

# MIN HEAP

- ① Build
- ② Sift up & down
- ③ pop, swap, insert, pop

## CONSTRUCTION

★ not  $n(\log n)$

$O(n)$

### Build Heap

- call **Sift Down**
- Start w the last parent node.

- (17)

- compare w children node.  
17, 12, 31

12 > 17 > 31  $\therefore 12 \leftrightarrow 17$

- Second last parent (23)

9 < 23 < 49  $9 \leftrightarrow 23$

Min Heap = [ 8, 12, 23, 17, 31, 30, 44, 102, 18, 9 ]

- 3rd parent node (102)

12 < 18 < 102  $12 \leftrightarrow 102$

102 has child nodes 18 = floor((8-1)/2) = 3  
102, 17, 31  $17 < 31 < 102 \therefore 17 \leftrightarrow 102$

- ① Insert Node. eg 9

- add as the last node

- 31 > 9

- not satisfy MinHeap.

$O(\log n)$  - eliminate half the tree everytime sift is used.

- **Sift up** (swap current w parent)

31 < 9  
12 < 9

8 < 9

- ② Remove Node [ 21, 9, 23, 17, 12, 30, 44, 102, 18 ]

- swap w final value in the heap.

- pop it & remove.

- But the heap is OUT OF ORDER

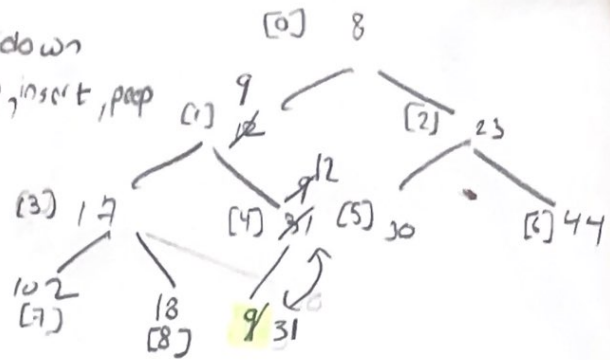
- **Sift down** (find the smallest of children node & swap w current unordered node)

① 23 > 9

9 < 31

② 31, 17, 12

③ < 17, ② < 31  $\therefore 12 \leftrightarrow 31$



currentNode  $\rightarrow i$   
 child one  $\rightarrow 2i+1$   
 child two  $\rightarrow 2i+2$

parentNode = floor((i-1)/2)

time = 0  
space = 0

