Illustration of a Max-heap

(a) Illustrates relationships between values in the array

(b) Binary heap is an array satisfying heap property

Note array is 1-indexed.
**Parent(i)**

1. return \( \lfloor i/2 \rfloor \)

**Left(i)**

1. return \( 2i \)

**Right(i)**

1. return \( 2i + 1 \)

\( \text{Left}(i) \) is a max-heap, \( \text{Right}(i) \) is a max-heap, \( 1 \leq i \leq A.\text{heap-size} \).

**Max-Heapify(A, i)**

1. \( l = \text{Left}(i) \)
2. \( r = \text{Right}(i) \) \( \text{if} \ l \leq A.\text{heap-size} \text{ and } A[l] > A[i] \)
3. \( \text{largest} = l \)
4. \( \text{if } r \leq A.\text{heap-size} \text{ and } A[r] > A[\text{largest}] \)
5. \( \text{largest} = r \)
6. \( \text{if largest} \neq i \)
7. exchange \( A[i] \) with \( A[\text{largest}] \)
8. \( \text{Max-Heapify}(A, \text{largest}) \)

\( i \) is a max-heap.
1, 4, 2, 3, 9, 7, 8, 10, 14, 16

All are trivially max-heaps

10 14 16
null null

14 16 7
10 3 9

These are max-heaps

These are max-heaps

These are max-heaps
Max-heapify

\[ n = A.heap\_size \]

*best-case is \( \Theta(1) \)*

worst-case is \( \Theta(\lg n) \)

\[ T(n) = T\left(\frac{n}{2}\right) + 2 \]

\( T(n) \approx 2 \lg n \)

Max-heapify is \( \Omega(1) \) and \( \Theta(\lg n) \)