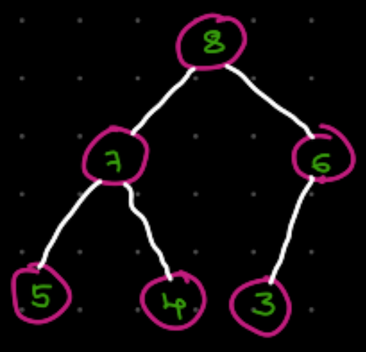
↳ A Heap is a special tree based data structure

- It is a complete Binary tree
- Two types of heaps
  - → Max heap
  - → Min heap

① Max heap:



↳ The root value must be greatest among all it's descendant nodes. All subtrees should follow this condition

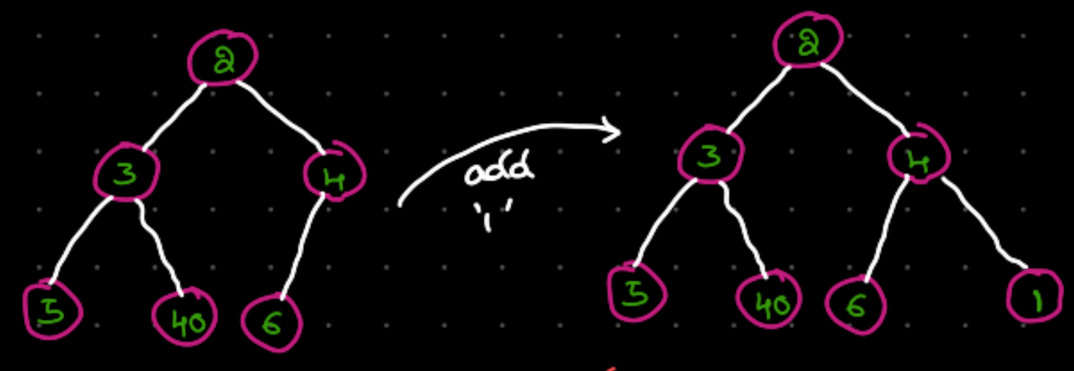
① Min heap:

↳ The root value must be Smallest among all it's descendant nodes. All subtrees should follow this condition

* It is more likely represented by a list of elements.

Functions while implementing Heap:

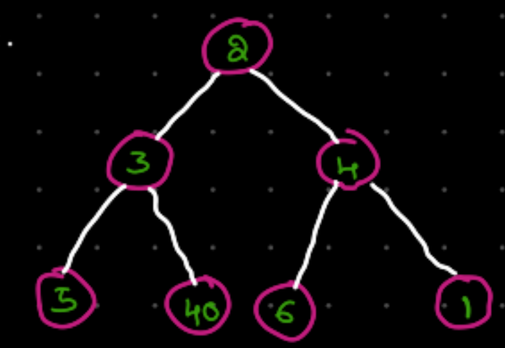
- `__len__`:
  - ↳ return no of nodes / vertices (simply no of elements in a heap)
- `__repr__`:
  - ↳ simply returns the list of elements
- insert:
  - ↳ Inserting a element into a heap needs to follow heap property
    - for a min heap if we add Smallest element at last it's fine
      - ↳ But in General we think about worst case so what if we insert the Smallest element at the last
        - → To maintain the heap property we use "_shift-up" function



↳ not maintaining heap Property * restructure *

↳ to restructure we use _shift-up function

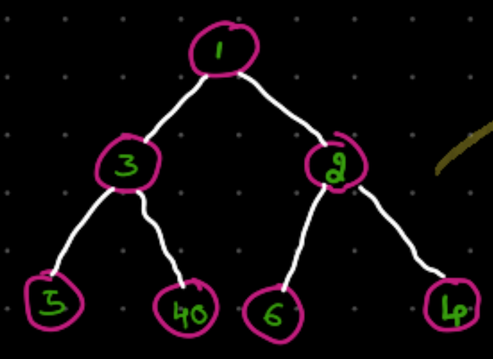
1.



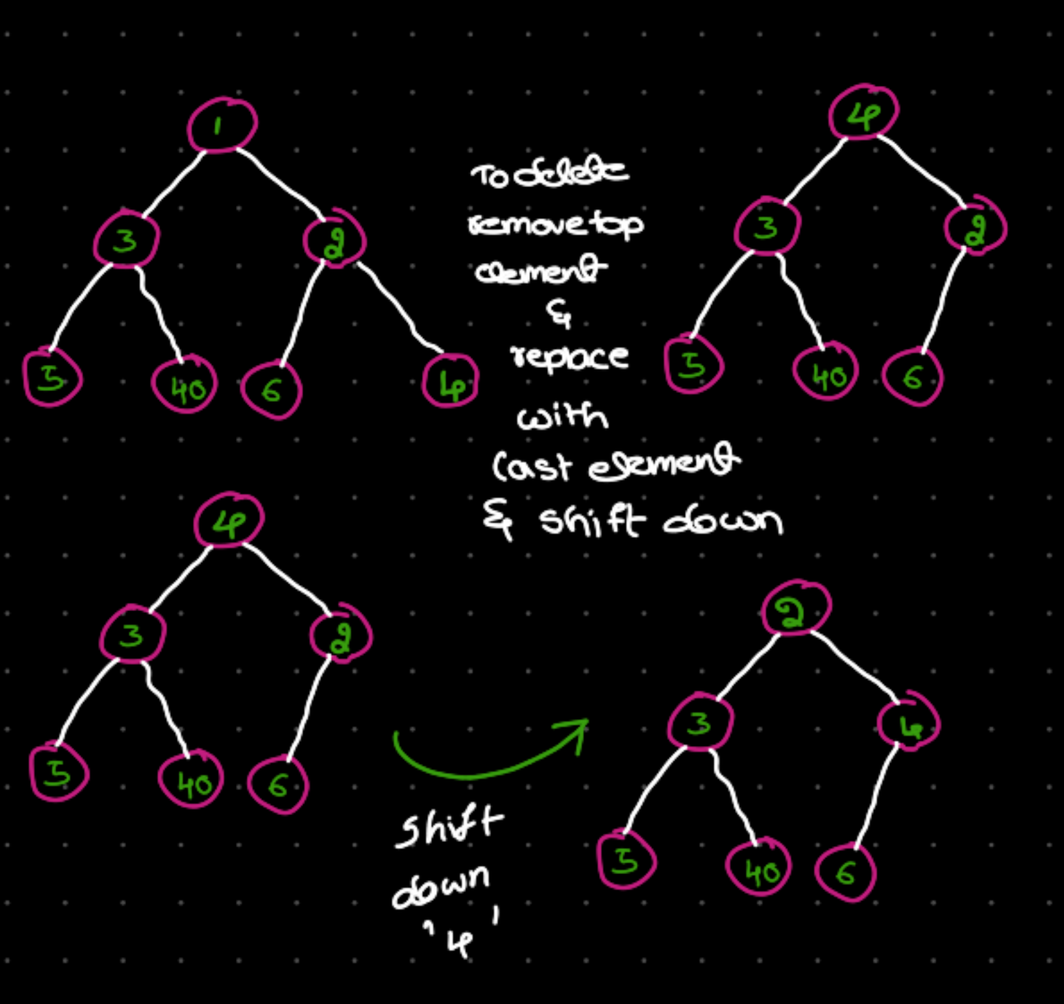
Shift UP

This is what the _shift_up function do

- extract_min
  - ↳ * As in python in built functions * heapq * has only min heap we are implementing min heap
    - → In order to extract min element we need to remove last element & add it in place of first element (min-element) & '_shift_down' the element in order to get heap structure correct



Extract Min element ↳ 1



* This is what the _shift_up function do *

* Time Complexity:

- Insert : O(logn)
  ↳ worst case
- Peek : O(1)
- extract_min : O(logn)
- heapify : O(n)
- meld : O(n)
- Parent, left, right : O(1)
- _shift_up, _shift_down = O(logn)

Youtube tutorial : Neural nine - Heap (ooDatastructureinpython)