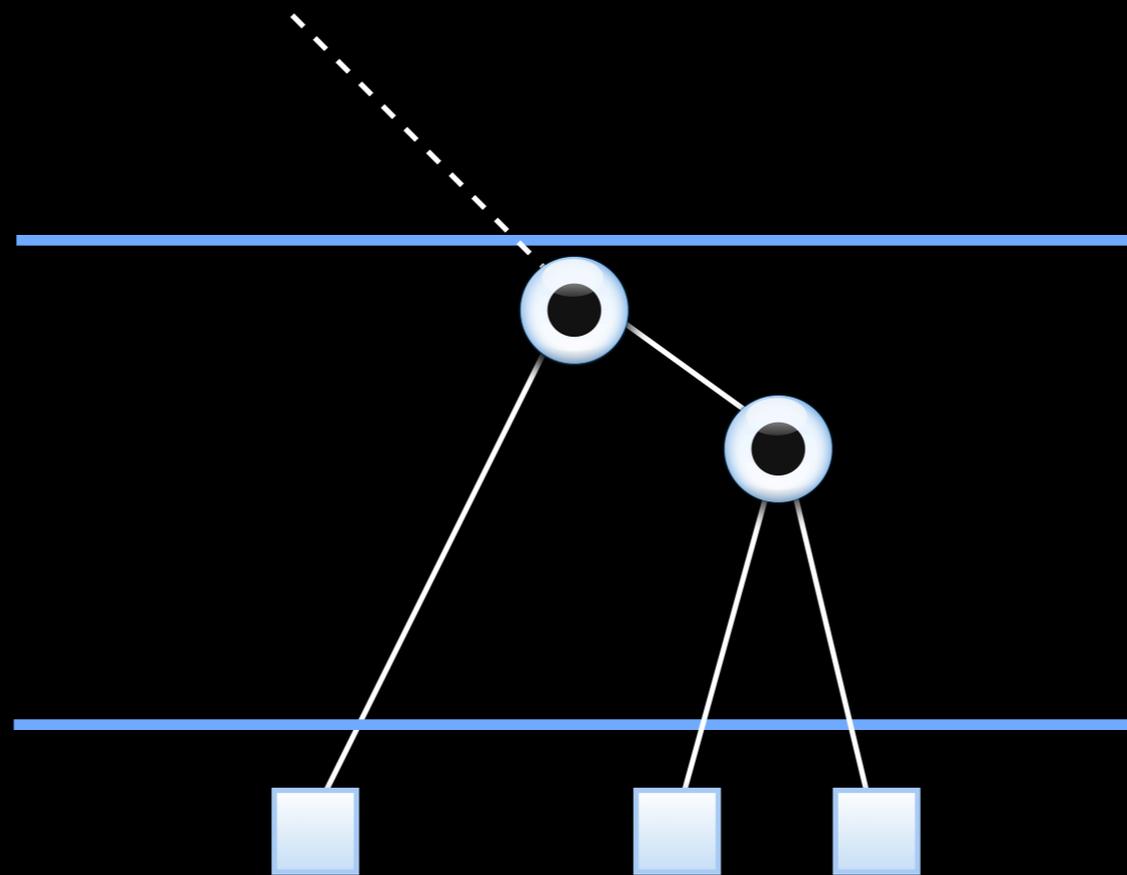
Deletion Case 2



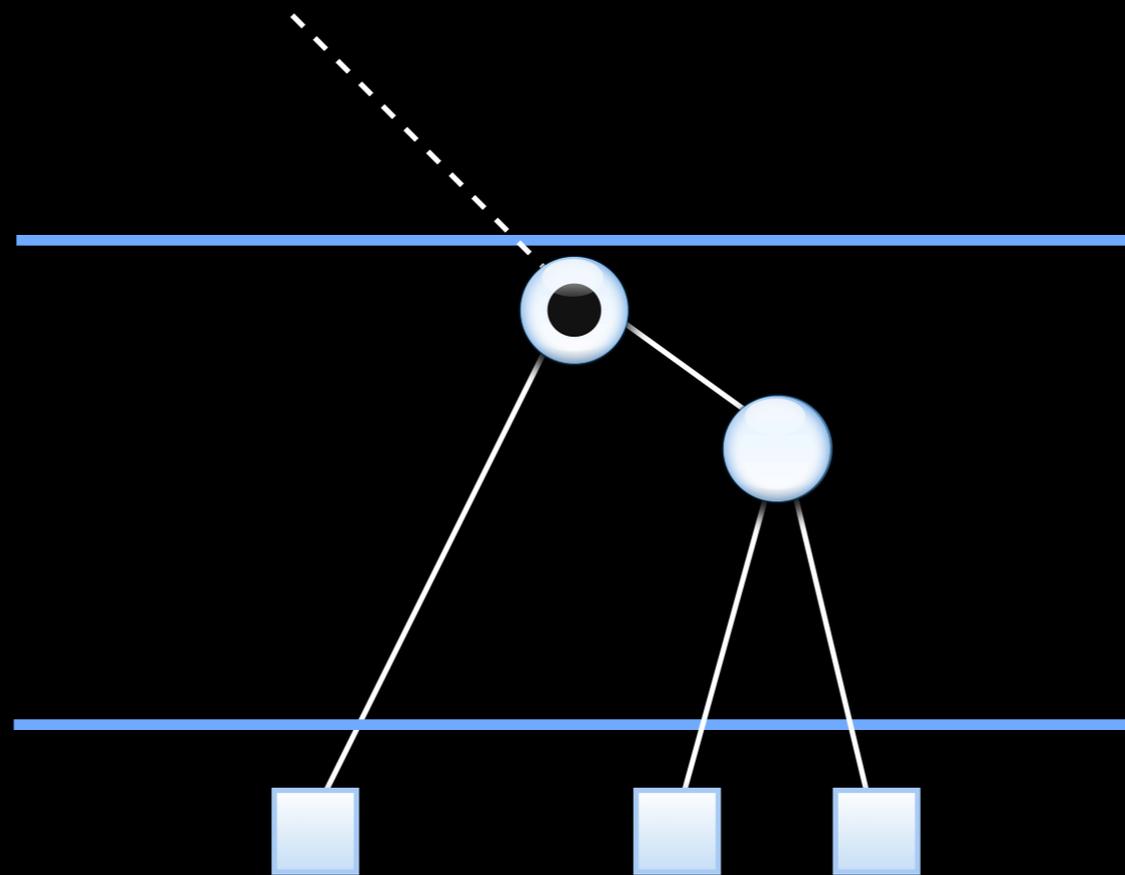
Deletion Case 2



Deletion Case 2



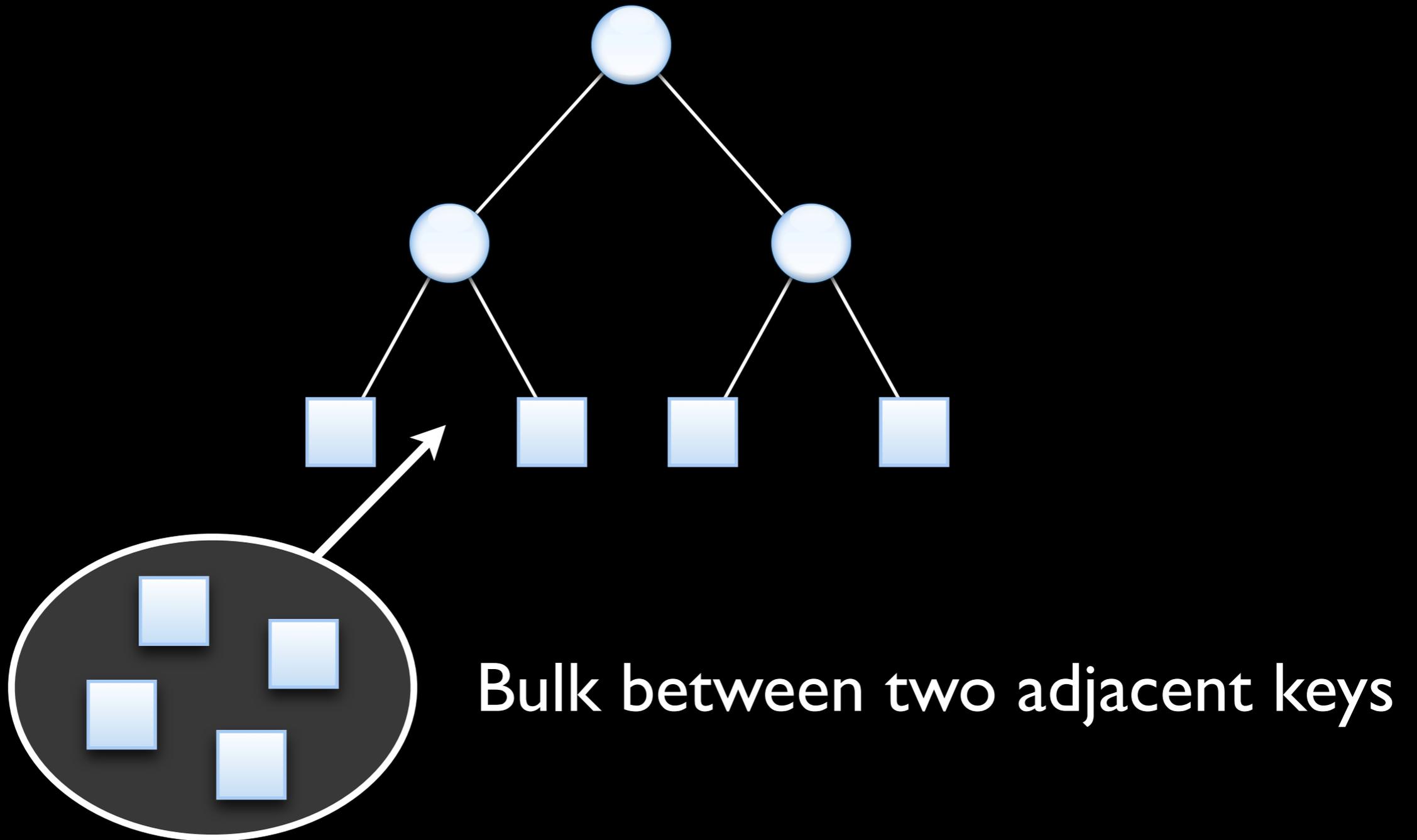
Deletion Case 2



Bulk Insertion

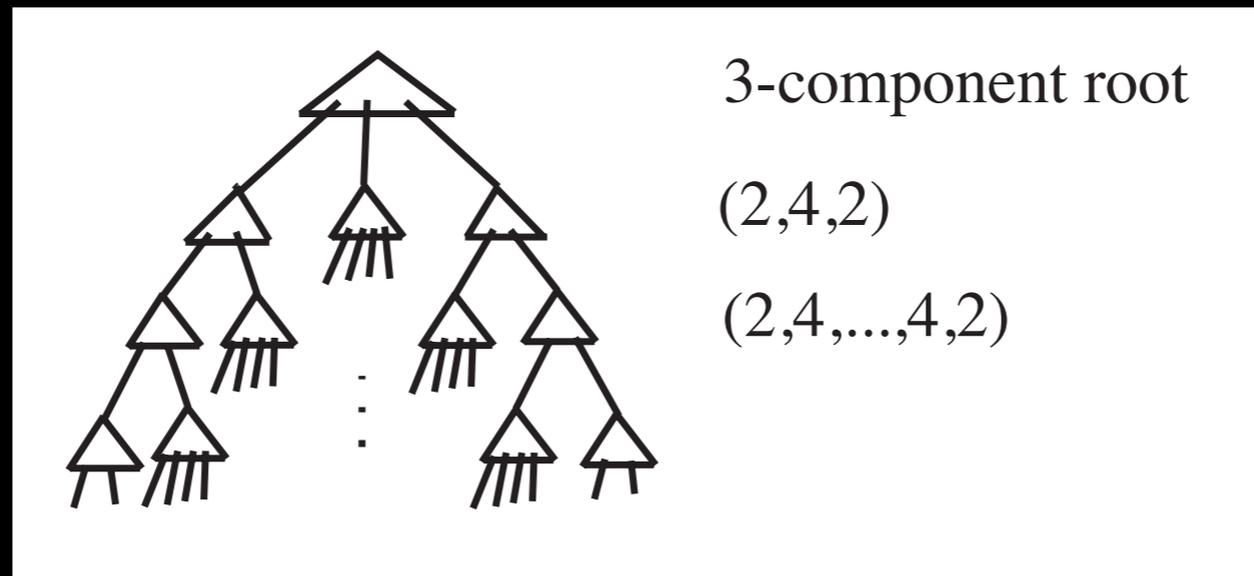
- Insertion of an entire set of keys
- Two phases
 - Bulk tree construction
 - Bulk tree insertion

Bulk



Bulk tree construction

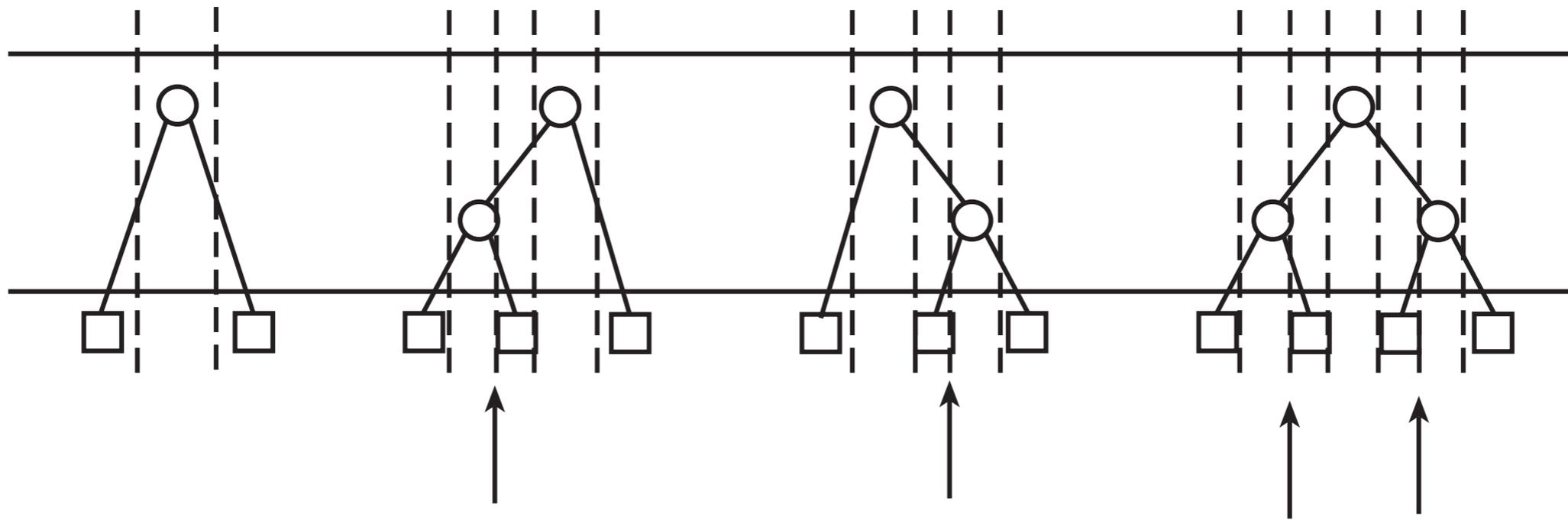
- Constructed according to predefined rules
- Shape chosen to ease insertion



Bulk tree insertion

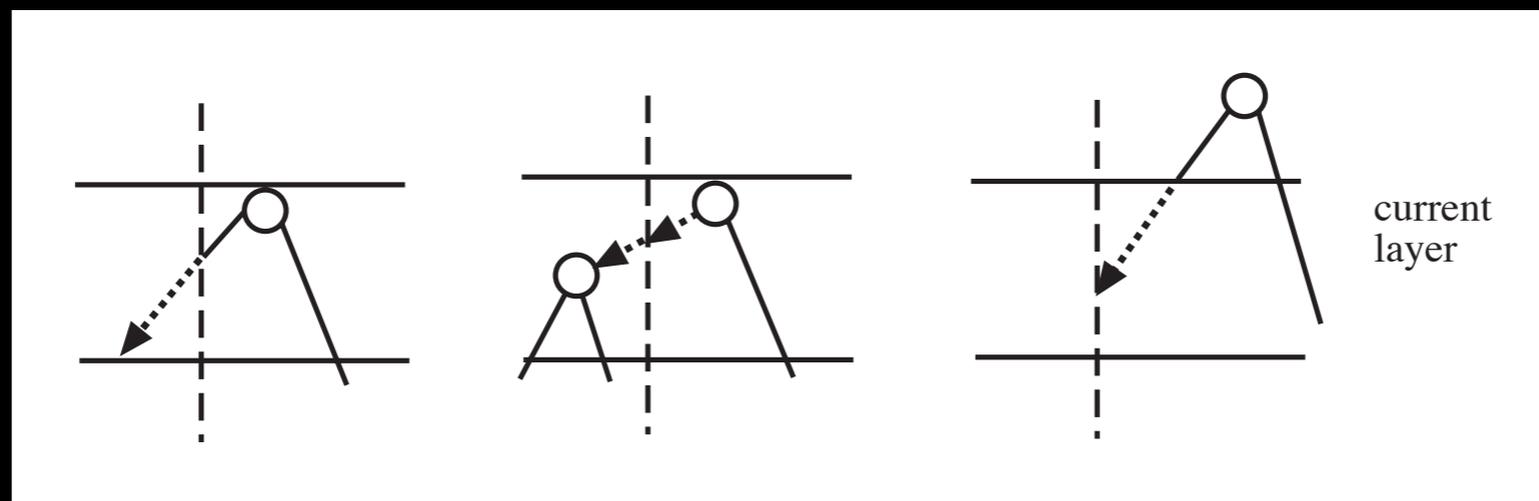
- Start at the bottom, i.e.. at the leaves
- Processes one layer at a time
- Links may be cut and new ones created
- The root layer is treated differently

Cut positions

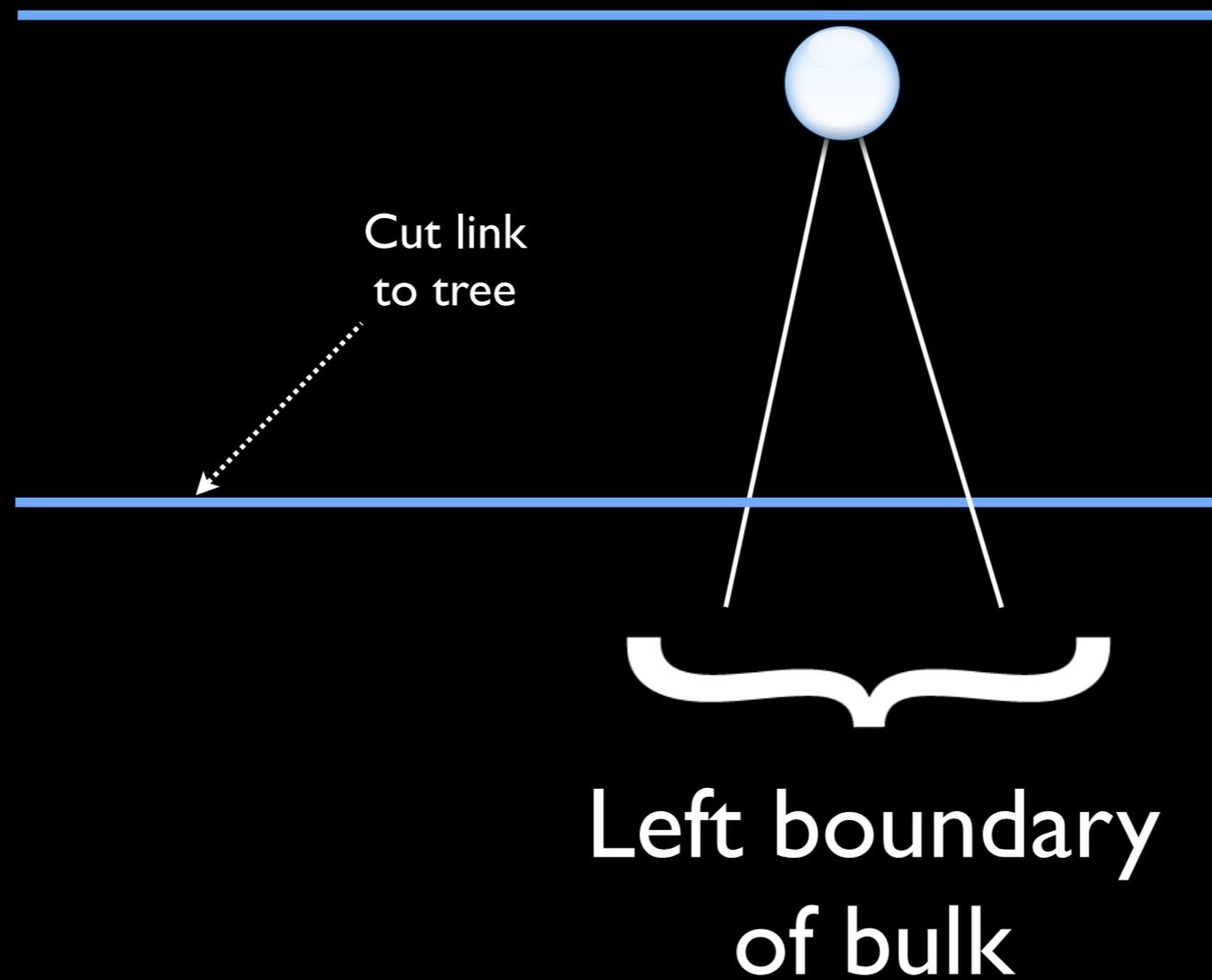


Cut links

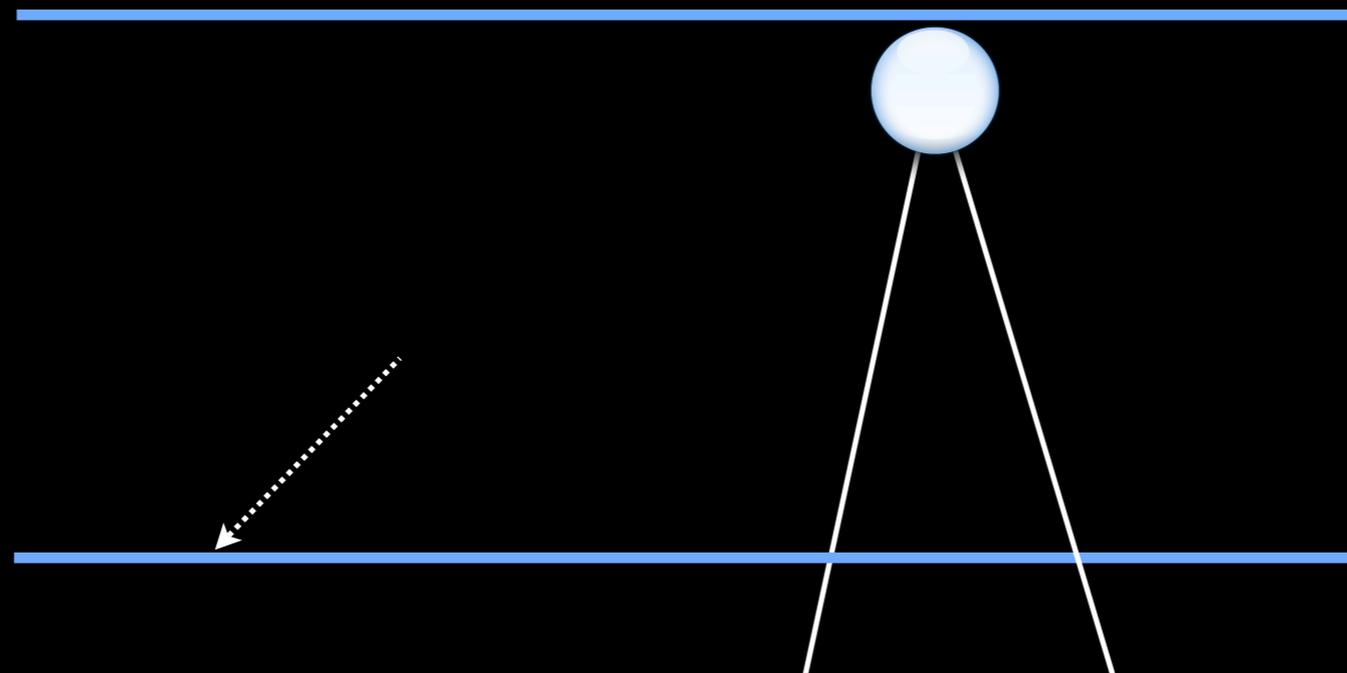
- A cut creates at least two dangling links
- Two types of dangling links
 - Low Half Links
 - High Half Links



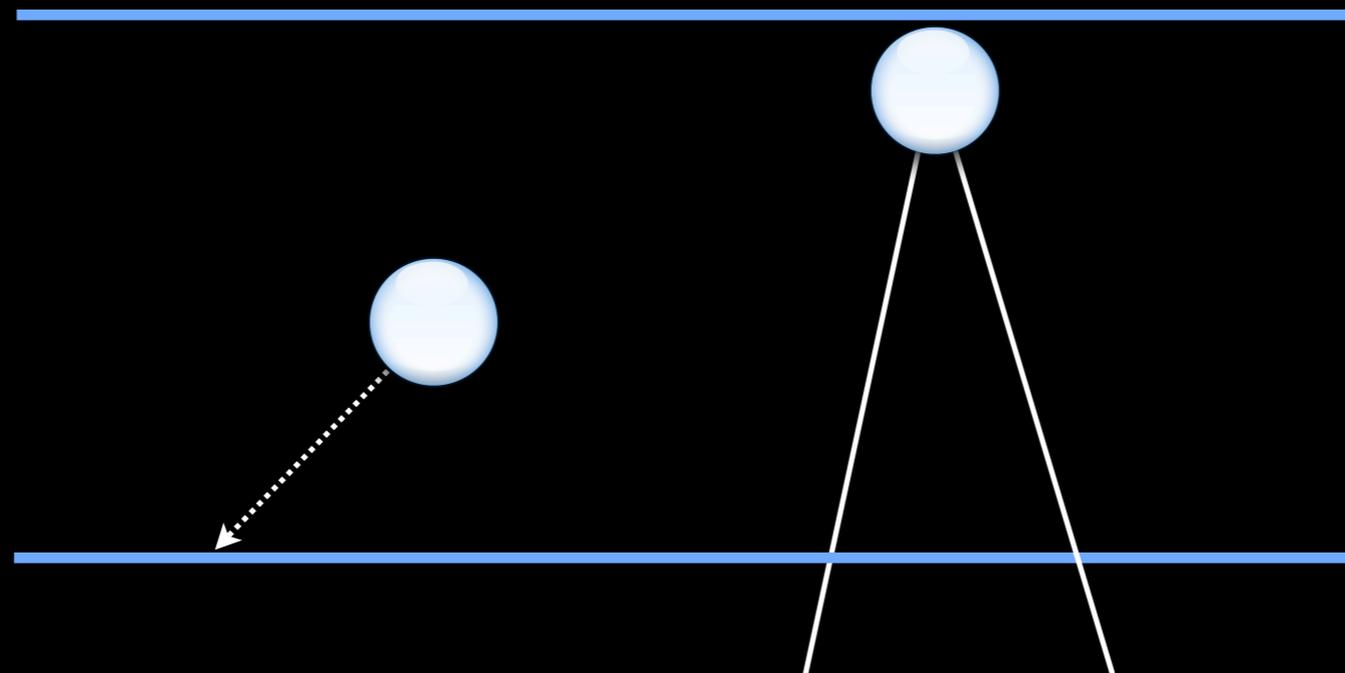
Low Half Links Case I



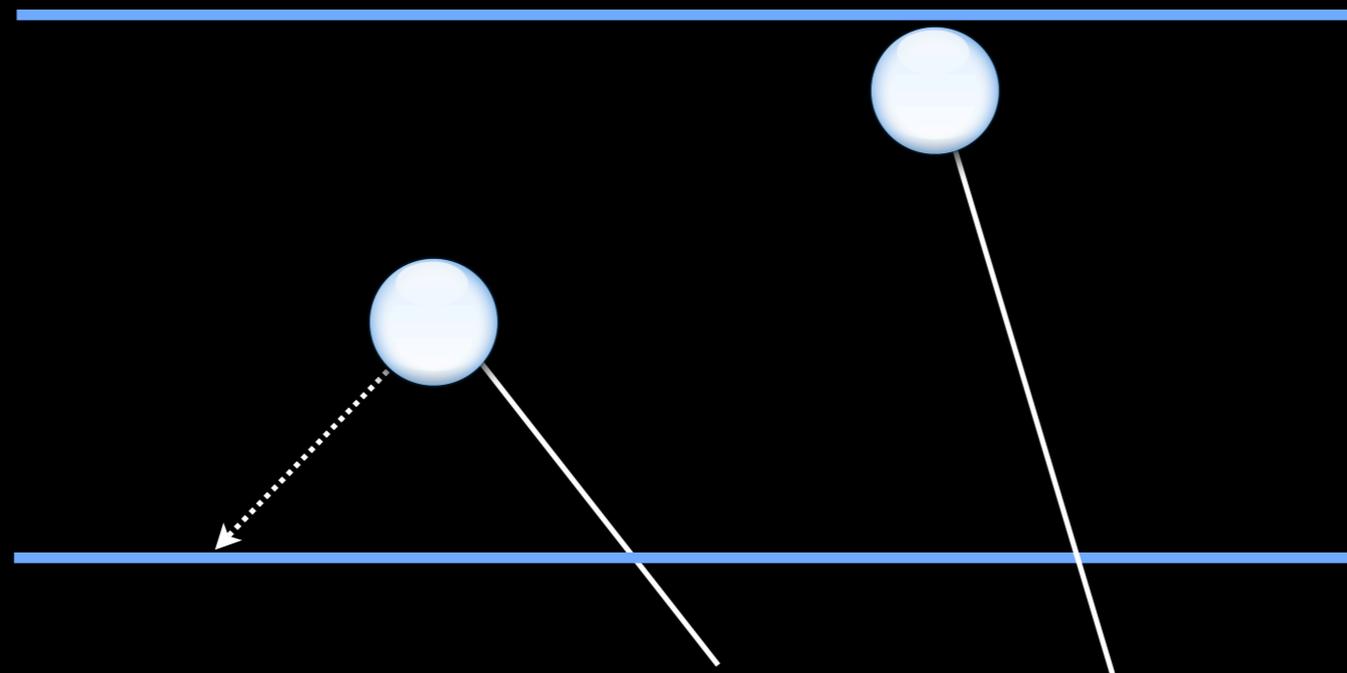
Low Half Links Case I



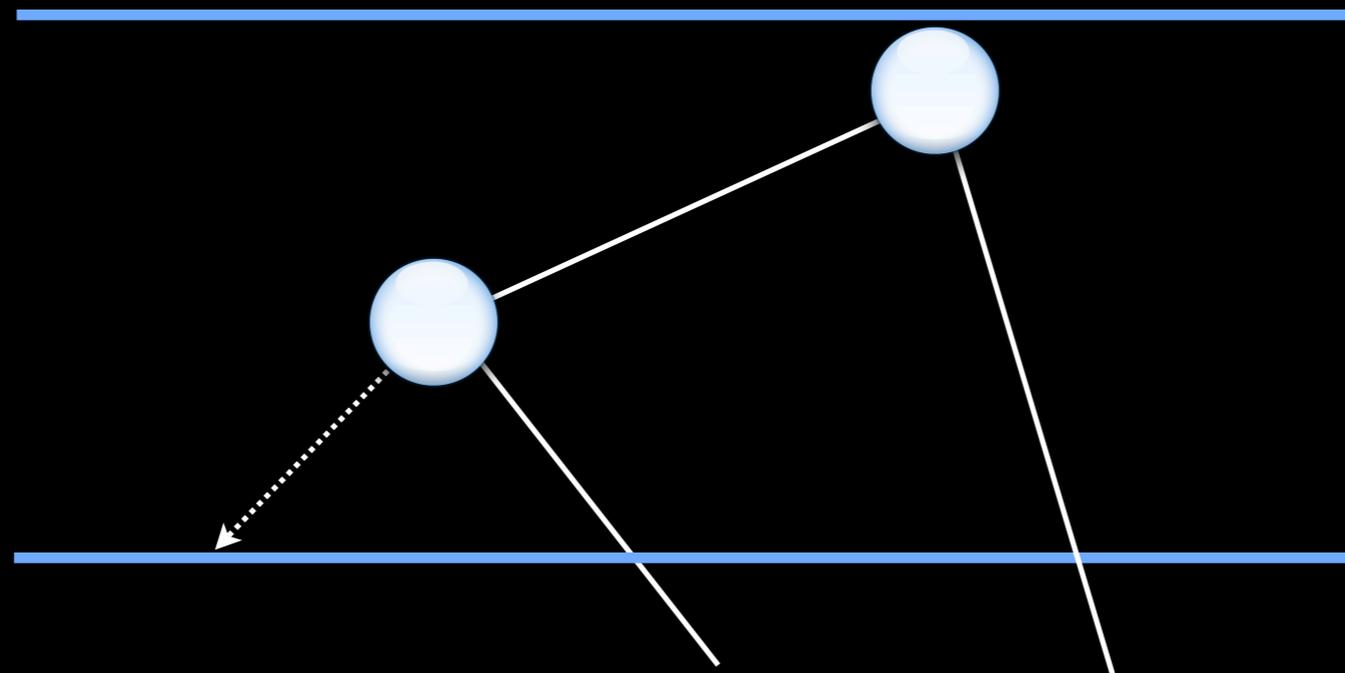
Low Half Links Case I



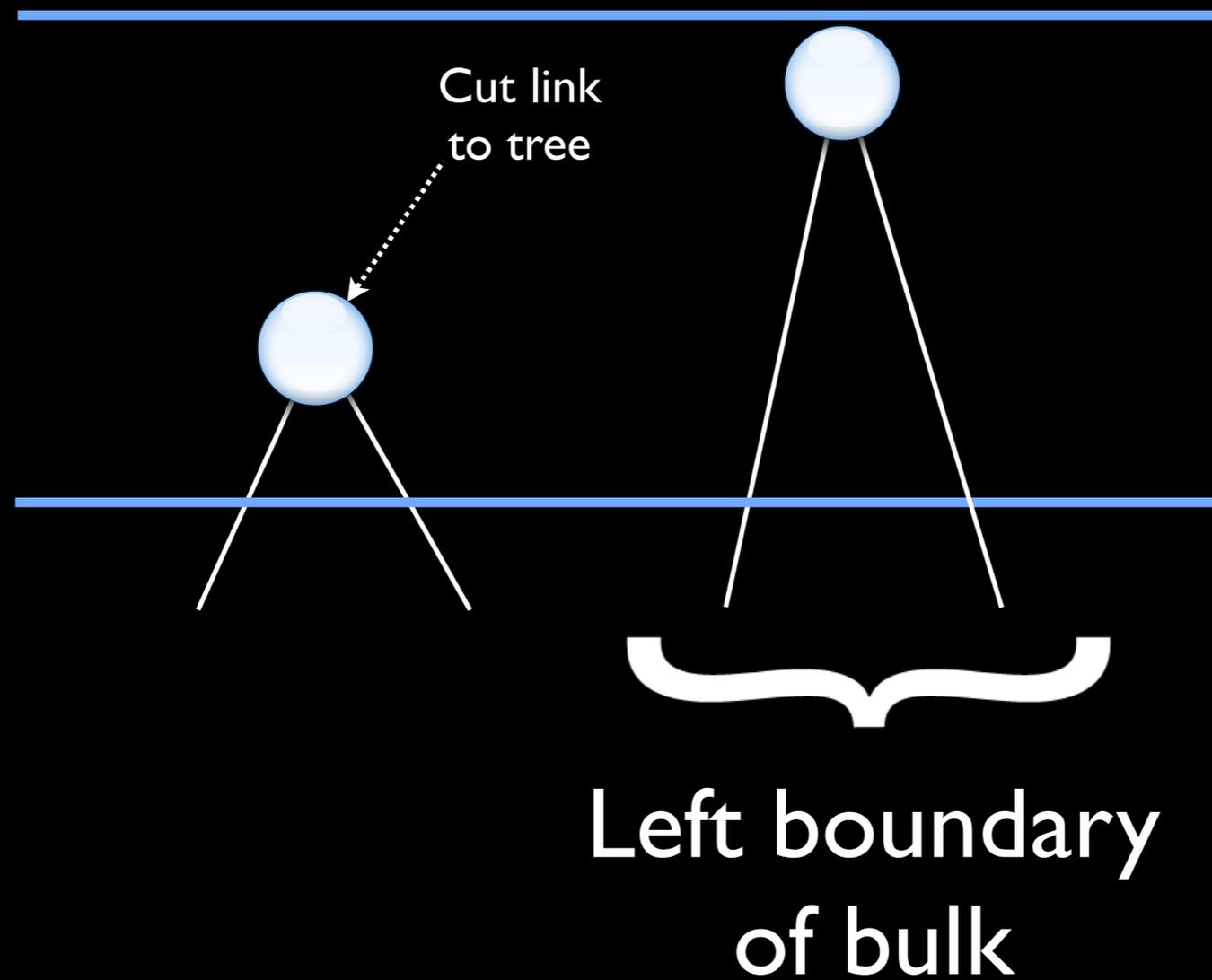
Low Half Links Case I



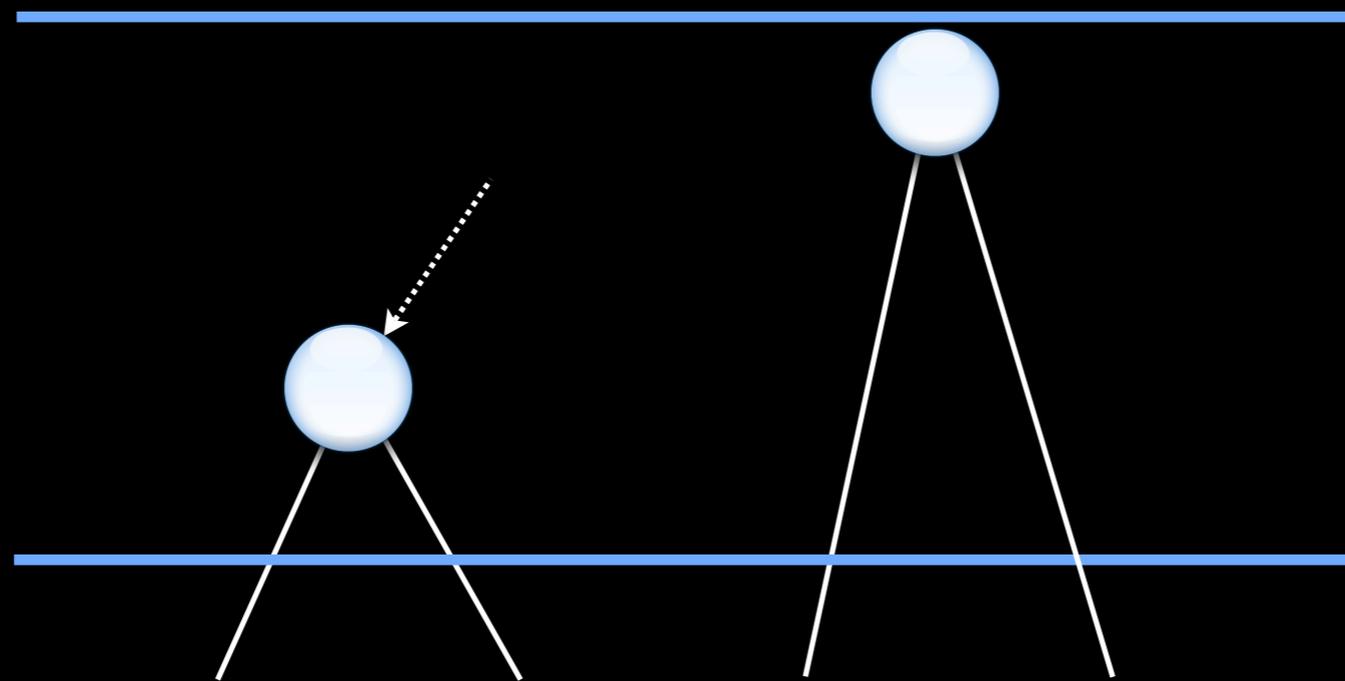
Low Half Links Case I



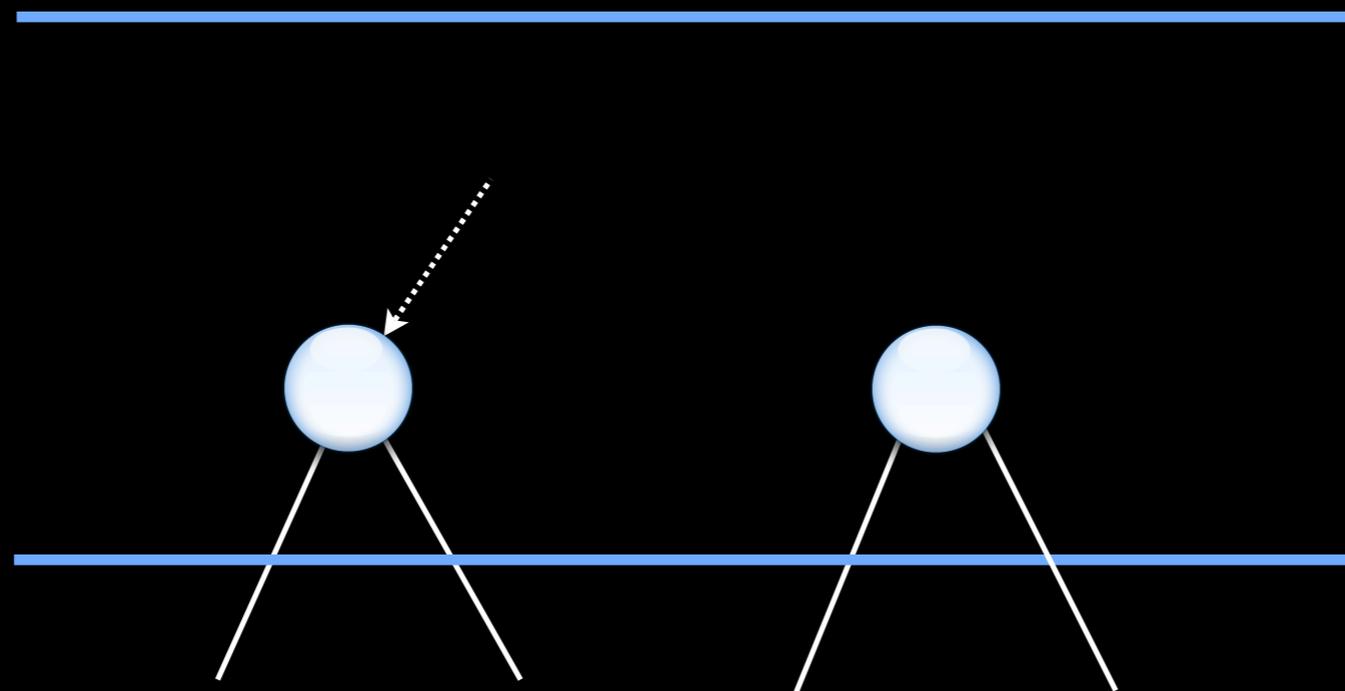
Low Half Links Case 2



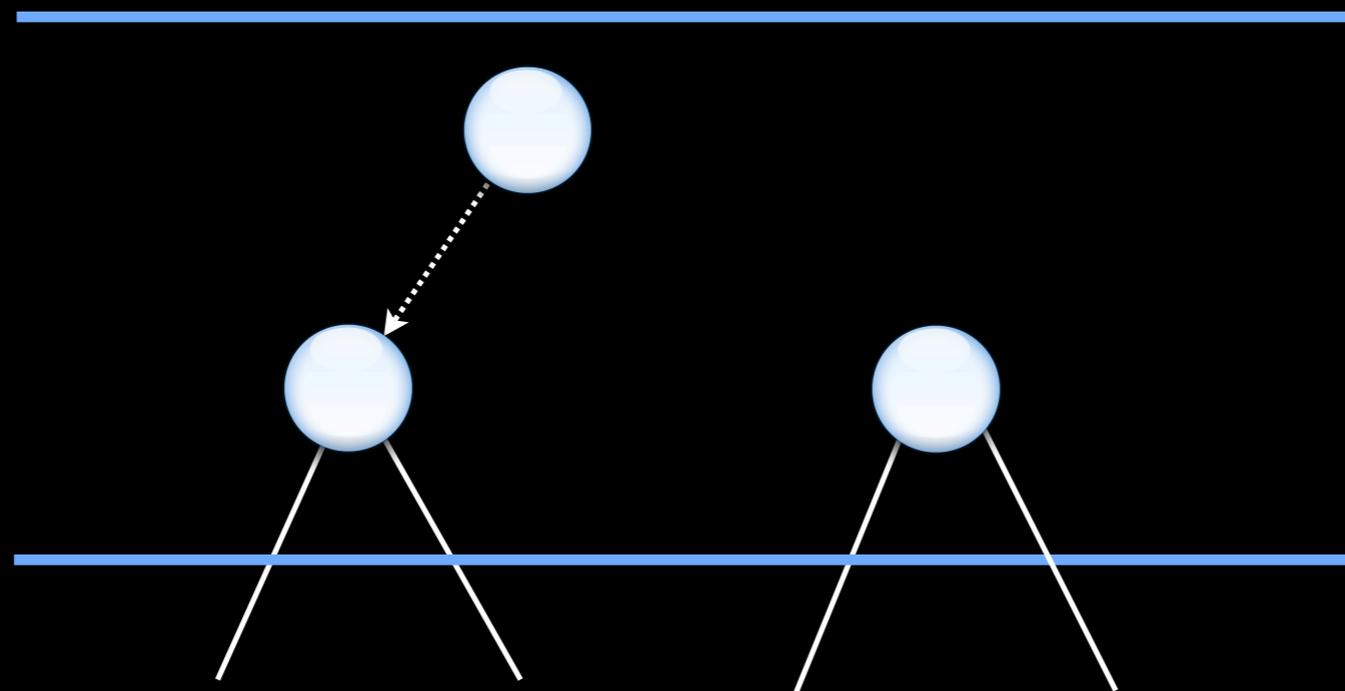
Low Half Links Case 2



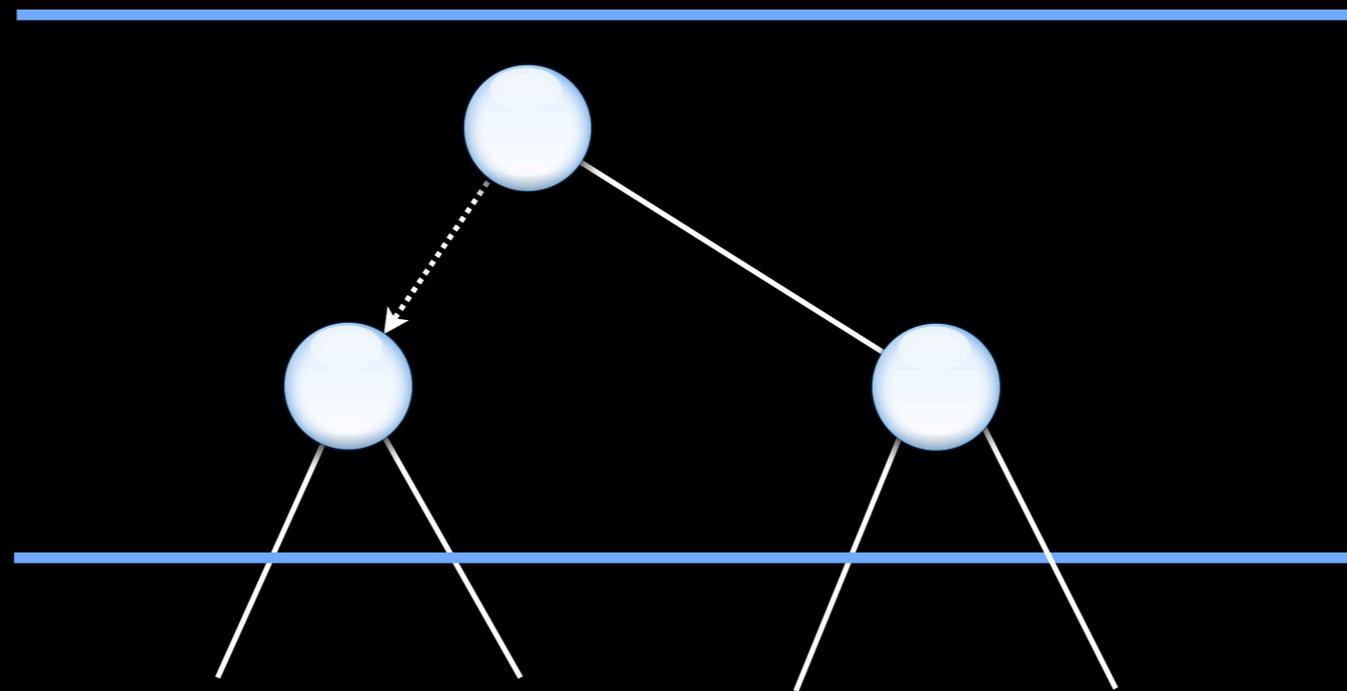
Low Half Links Case 2



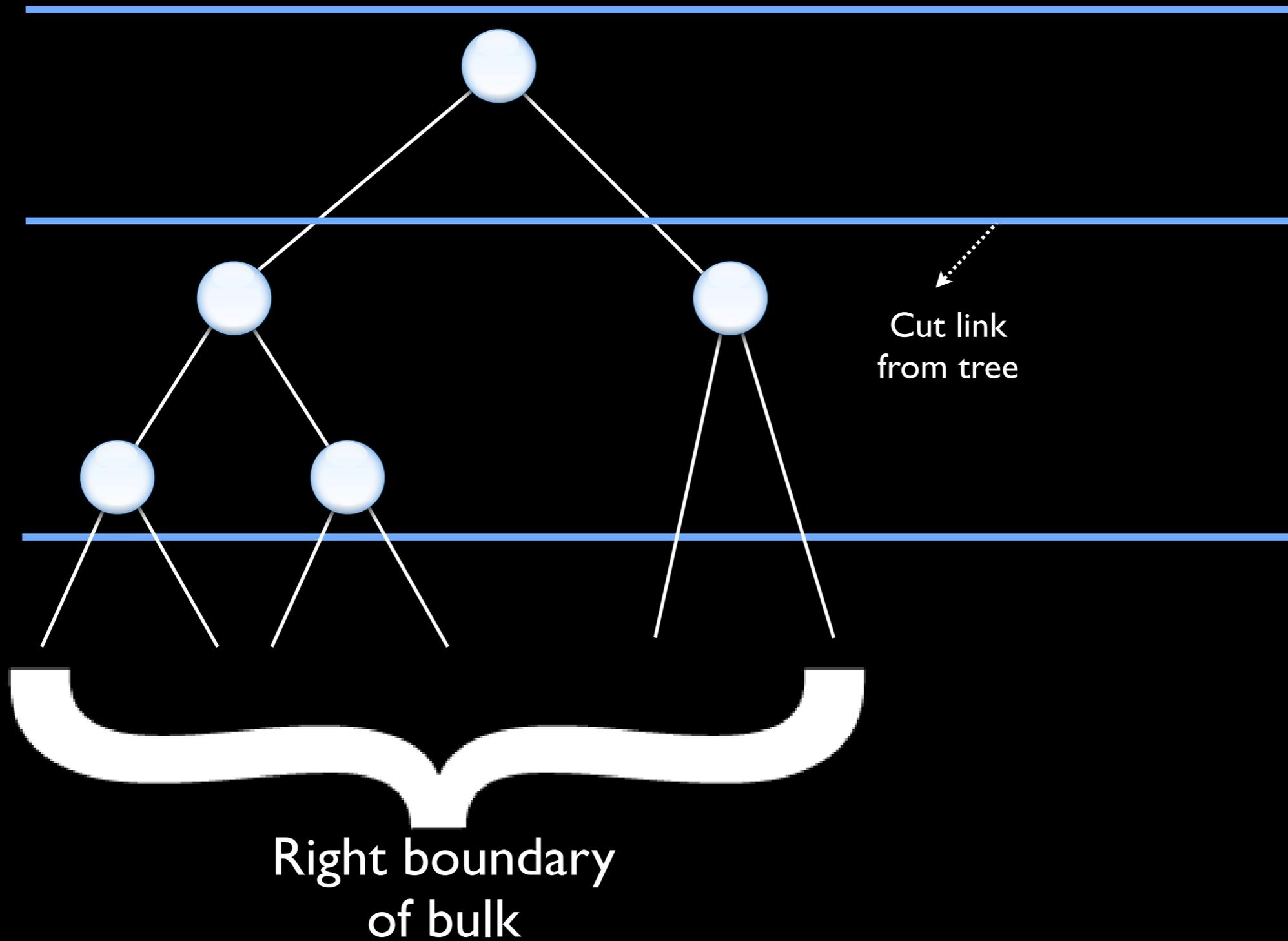
Low Half Links Case 2



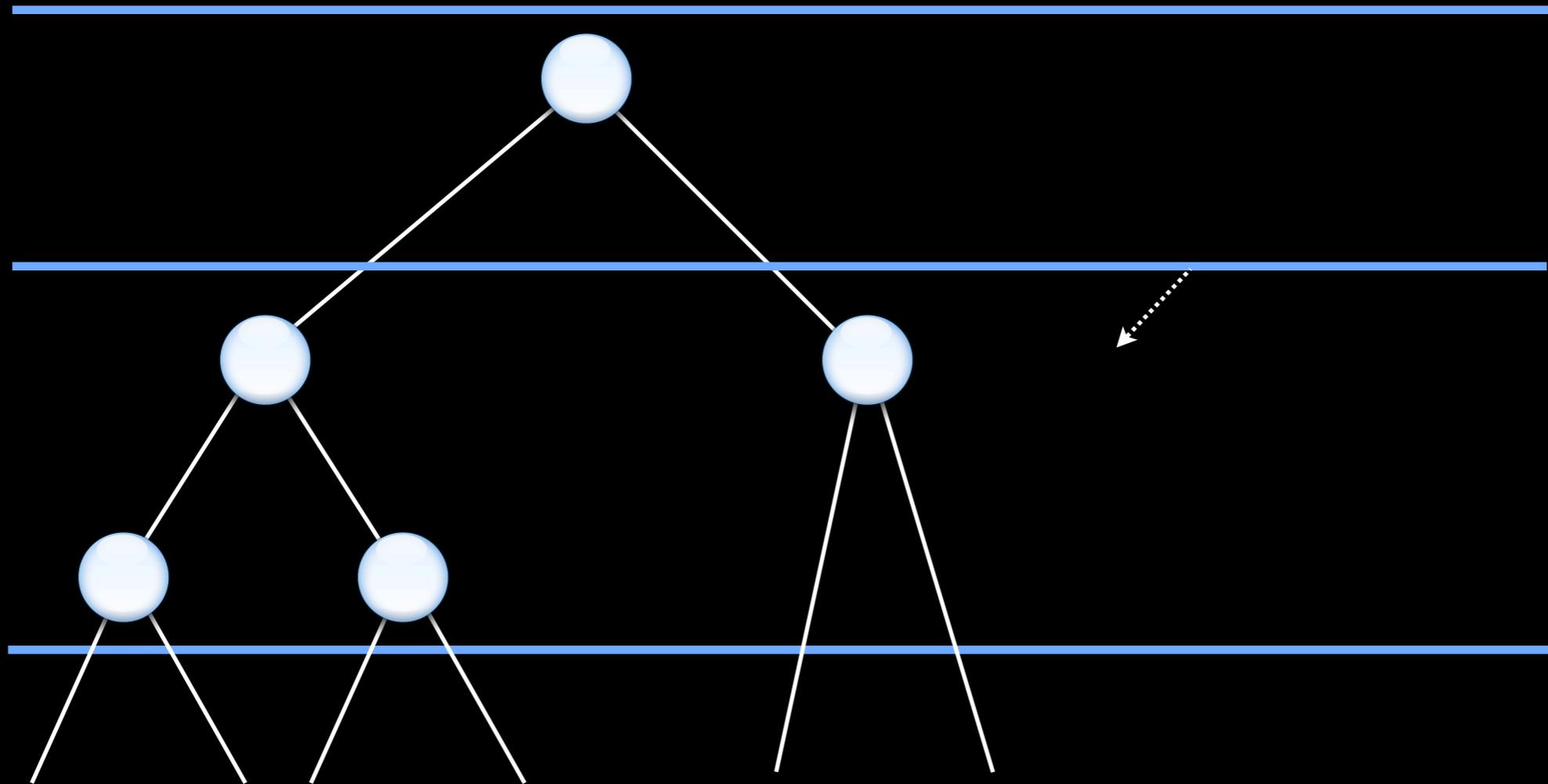
Low Half Links Case 2



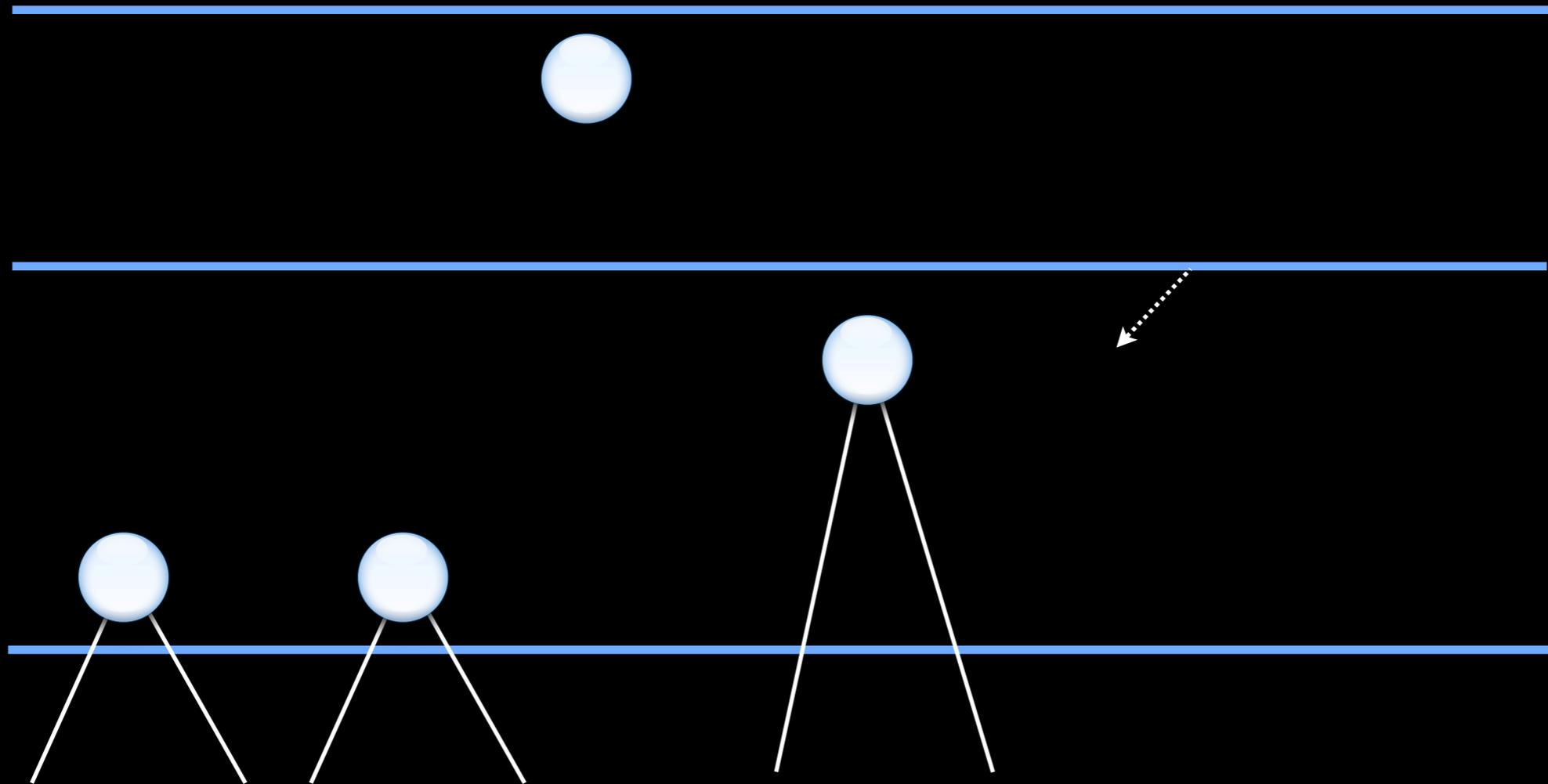
High Half Links Case I



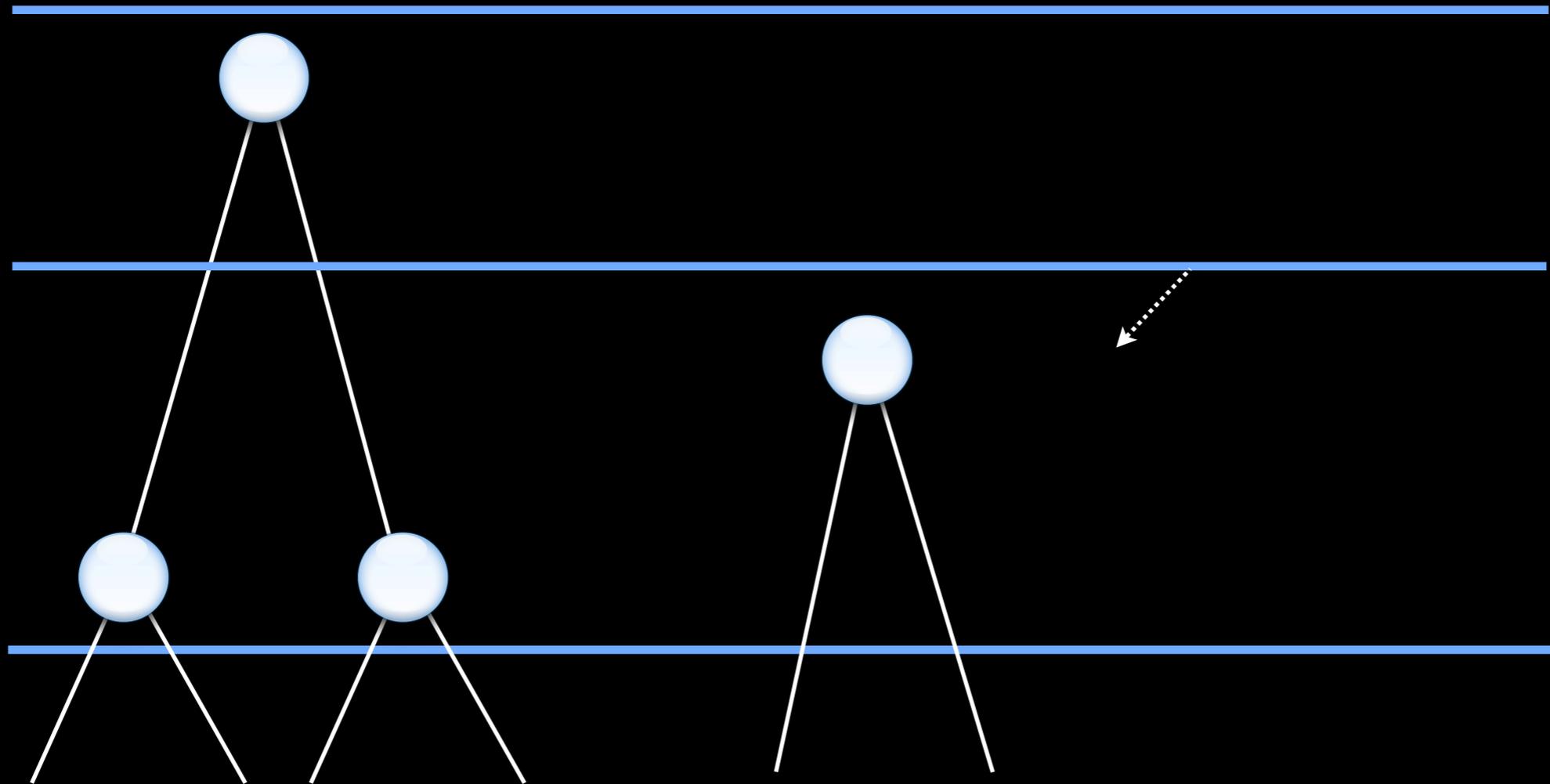
High Half Links Case I



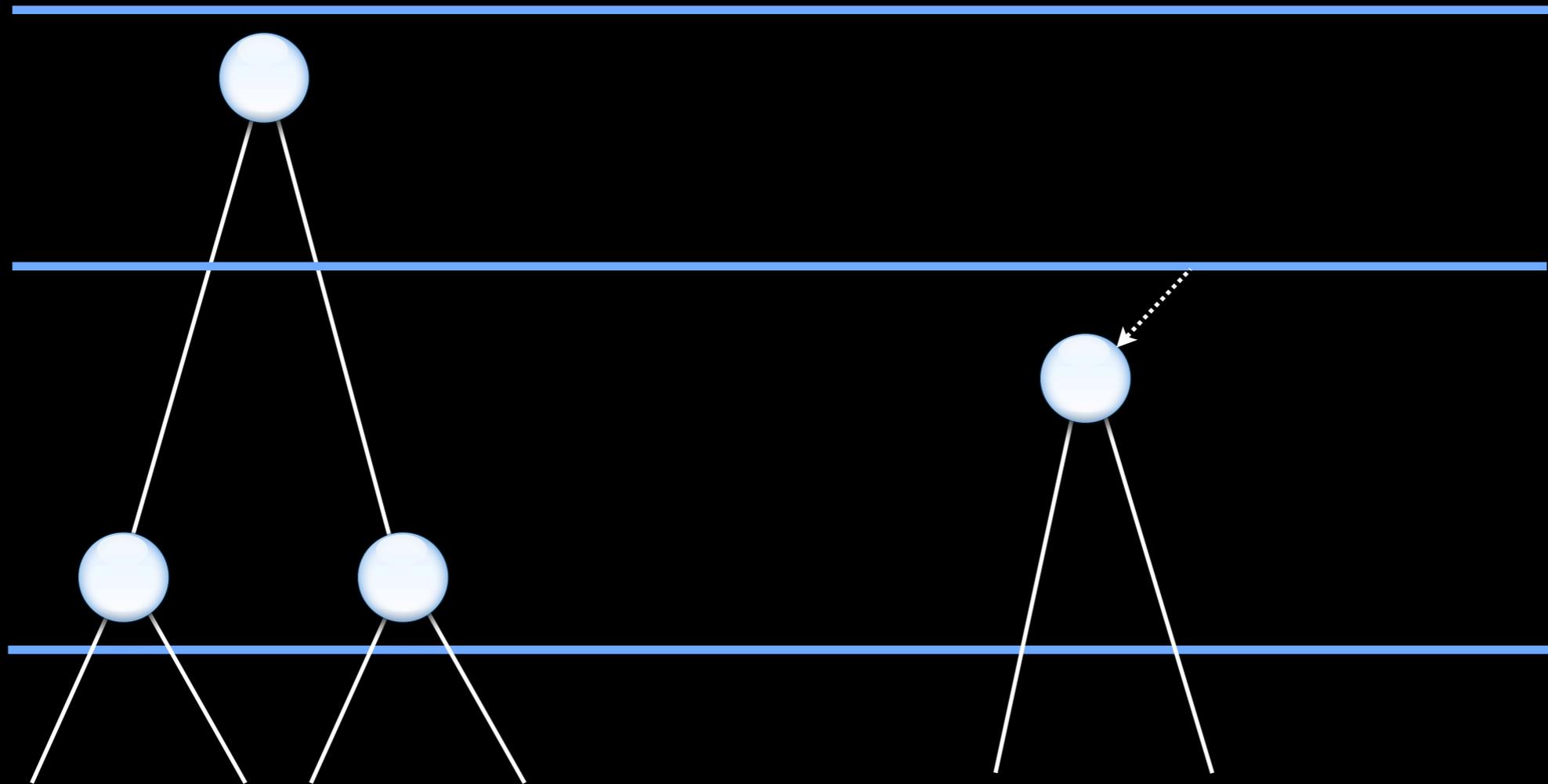
High Half Links Case I



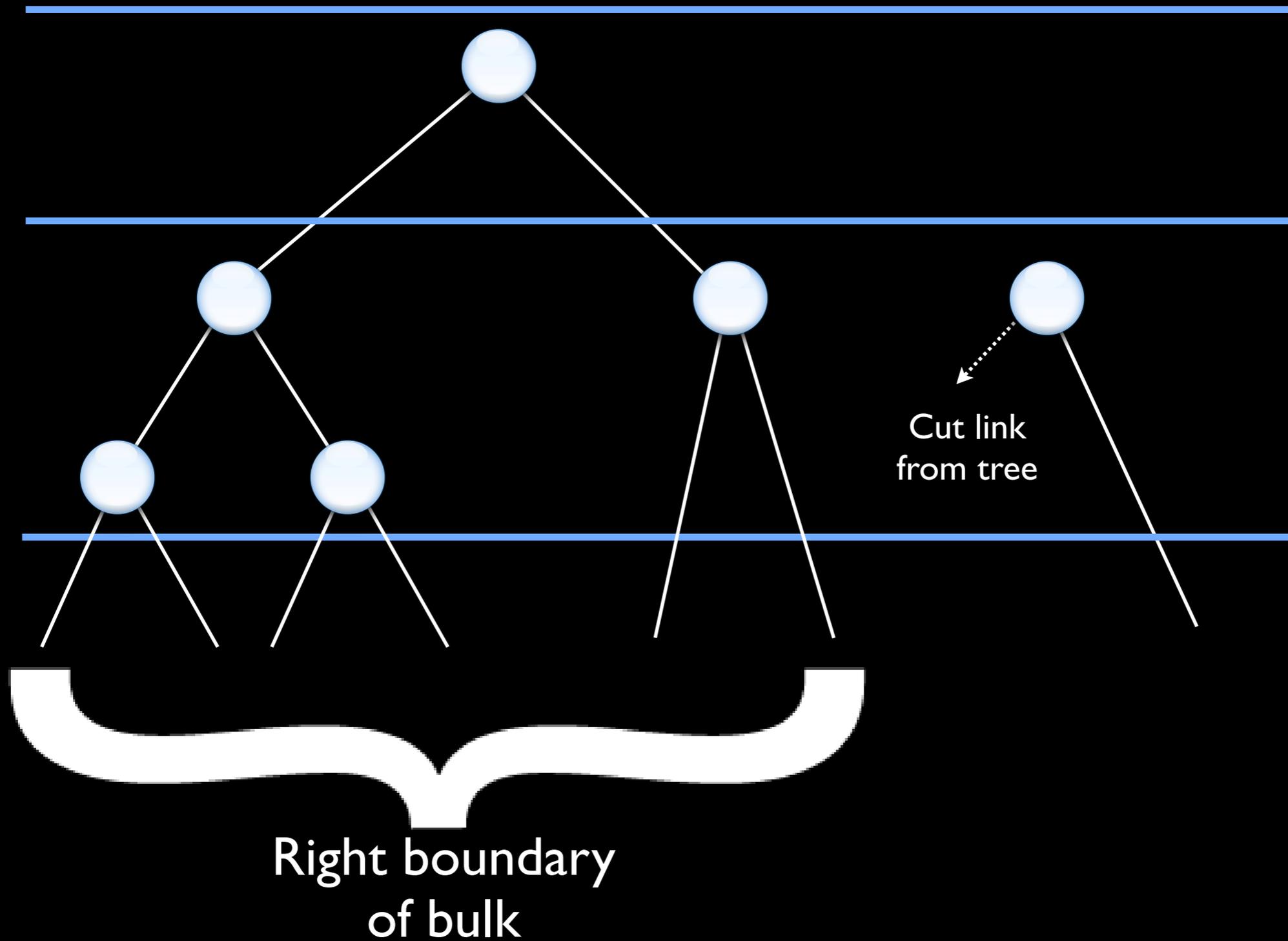
High Half Links Case I



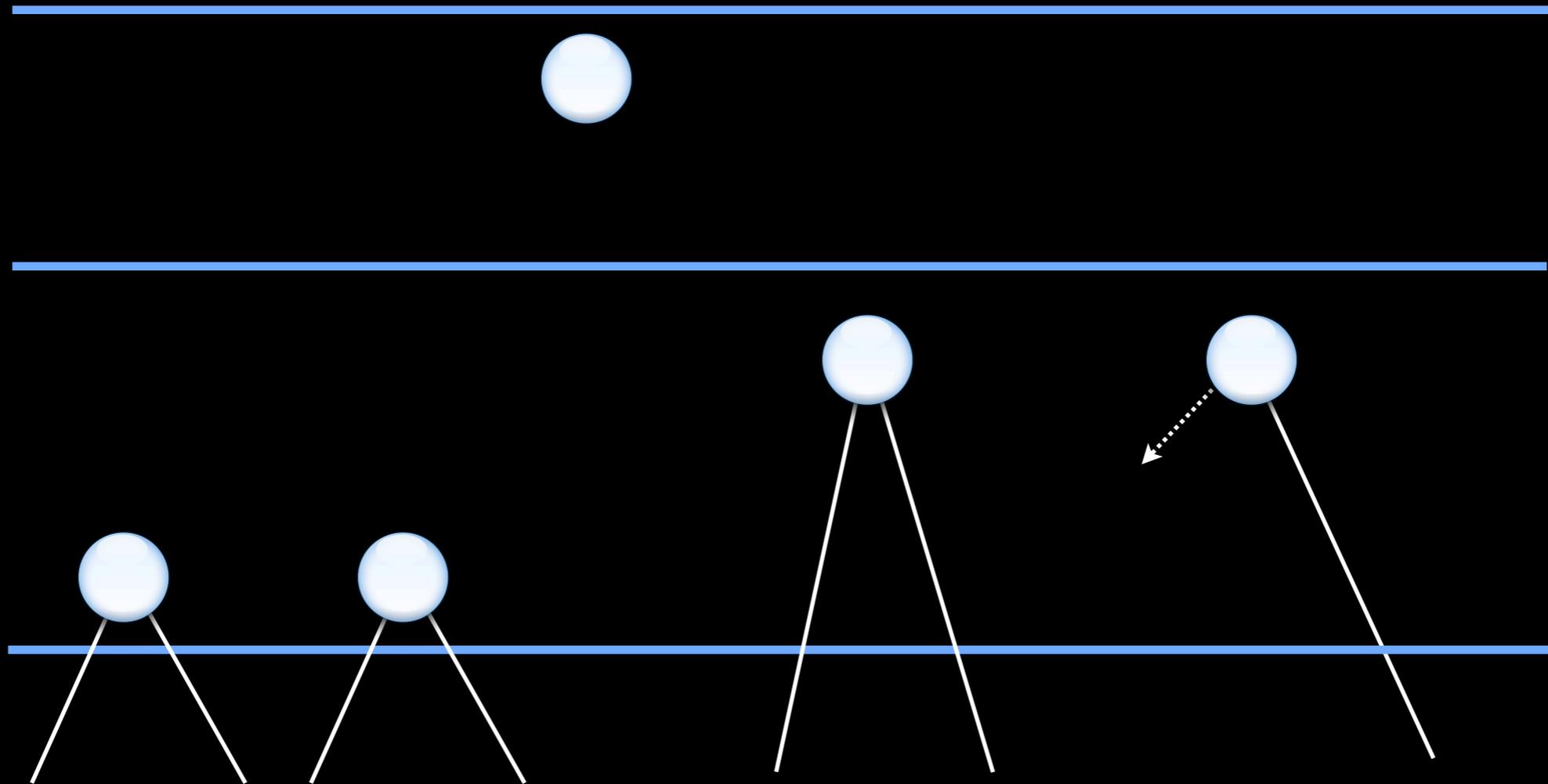
High Half Links Case I



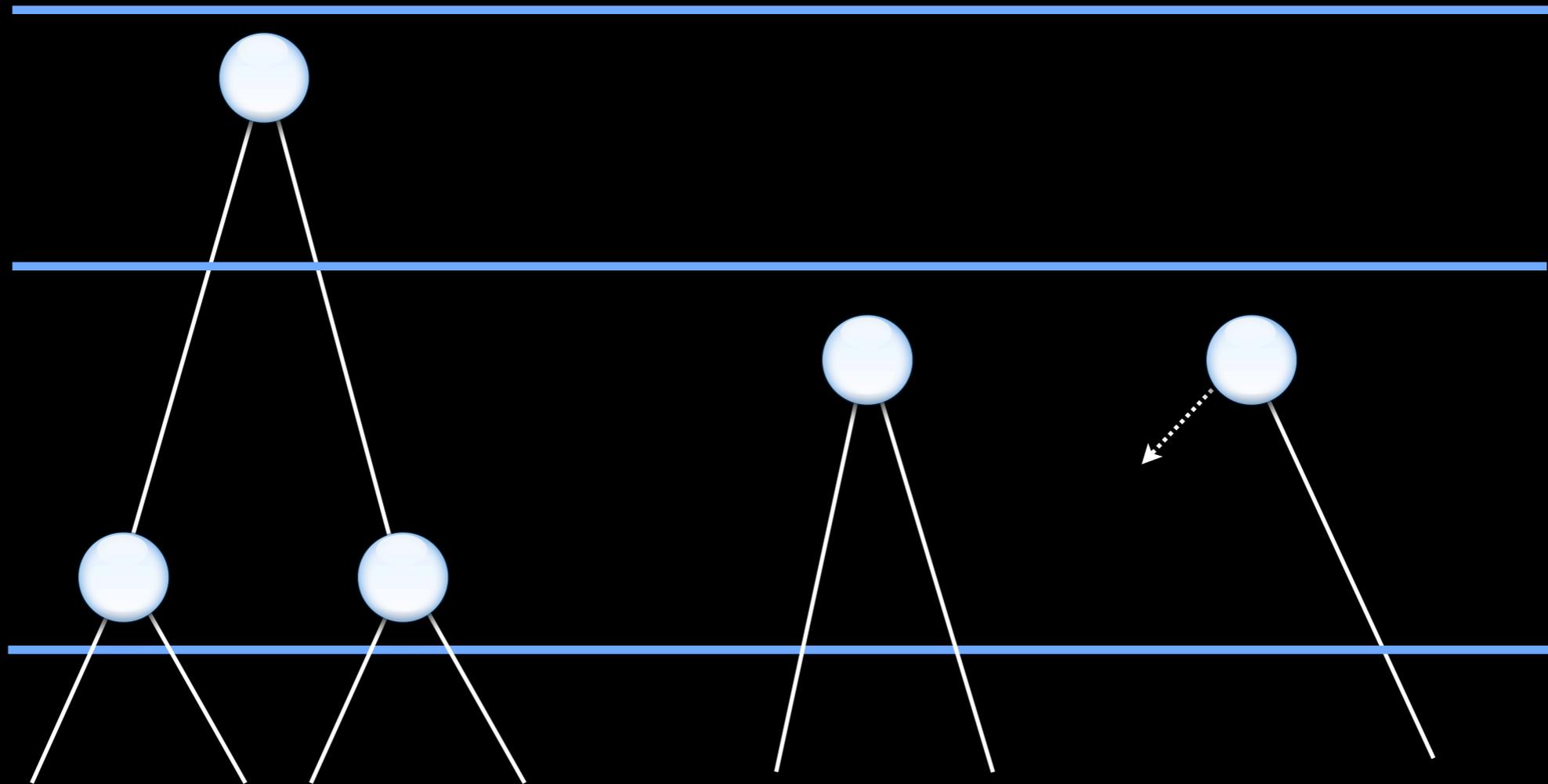
High Half Links Case 2



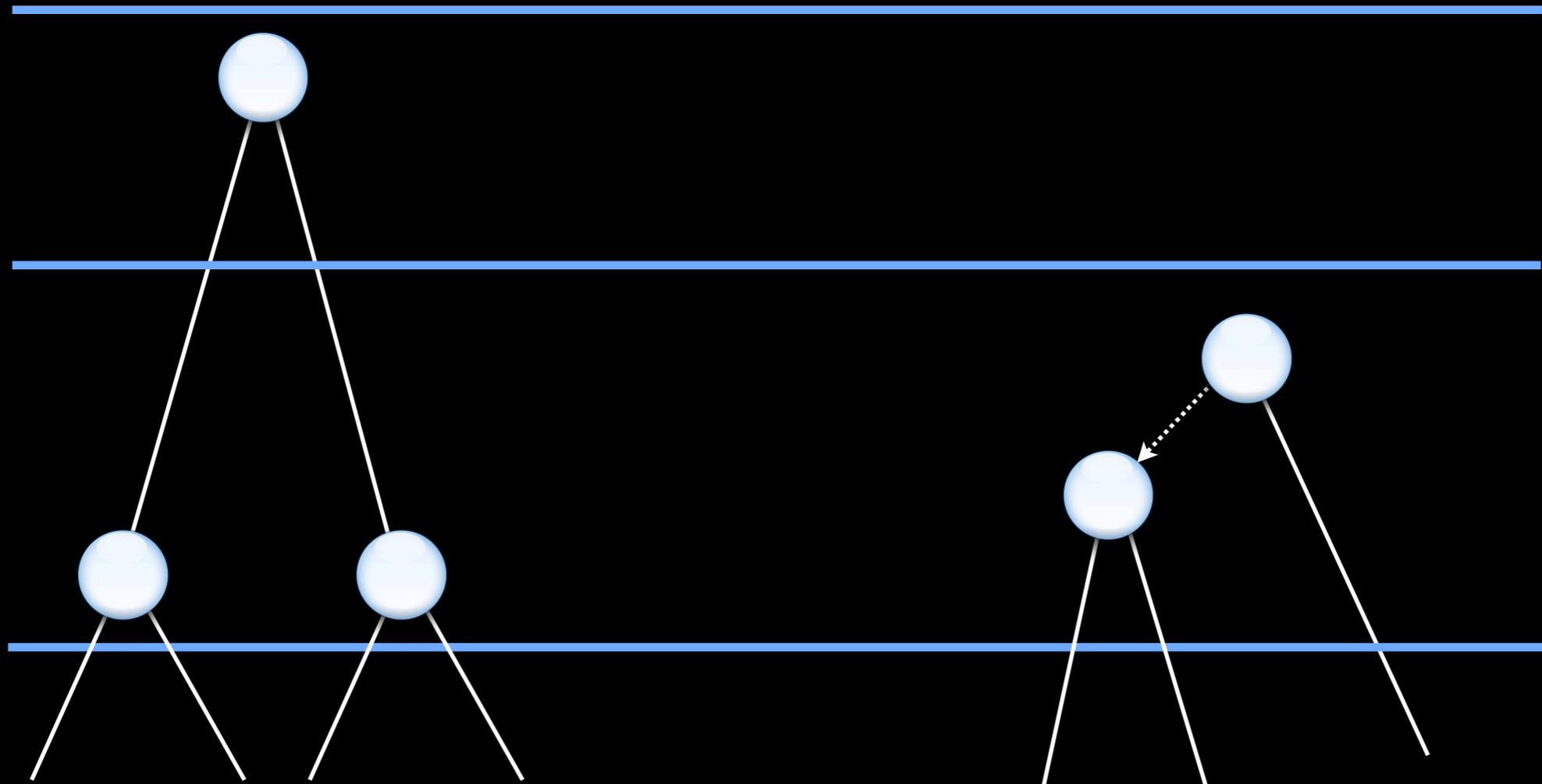
High Half Links Case 2



High Half Links Case 2



High Half Links Case 2

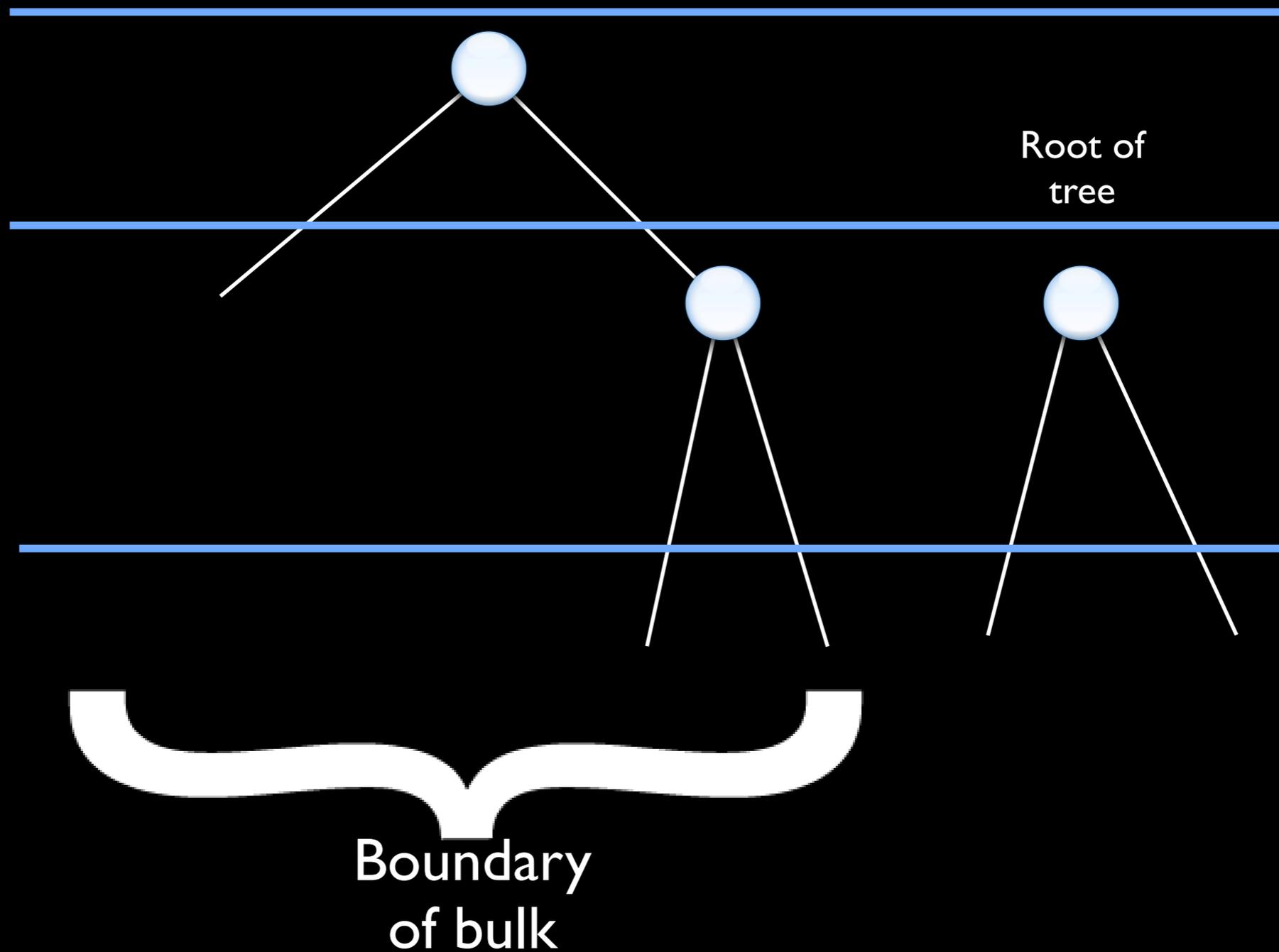


Root Layer

- Treated differently
- Two cases

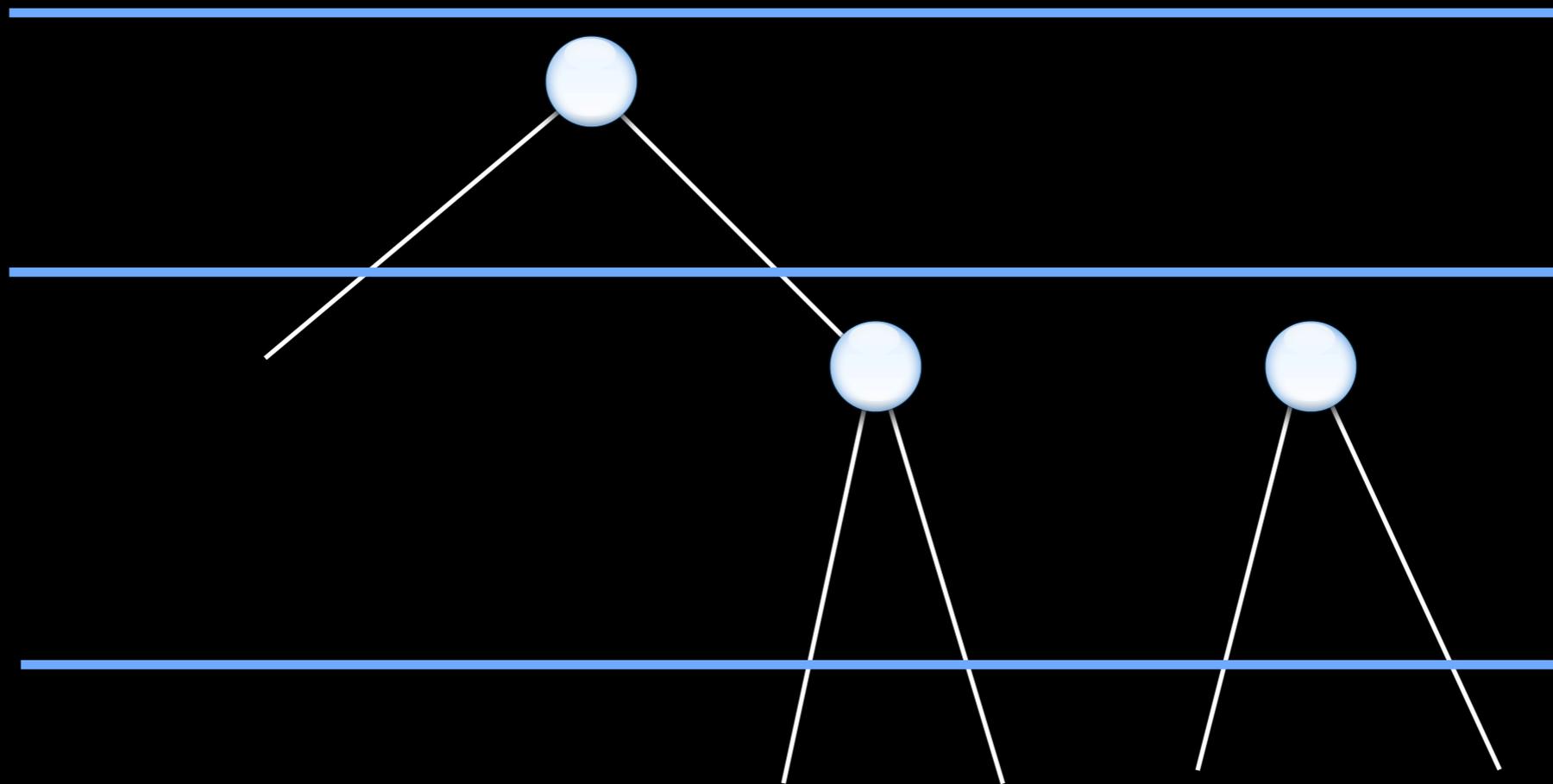
The root of the tree is lower

The root of the bulk tree is at a higher level than the root of the search tree



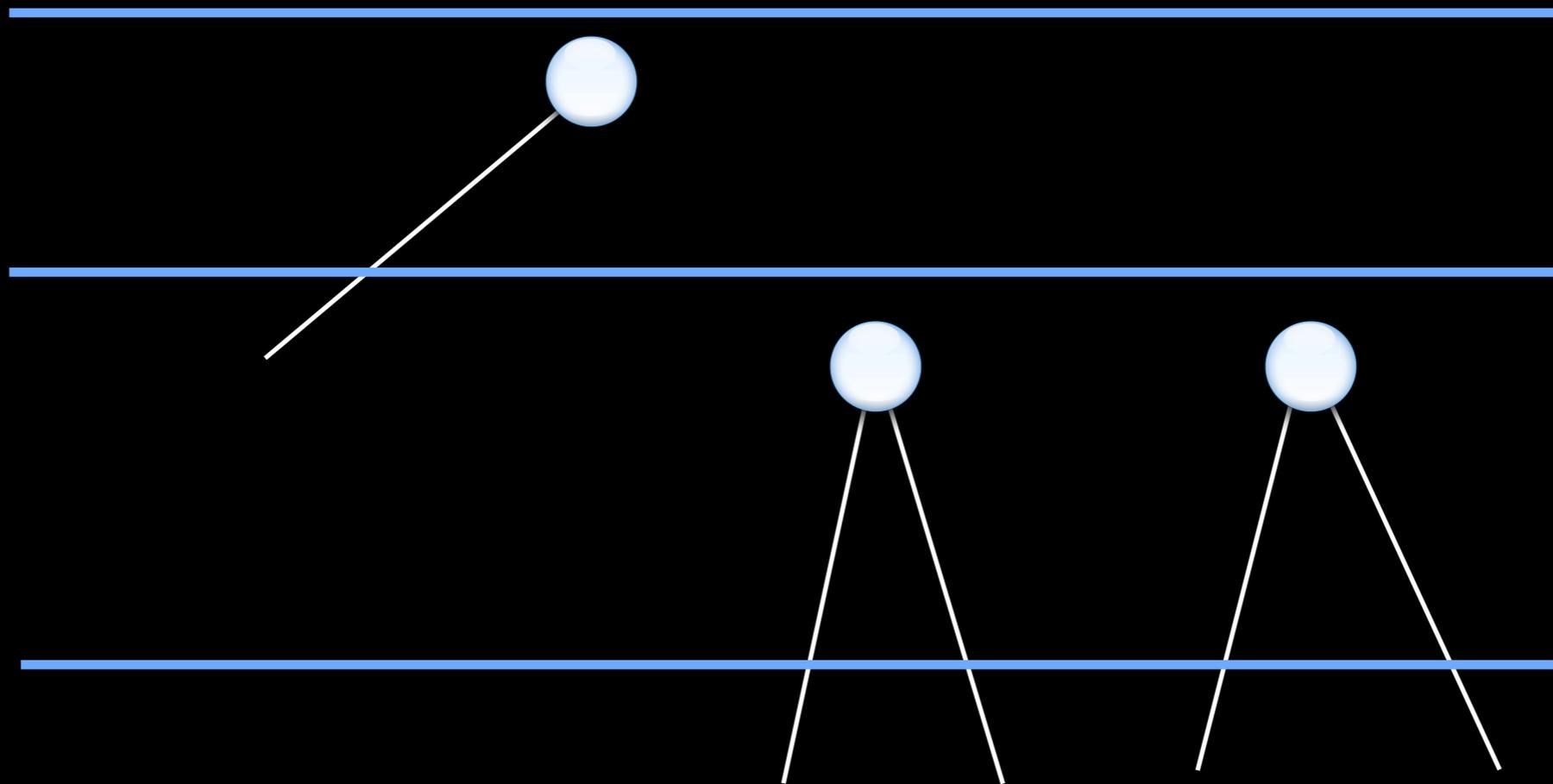
The root of the tree is lower

The root of the bulk tree is at a higher level than the root of the search tree



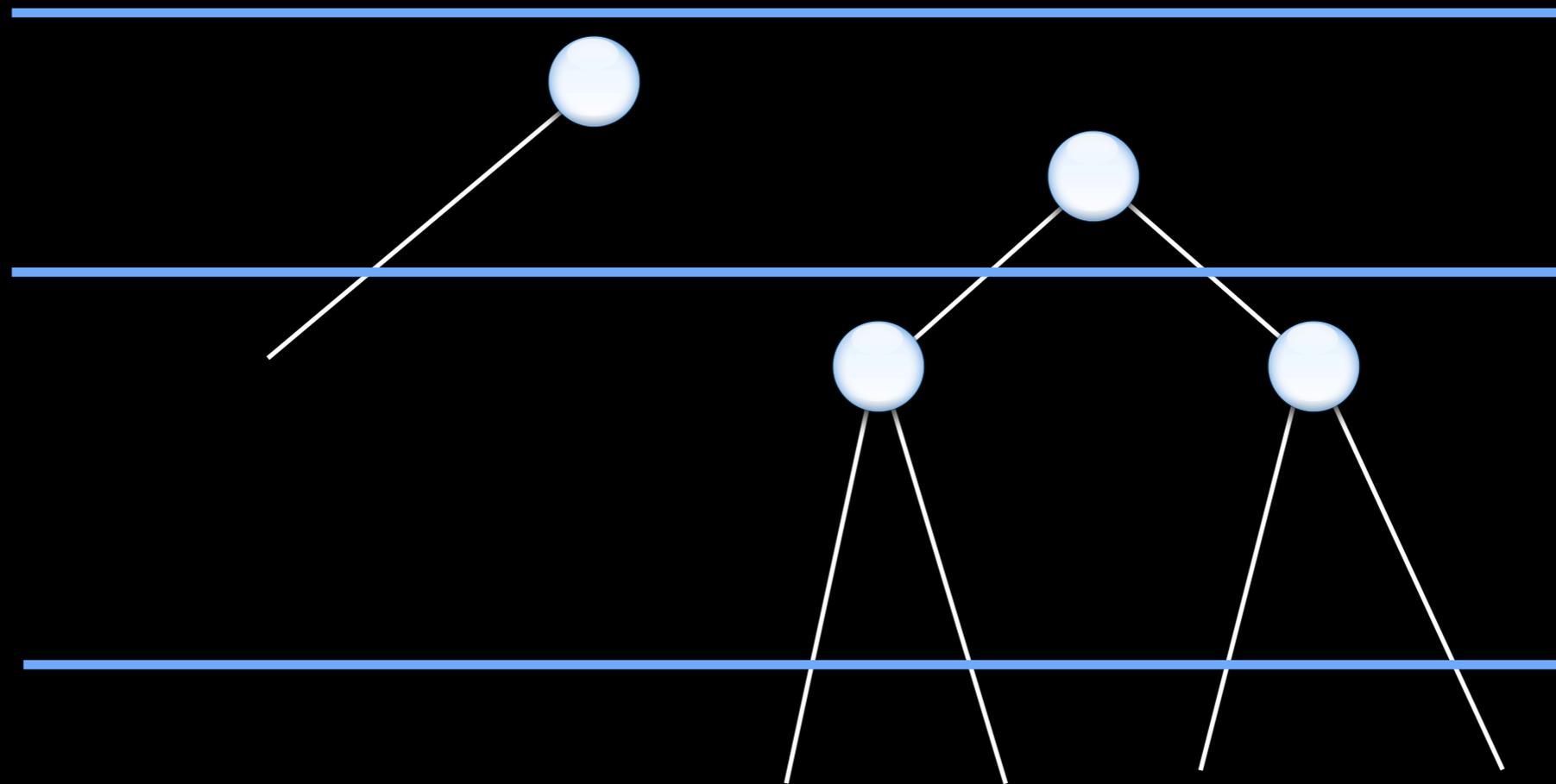
The root of the tree is lower

The root of the bulk tree is at a higher level than the root of the search tree



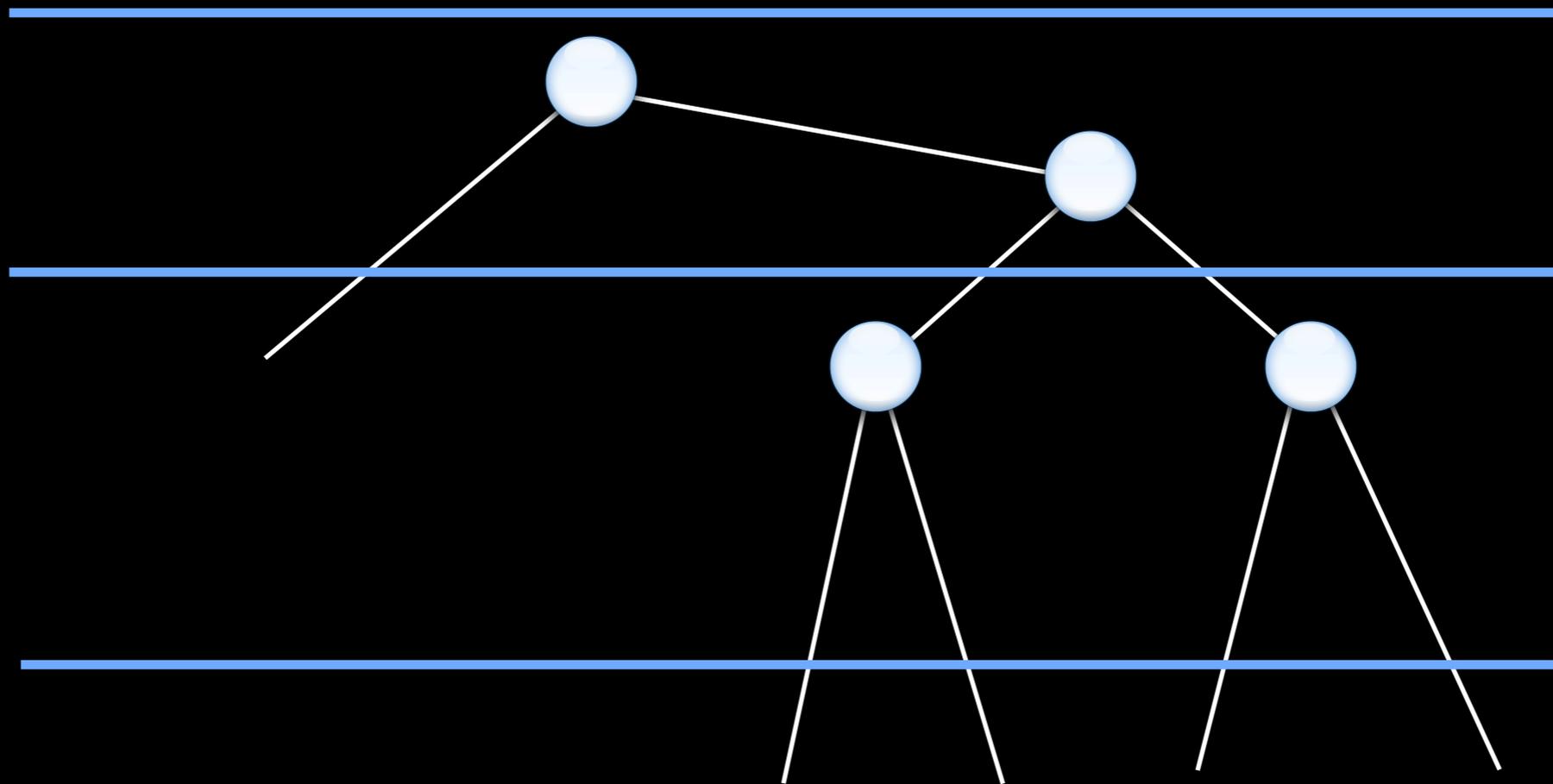
The root of the tree is lower

The root of the bulk tree is at a higher level than the root of the search tree



The root of the tree is lower

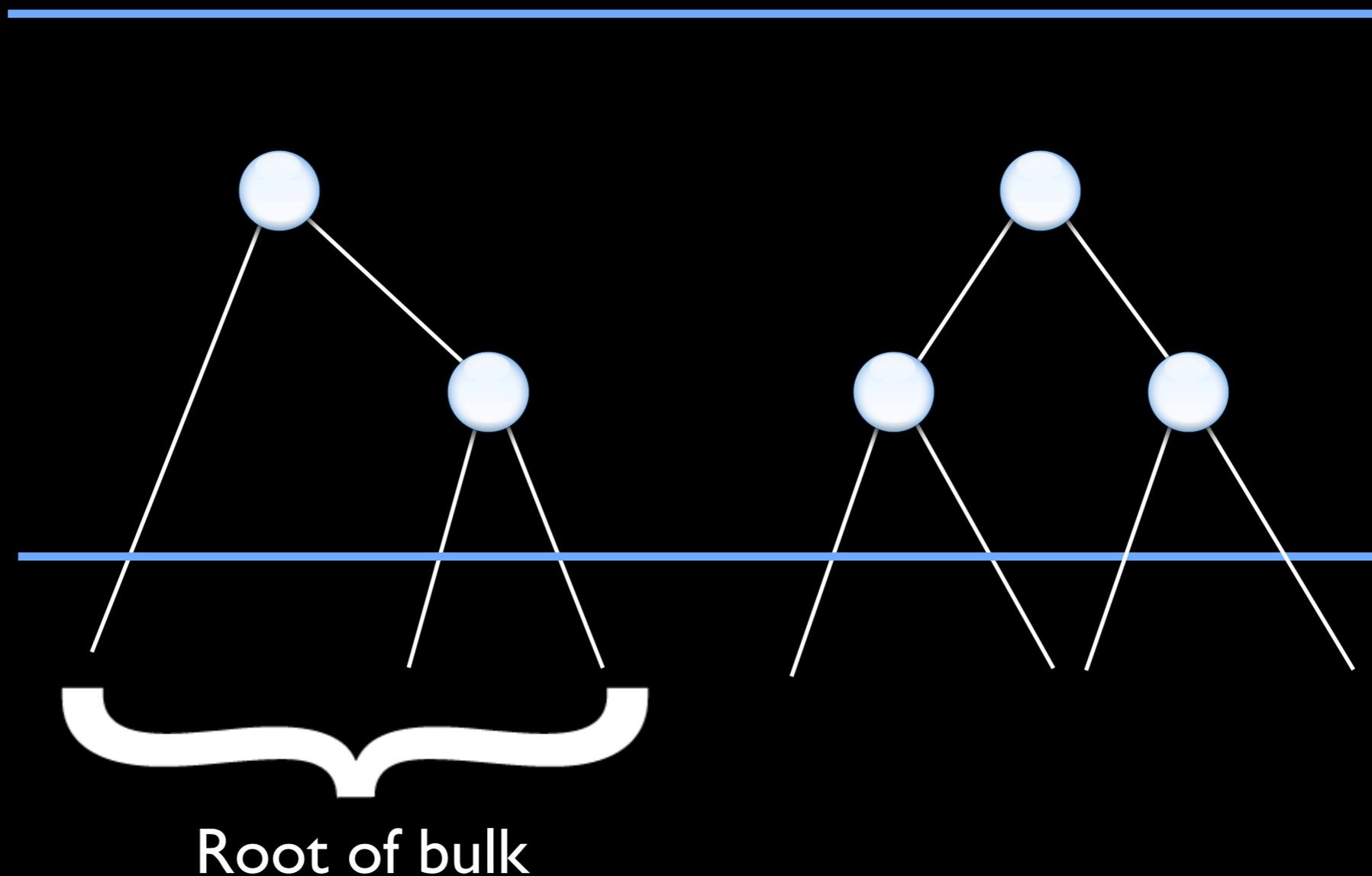
The root of the bulk tree is at a higher level than the root of the search tree



The root of the tree is not lower

The root of the bulk tree is at a lower or equal level than the root of the search tree

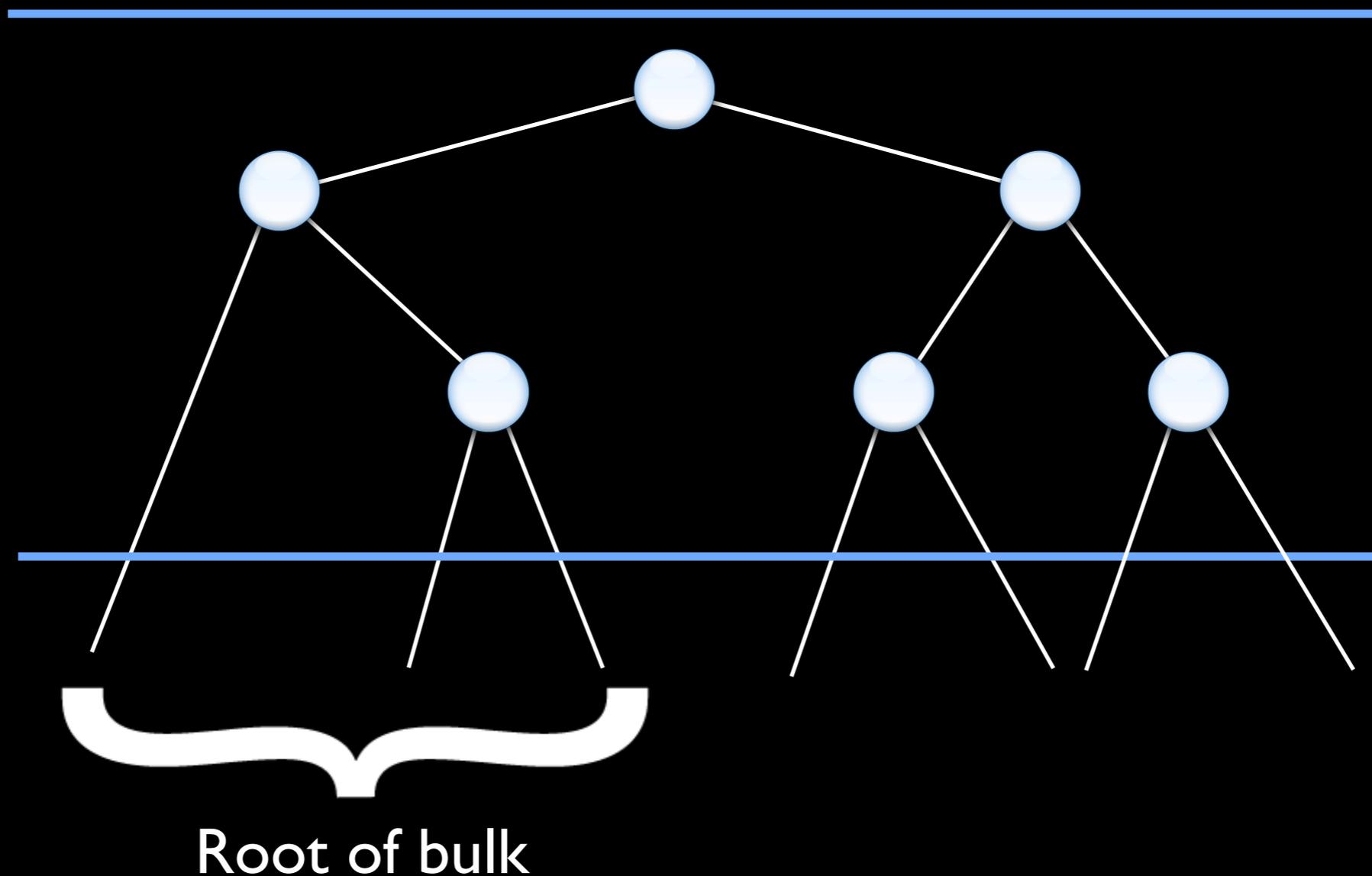
Case I



The root of the tree is not lower

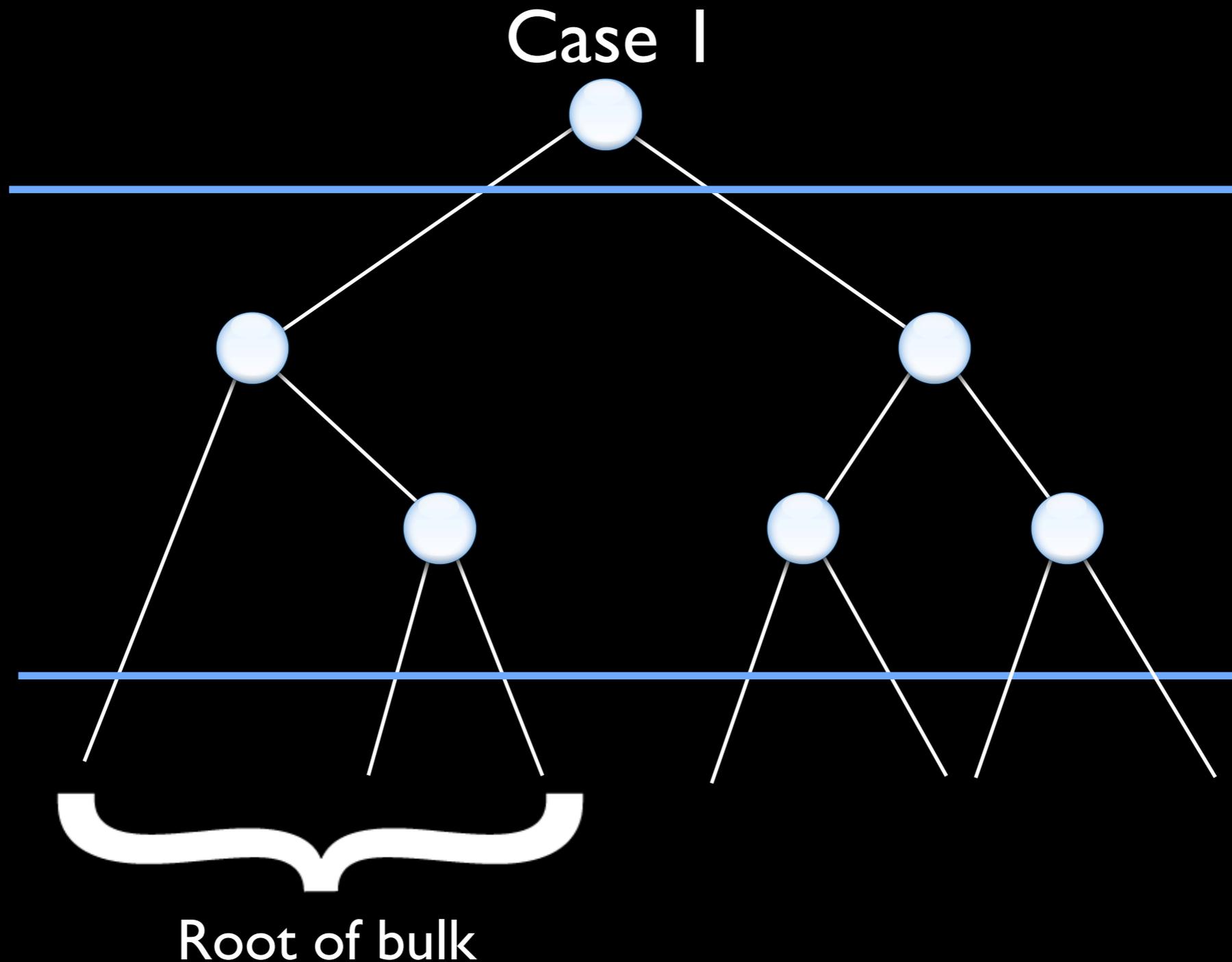
The root of the bulk tree is at a lower or equal level than the root of the search tree

Case I



The root of the tree is not lower

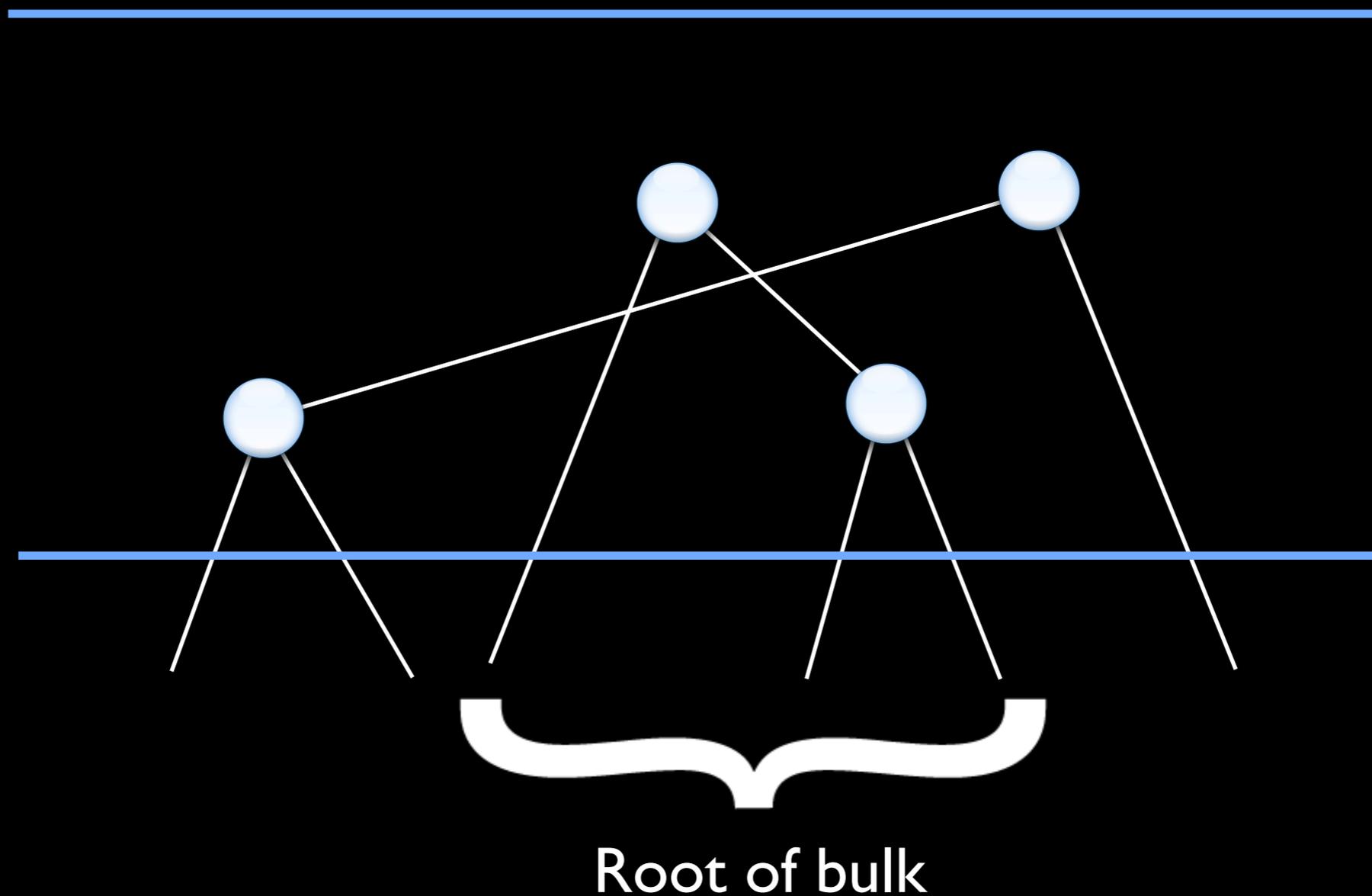
The root of the bulk tree is at a lower or equal level than the root of the search tree



The root of the tree is not lower

The root of the bulk tree is at a lower or equal level than the root of the search tree

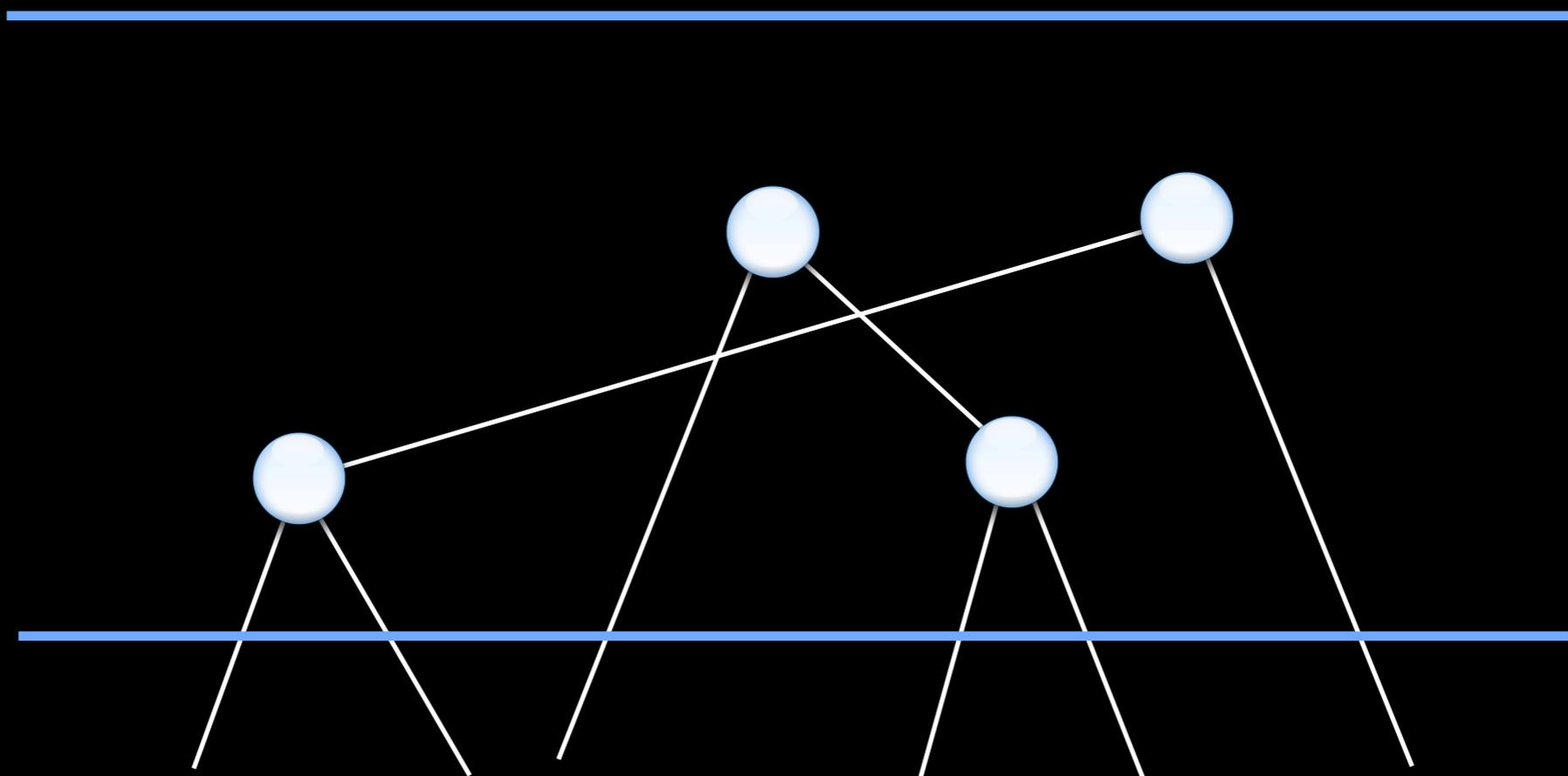
Case 2



The root of the tree is not lower

The root of the bulk tree is at a lower or equal level than the root of the search tree

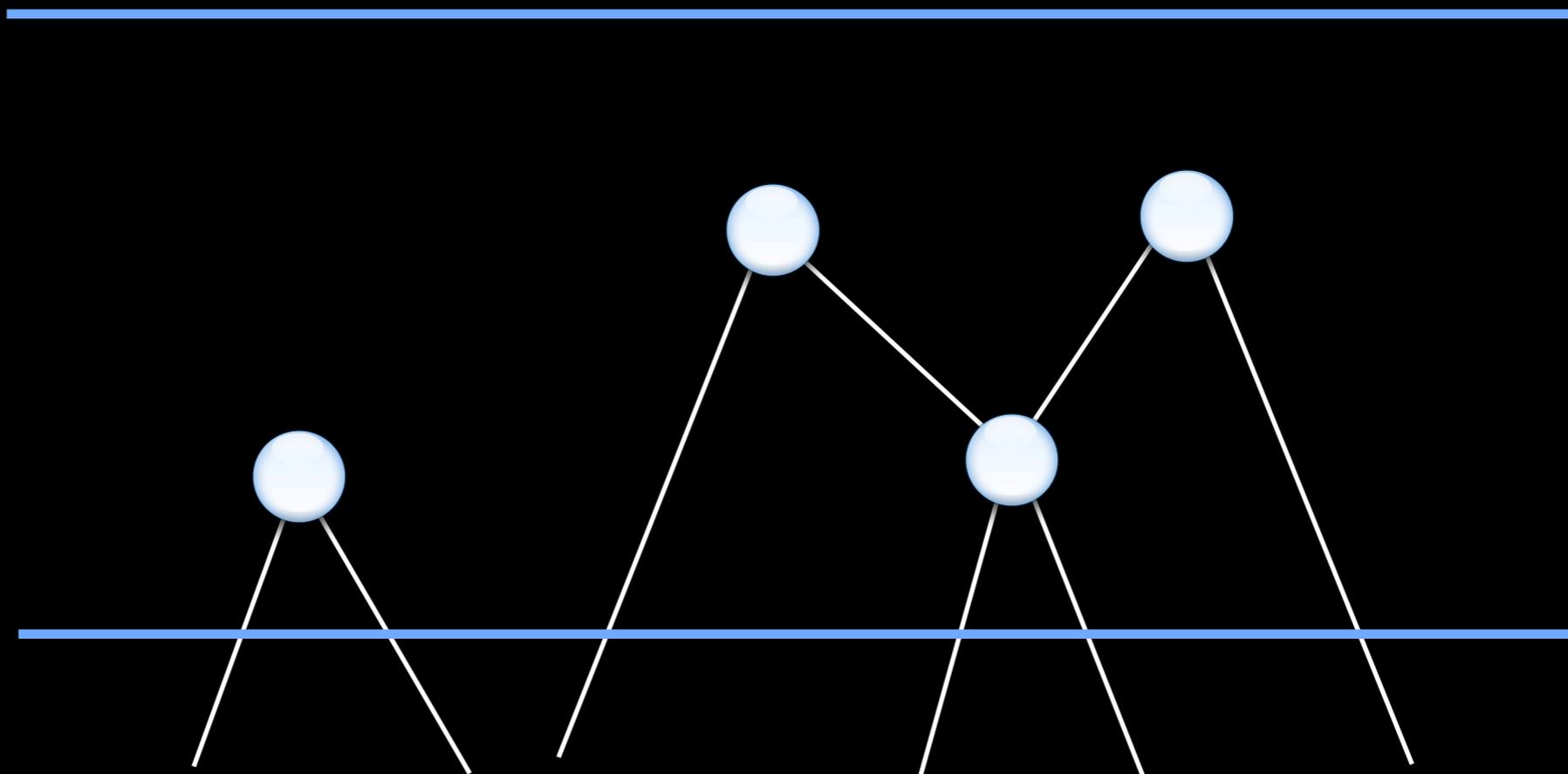
Case 2



The root of the tree is not lower

The root of the bulk tree is at a lower or equal level than the root of the search tree

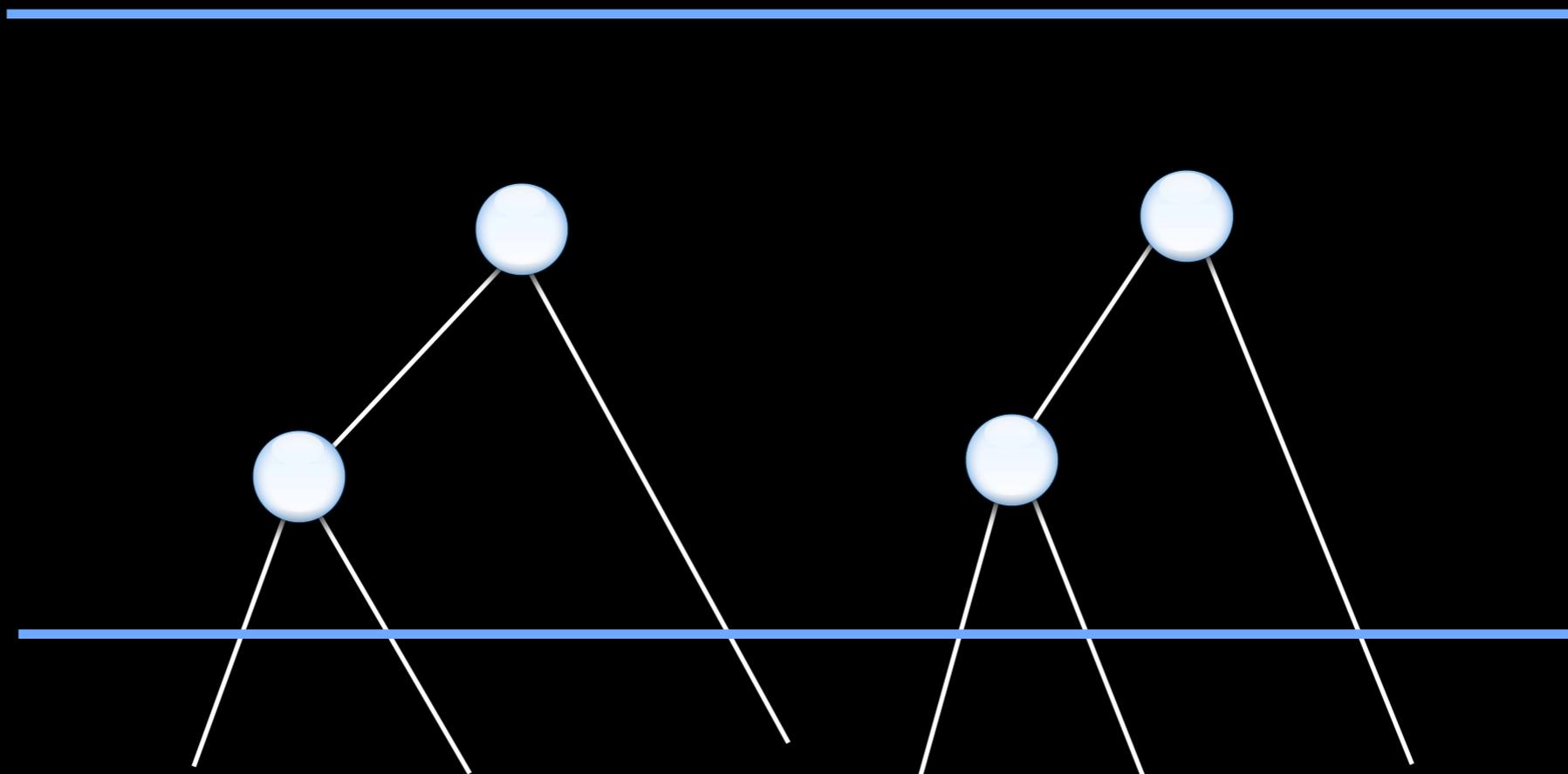
Case 2



The root of the tree is not lower

The root of the bulk tree is at a lower or equal level than the root of the search tree

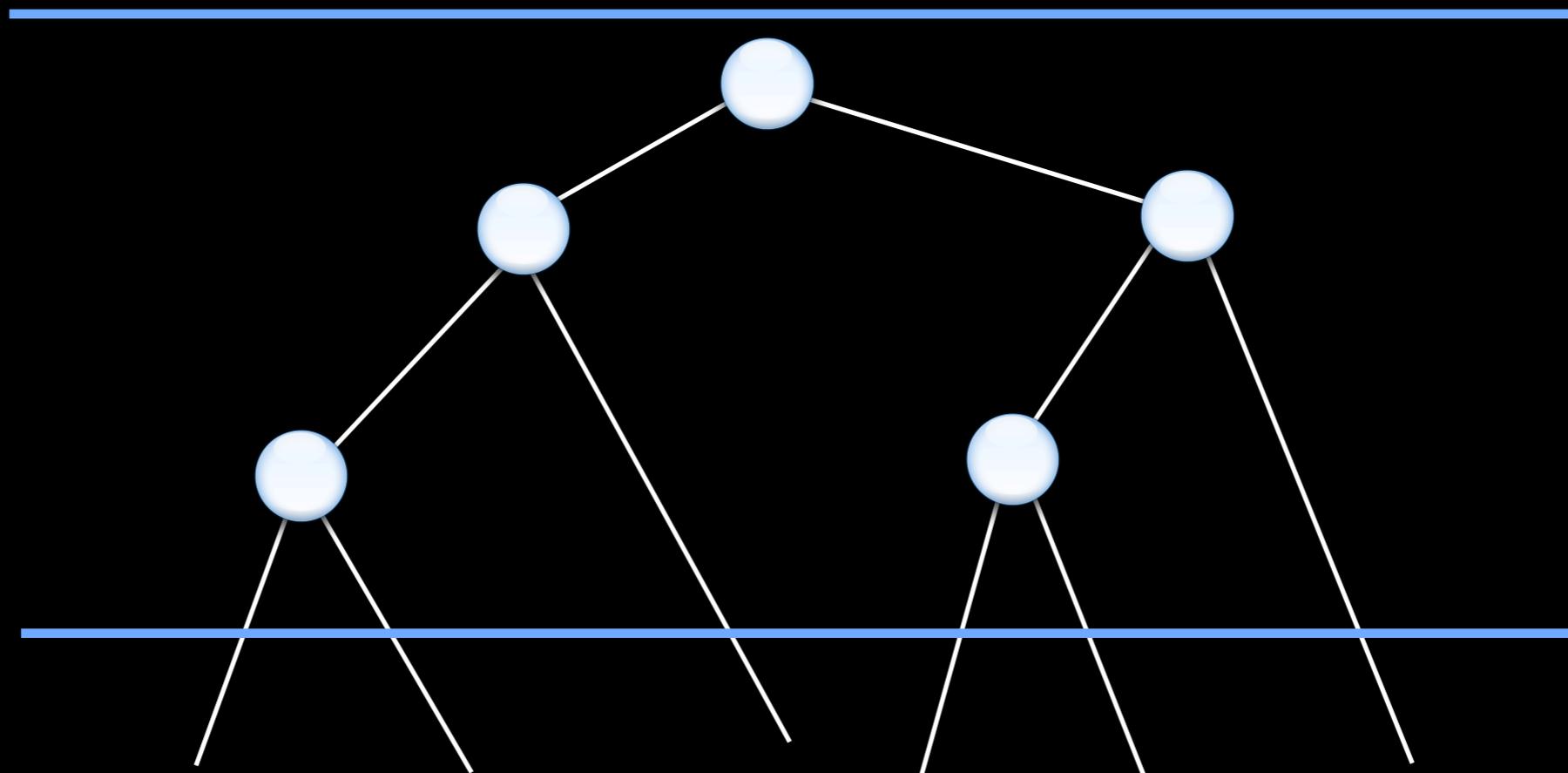
Case 2



The root of the tree is not lower

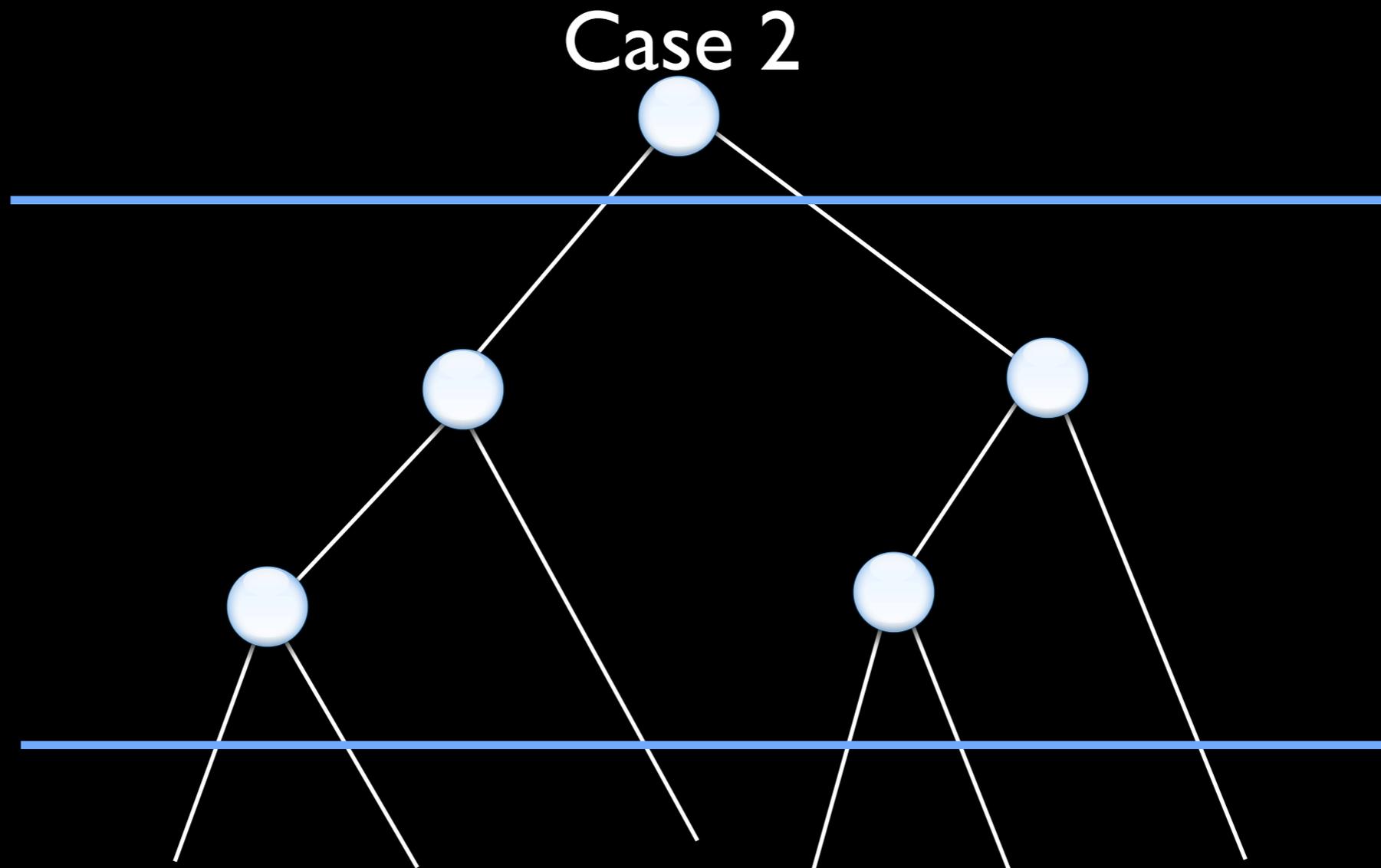
The root of the bulk tree is at a lower or equal level than the root of the search tree

Case 2



The root of the tree is not lower

The root of the bulk tree is at a lower or equal level than the root of the search tree



Conclusion

- Bulk insertion Complexity Analysis
 - Bulk linkage cost is $O(\log m)$
 - Bulk insertions together with singleton insertions and deletions, the amortized cost is $O(\log m)$