Heapsort (1964)

Adapted from the CLRS book slides

THE HEAPSORT ALGORITHM

Given an input array, the heapsort algorithm acts as follows:

- Builds a max-heap from the array.
- Starting with the root (the maximum element), the algorithm places the maximum element into the correct place in the array by swapping it with the element in the last position in the array.
- "Discard" this last node (knowing that it is in its correct place) by decreasing the heap size, and calling MAX-HEAPIFY on the new (possibly incorrectly-placed) root.
- Repeat this "discarding" process until only one node (the smallest element) remains, and therefore is in the correct place in the array.
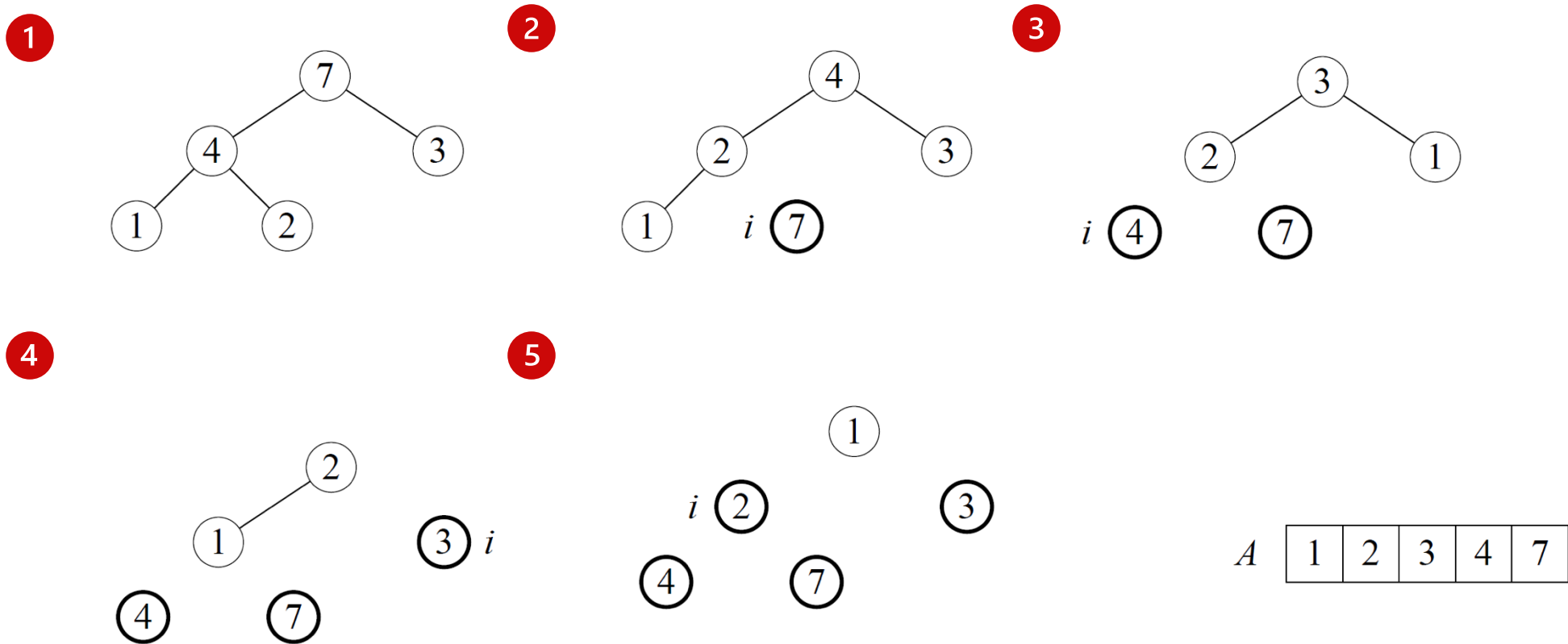
PSEUDOCODE

$\text{HEAPSORT}(A, n)$

1    $\text{BUILD-MAX-HEAP}(A, n)$
2   **for** $i = n$ **downto** 2
3   exchange $A[1]$ with $A[i]$
4        $A.heap\text{-}size = A.heap\text{-}size - 1$
5        $\text{MAX-HEAPIFY}(A, 1)$

EXAMPLE

Sort an example heap on the board. [Nodes with heavy outline are no longer in the heap.]

A | 1 | 2 | 3 | 4 | 7

ANALYSIS

- BUILD-MAX-HEAP: $O(n)$
- **for** loop: $n - 1$ times
- exchange elements: $O(1)$
- MAX-HEAPIFY: $O(\lg n)$

Total time: $O(n \lg n)$.

Though heapsort is a great algorithm, a well-implemented quicksort usually beats it in practice.