Data Structures

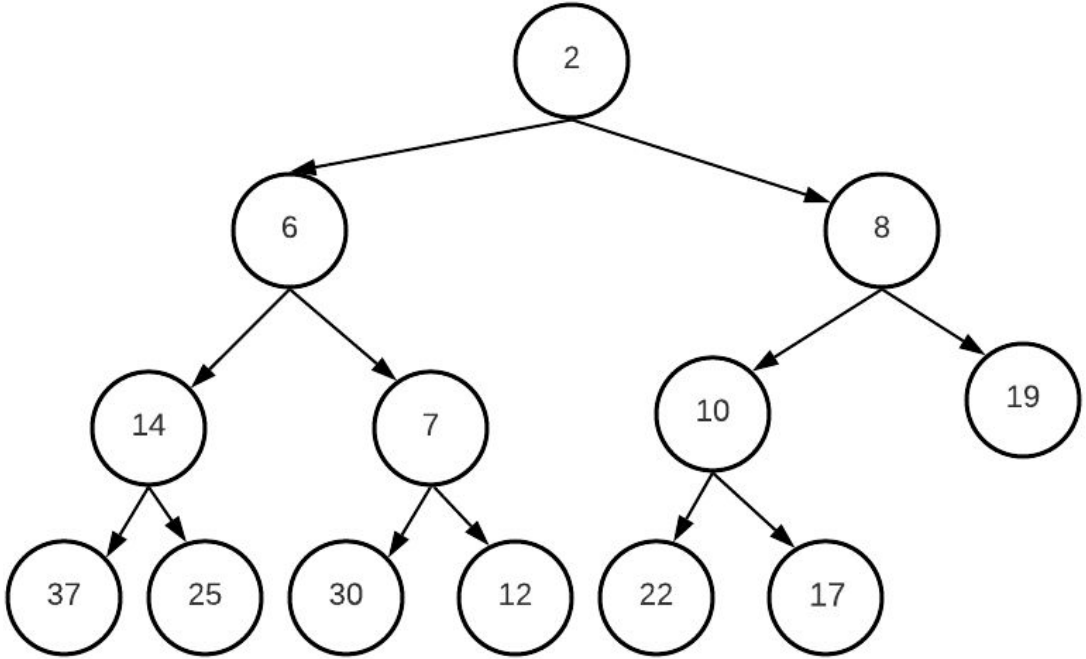
Heap Insertion

Mostafa S. Ibrahim
Teaching, Training and Coaching since 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)

Let’s insert 5

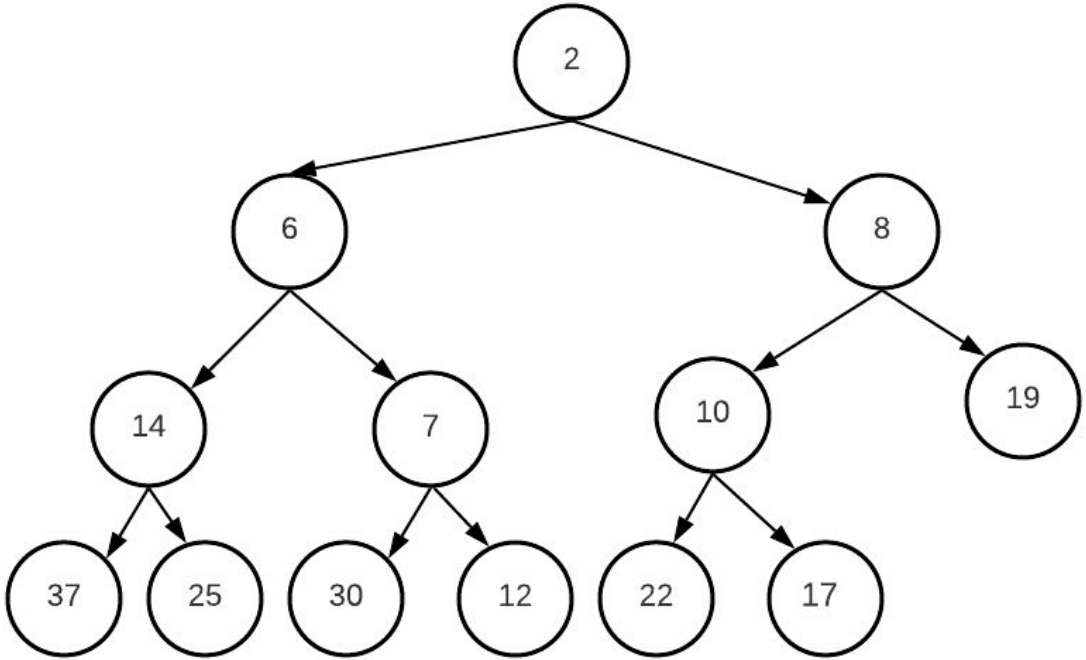
- To insert a value in the heap, we can use a nice trick
- We first ADD the item to the tree, and then FIX any corruption
    - A smart approach, but sadly not widely applicable



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	6	8	14	7	10	19	37	25	30	12	22	17		

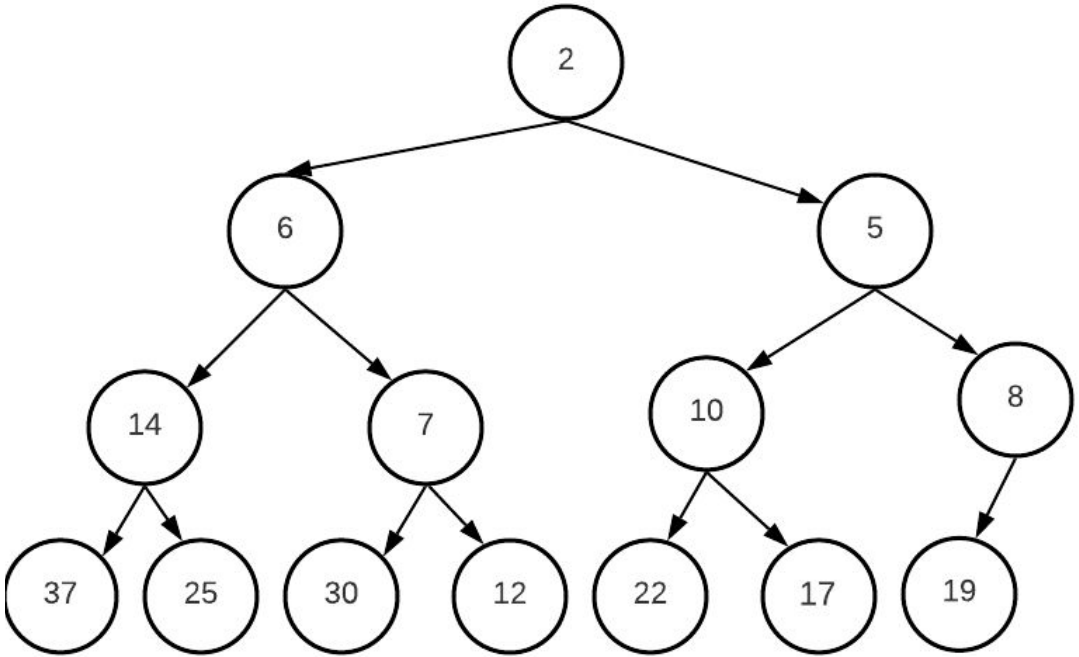
Let’s insert 5

- What is the first available node in this tree?
  - Left of 19
- Add 5
- What nodes make up the parent chain?
  - [5, 19, 8, 2]
- Shift up 5 to be in its **right location** (decreasing seq)
  - [19, 8, 5, 2]



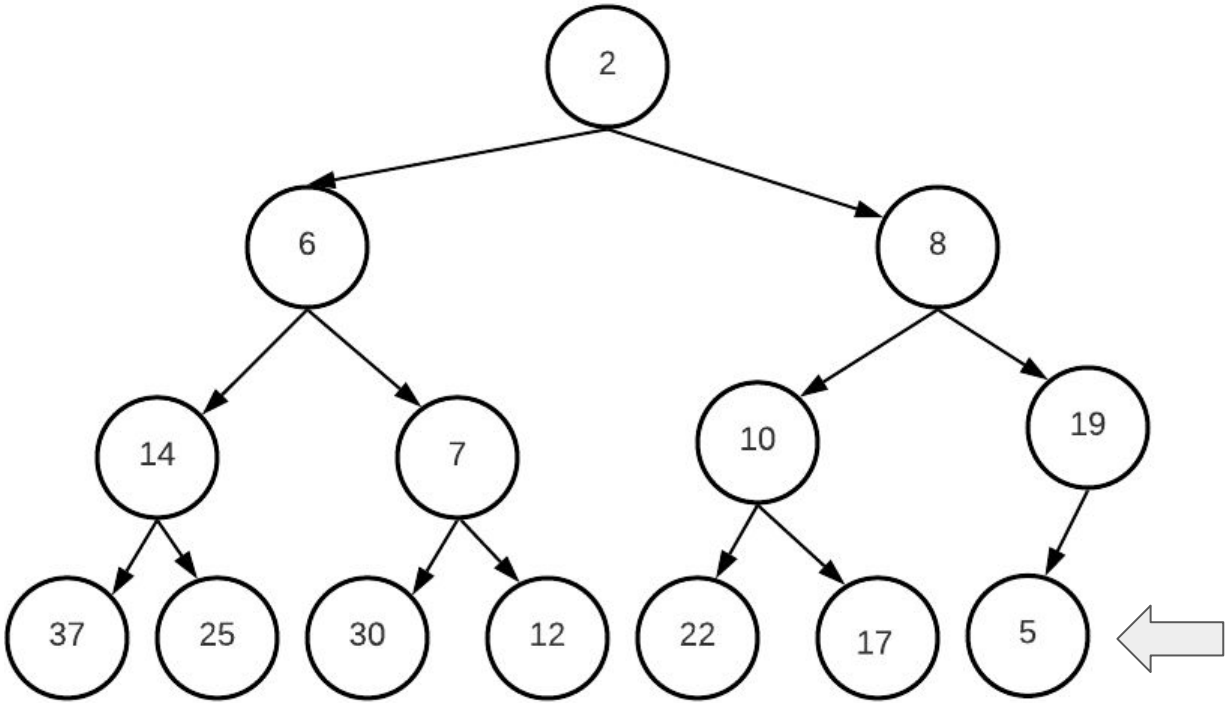
Heapify Up

- How can we implement that?
- Start from the index where the node was initially added
- If the value of the parent node is greater, do a simple swap! Shift the parent node down to where the current child node is, and move our newly added node up towards its correct location
    - Observe: 19 and 8 are moved down



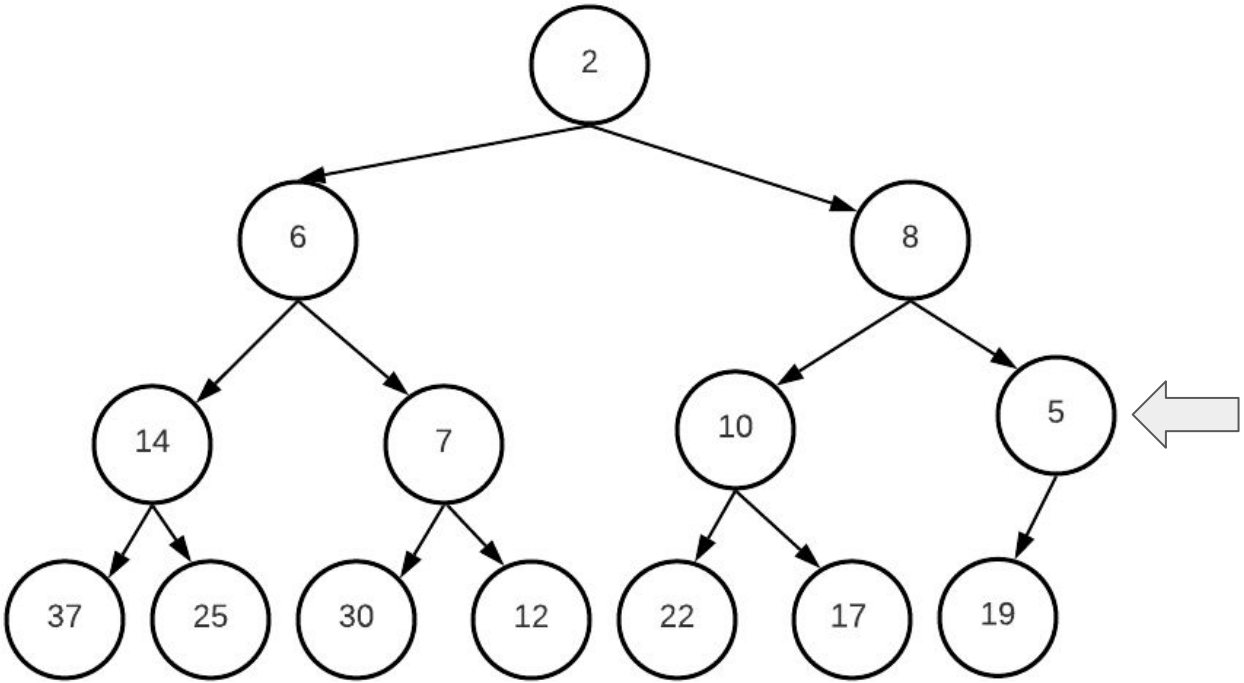
Simulation: Add 5 to the available position

- 5 index = 13
  - Parent idx = 6
  - Parent value 19
- 5 < 19
  - Push 19 down



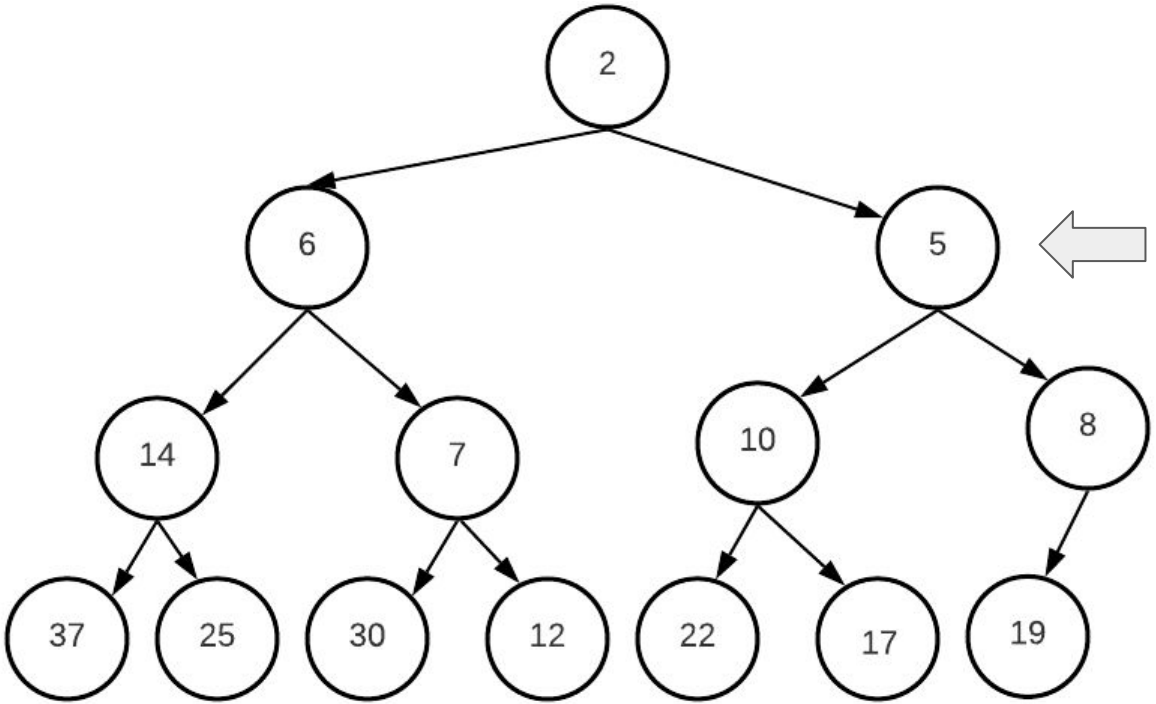
Simulation: Compare with parent

- 5 index = 6
  - Parent idx = 2
  - Parent value 8
- 5 < 8
  - Push 8 down



Simulation: Compare with parent

- 5 index = 2
  - Parent idx = 0
  - Parent value 2
- 5 > 2
  - Perfect heap
  - Stop
- Take 10 min to code it



"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."