- constant time: O(1)

    ☑constant number of operations

- logarithmic time: $O(\log n)$

  ☑ the depth of a balanced tree

- linear time: O(n)

  ☑ the complexity of iterating through a list

- linearithmic time: O(n log n)

  ☑ fastest sorting algorithms

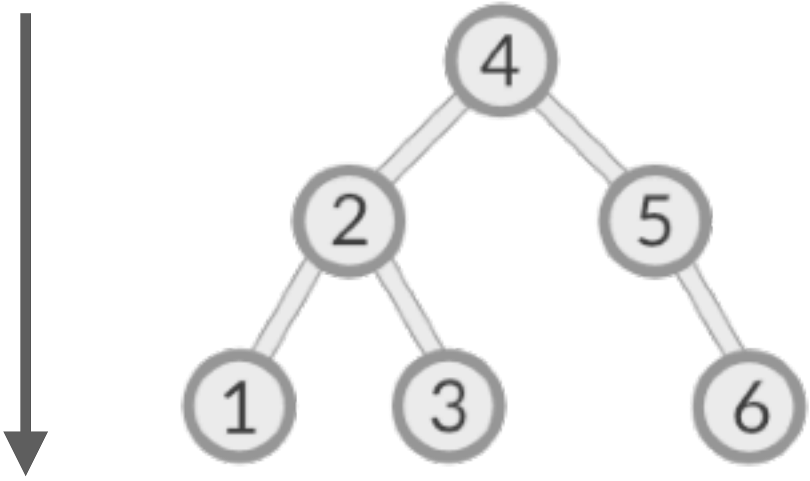
- exponential time: $O(n^2)$, $O(n^3)$…

    ☑ for in for

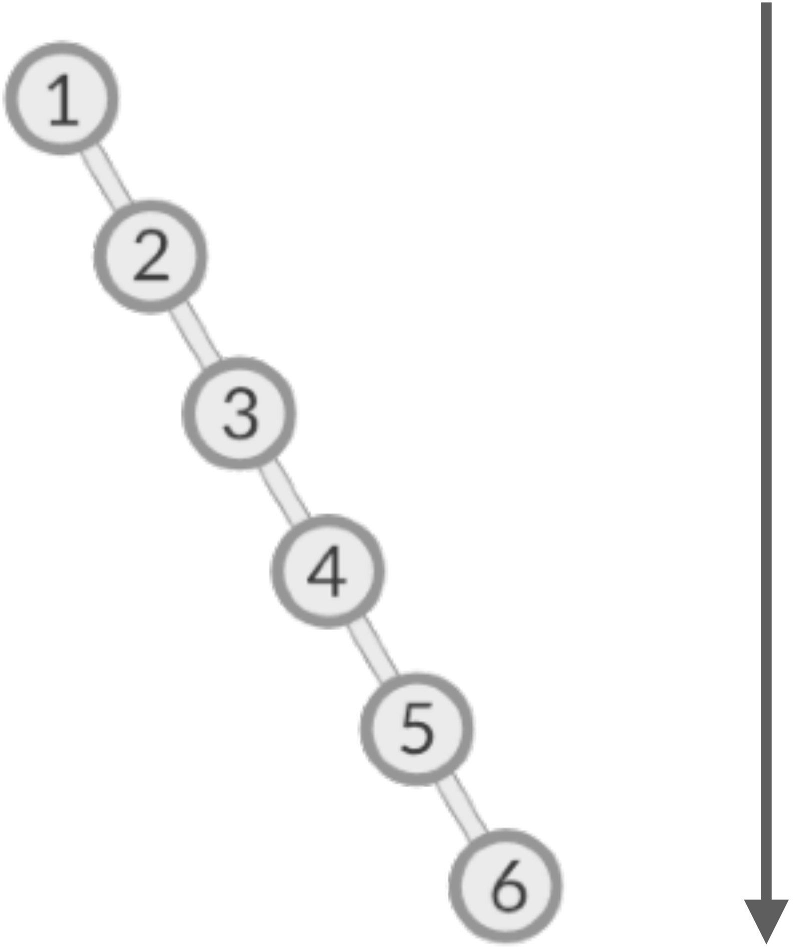
Time Complexities Refresh

maximum number of operations that an algorithm may perform

Balanced: $\log_2(n)$



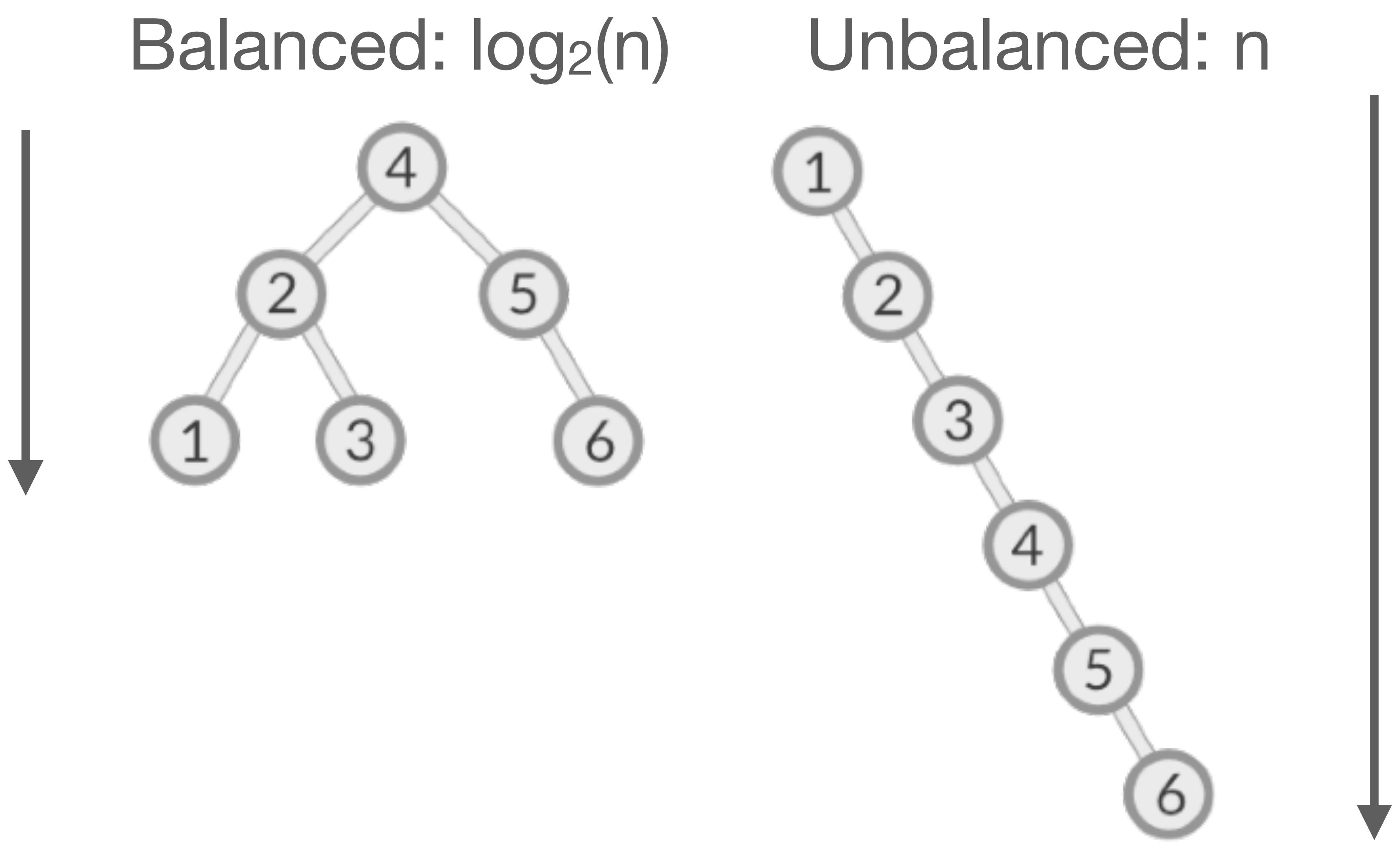
Unbalanced: n



Time Complexities Refresh

maximum number of operations that an algorithm may perform

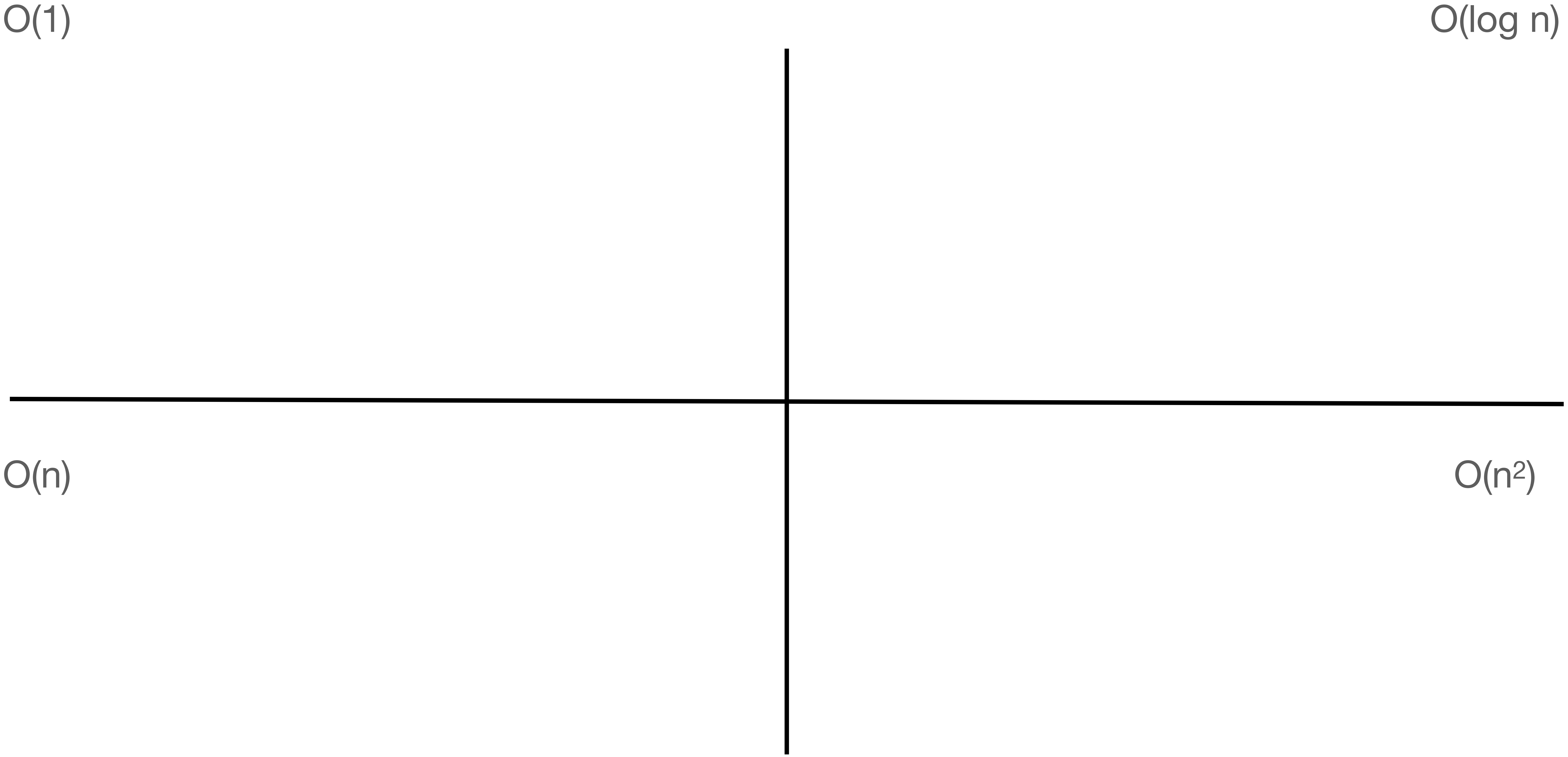
- constant time: O(1)
  - ☑constant number of operations
- logarithmic time: O(log n)
  - ☑the depth of a balanced tree
- linear time: O(n)
  - ☑the complexity of iterating through a list
- linearithmic time: O(n log n)
  - ☑fastest sorting algorithms
- exponential time: $O(n^2)$, $O(n^3)$…
  - ☑for in for



Balanced: $\log_2(n)$

Unbalanced: n

Time Complexities Game



O(1)

O(log n)

$O(n)$

$O(n^2)$