Data Structures

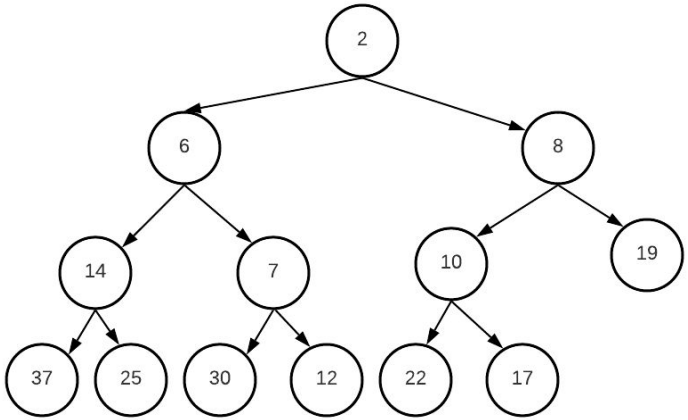
Heap Deletion Code

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

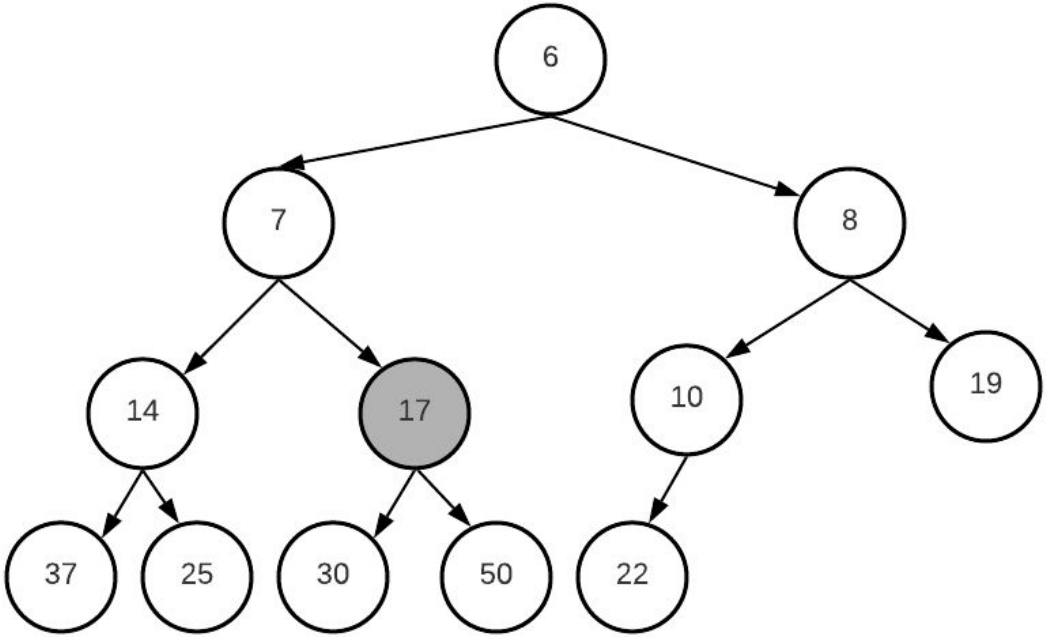
```
def _heapify_down(self, parent_pos):     # O(logn)
    child_pos = self._left(parent_pos)
    right_child = self._right(parent_pos)

    if child_pos == -1: # no children
        return
    # is right smaller than left?
    if right_child != -1 and self.array[right_child] < self.array[child_pos]:
        child_pos = right_child

    if self.array[parent_pos] > self.array[child_pos]:
        self.array[parent_pos], self.array[child_pos] = \
            self.array[child_pos], self.array[parent_pos]
        self._heapify_down(child_pos)
```

```
def pop(self):  # remove the minimum element
    assert not self.empty()
    self.size -= 1
    result = self.array[0]
    self.array[0] = self.array[self.size]
    self._heapify_down(0)
    return result
```



Let’s remove all elements

- Now we have:
  - Value 2 (min)
  - Fixed tree
- Imagine we keep popping the elements until the heap is empty
- Can you guess the output?



Let’s remove all elements: O(nlogn)

```
minHeap = MinHeap()
lst = [2, 17, 22, 10, 8, 37, 14, 19, 7, 6, 5, 12, 25, 30]

for val in lst:
    minHeap.push(val)

print(minHeap.array)
# 2, 5, 12, 8, 6, 14, 22, 19, 17, 10, 7, 37, 25, 30

while not minHeap.empty():
    print(minHeap.pop(), end=', ')
# 2, 5, 6, 7, 8, 10, 12, 14, 17, 19, 22, 25, 30, 37
# Sorted list in O(nlogn)
```

Heap sort

- You will get the array content, but **sorted from small to large**!
  - Or large to small in max-heap
- This is called heap sort!
  - It can be done in-place. See homework
- To sort data, we simply add it to the heap structure:
  - Loop to add O(n)
    - Push is O(logn)
  - Loop to remove O(n)
    - Pop is O(logn)
- To sort n number, our total is O(n log n)!

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."