Heapify A[1..n] in O(n) time


2. Exchange A[1] with A[i], where i should be the largest

3. Reposition heap order within A[1..i]

Example: A = (1, 3, 2, 16, 9, 10, 14, 8, 7, 1, 2, 4, 19, 10, 18, 7, 17, 20, 11, 6)
**Property Query as a Data Structure**

- heap: data structure for maintaining a set of elements, each with a key and data
- extract-max: removes and returns the element of $S$ with the largest key
- extract-min: removes and returns the element of $S$ with the smallest key
- insert: inserts a new element into the heap
- decrease-key: changes the key of an element

*Heap Operations*

- Make heap key
- Insert key
- Extract max
- Extract min

*Heap Example*

H = (1, 2, 3, 4, 5)

- insert(6)
- extract-max
- decrease-key(2, 1)
- extract-min
- insert(7)
- extract-max

**Implementation**

- A heap $H$ is a complete binary tree.
- The value of a node $v$ is always greater than or equal to the values of its children.

**Extract Max and Min**

- $O(1)$
- $O(1)$
- $O(1)$
- $O(1)$
- $O(log n)$

**Complexity**

- $O(n)$
- $O(n)$
- $O(n)$
- $O(n)$
- $O(n)$

**Application**

- Implement a priority queue to manage tasks with different priorities.
Heap insertion algorithm:

1. If the tree is empty, create a new node and return it.
2. Insert the new node into the tree.
3. Update the heap size.

Heap deletion algorithm:

1. Remove the root node.
2. Replace the root with the last node in the tree.
3. Update the heap size.

Bubble sort:

1. Compare each pair of adjacent items.
2. Swap them if they are in the wrong order.
3. Repeat until no swaps are needed.

Example:

- Heap insertion:
  - Insert a new node into the tree.
  - Update the heap size.

- Heap deletion:
  - Remove the root node.
  - Replace the root with the last node in the tree.
  - Update the heap size.

- Bubble sort:
  - Compare each pair of adjacent items.
  - Swap them if they are in the wrong order.
  - Repeat until no swaps are needed.

Heaps are a property queue.
<table>
<thead>
<tr>
<th>Name</th>
<th>Upper bound</th>
<th>Lower bound</th>
<th>Tight bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-st order</td>
<td>(n)</td>
<td>(n)</td>
<td>(n)</td>
</tr>
<tr>
<td>n-th order</td>
<td>(n^2)</td>
<td>(n^2)</td>
<td>(n^2)</td>
</tr>
</tbody>
</table>